Published in final edited form as:

Stat Biosci. 2015 October 1; 7(2): 225–244. doi:10.1007/s12561-014-9114-4.

Forward Stagewise Shrinkage and Addition for High Dimensional Censored Regression

Zifang Guo

Department of Statistics, North Carolina State University, Raleigh, NC, 27695, USA
zifang.guo@gmail.com

Wenbin Lu

Department of Statistics, North Carolina State University, Raleigh, NC, 27695, USA
lu@stat.ncsu.edu

Lexin Li

Department of Statistics, North Carolina State University, Raleigh, NC, 27695, USA
li@stat.ncsu.edu

Abstract

Despite enormous development on variable selection approaches in recent years, modeling and selection of high dimensional censored regression remains a challenging question. When the number of predictors p far exceeds the number of observational units n and the outcome is censored, computations of existing solutions often become difficult, or even infeasible in some situations, while performances frequently deteriorate. In this article, we aim at simultaneous model estimation and variable selection for Cox proportional hazards models with high dimensional covariates. We propose a forward stage-wise shrinkage and addition approach for that purpose. Our proposal extends a popular statistical learning technique, the boosting method. It inherits the flexible nature of boosting and is straightforward to extend to nonlinear Cox models. Meanwhile it advances the classical boosting method by adding explicit variable selection and substantially reducing the number of iterations to the algorithm convergence. Our intensive simulations have showed that the new method enjoys a competitive performance in Cox models with both $p < n$ and $p > n$ scenarios. The new method was also illustrated with analysis of two real microarray survival datasets.

Keywords

Adaptive LASSO; boosting; forward stagewise regression; proportional hazards model; variable selection

1 Introduction

With rapidly advancing technologies, high dimensional data analysis has become ubiquitous in modern scientific research. For instance, in a typical microarray study, the number of covariates (genes) p is in thousands or more, whereas the number of experimental units n is much smaller, very often only in tens to a few hundreds. For such high dimensional data, it is commonly conjectured that only a subset of variables are relevant to the outcome, and as

such variable selection is a vital component of data analysis. In addition to the high dimensionality, in many biomedical applications, the outcome variable is also subject to censoring, for example, time to death and time to cancer recurrence. Cox proportional hazards model (Cox, 1972) has been the most popular tool for analyzing censored responses. However, Cox model estimation and variable selection when $p \gg n$ is challenging, and this is to be the focus of our article.

In the past decade, many research efforts have been devoted to the area of variable selection, and a large number of selection methods have been proposed. Among them, an outstanding class is the shrinkage approach, e.g., least absolute shrinkage and selection operator (LASSO) (Tibshirani, 1996), smoothly clipped absolute deviation (SCAD) method (Fan and Li, 2001), elastic net (Zou and Hastie, 2005), adaptive LASSO (Zou, 2006), nonnegative garrote (Yuan and Lin, 2007), and many others. Some of those methods have been extended to the Cox model, e.g., Fan and Li (2002), Tibshirani (1997), and Zhang and Lu (2007). In general, this class of solutions can be formulated in terms of a loss function plus a regularization term, and its minimization leads to a sparse estimate of the regression parameter, which in effect achieves simultaneously variable selection and parameter estimation. It has been shown that those methods perform competently in various settings especially when p is small to moderate.

However, when facing high dimensional problems, for instance, microarray data analysis where p is huge, the inherent computational complexity of the aforementioned methods may cause algorithmic instability and yield estimators with large variance. In addition, when $p \gg n$, methods such as the adaptive LASSO become either difficult or infeasible to implement. One practical solution is sure independence screening (SIS) (Fan and Lv, 2008; Fan et al, 2010) that filters the predictors via marginal measure between individual predictor and the response. An alternative solution for $p \gg n$ is forward stepwise regression (FR) that selects one variable at a time and avoids simultaneous handling of all predictors (Wang, 2009). Numerical studies have suggested superior performances of both SIS and FR when $p \gg n$. On the other hand, both focus on variable screening, while model estimation is usually carried out separately *after* the selection. Other examples of high dimensional survival analysis include Ishwaran et al (2010), van Wieringen et al (2009) and Witten and Tibshirani (2010).

A method that works in a similar fashion as forward stepwise regression is *forward stagewise regression*, which builds a model iteratively, and usually involves only one variable at each iteration (Li and Luan, 2005; Lu and Li, 2008). It differs from FR in that, once a term is added into the model, its coefficient remains unchanged in all subsequent iterations. A successful application and generalization of forward stagewise regression is *boosting*. Boosting repeatedly applies a fitting method, called the base learner, to the reweighted data and updates the estimator by adding a newly fitted base learner at each iteration. The final estimator is constructed as a weighted sum of the series of the fitted base learners. Because the base learner used in boosting can be both linear and nonlinear, this method can fit data with a more flexible structure than a linear form. Besides, the base learner usually only involves one variable at a time, and thus it works for large p regressions. Boosting first originated in the machine learning community known as

AdaBoost (Freund, 1995; Freund and Schapire, 1997; Schapire, 1990) for classification. Later Friedman et al (2000) showed that the Adaboost algorithm is in fact equivalent to fitting a forward stagewise additive model by minimizing a particular exponential loss function. Friedman (2001) further proposed a general gradient descent boosting algorithm that can accommodate a variety of loss functions, and a number of extensions followed (Bühlmann and Hothorn, 2007; Bühlmann and Yu, 2003; Ridgeway, 1999). Applications of boosting to censored data were also developed, including Li and Luan (2005) for Cox proportional hazards model and Lu and Li (2008) for transformation models. Despite its flexible nature, its generality to cope with various types of regressions, and its competitive empirical performance, boosting also has some limitations. First, boosting does not perform explicit variable selection. A variable is selected if its coefficient is nonzero when the algorithm converges or stops. Relative contributions of individual variables are measured by a heuristic importance measure (Friedman, 2001), while there is no associated inference available to separate the active predictors from the inactive ones. Second, a boosting algorithm often takes a very large number of iterations and thus a long computation time. This is mainly due to the fact that a small learning rate is imposed to the model update at each iteration to achieve proportional shrinkage (Friedman, 2001). As a consequence, adding an active covariate into the model may take tens or more iterations to complete.

In this article, we aim to address simultaneous model estimation and variable selection in Cox proportional hazards models with high dimensional predictors. We couple the strategies of shrinkage estimation and forward stagewise boosting, and propose a FOward Stagewise Shrinkage and Addition (FOSSA) method, which carries out an additive stagewise modeling while introducing shrinkage at each iteration. Our contributions are two fold. First, the proposed method works naturally for high dimensional regressions. When facing high dimensional predictors, existing solutions often require a pre-screening as the first step, followed by a refined variable selection and parameter estimation as the second step. By contrast, our method does not require any pre-screening, and in effect combines the two tasks in one stroke. Our intensive simulations show that it performs competitively compared to the existing methods for both $n > p$ and $n \leq p$ scenarios. Second, our solution inherits the flexibility of boosting. We will focus on linear Cox models in this article, but extensions to nonlinear Cox models are straightforward and will be briefly discussed at the end. On the other hand, the proposed method also extends the existing boosting method in that, it now performs explicit variable selection, and it greatly reduces the number of iterations for the algorithm to converge. As we will see later in simulations, the new method often converges in tens of steps, compared to hundreds or more iterations required by the usual boosting.

The rest of the article is organized as follows. In Section 2, we first briefly review the Cox proportional hazards model, adaptive LASSO, and forward stagewise boosting. We then present our new method in detail. In Section 3, we conduct an intensive simulation study to investigate the empirical performance of the proposed method, and to compare it with existing variable selection and boosting solutions, followed by the analyses of two microarray data examples for further illustration. We conclude in Section 4 with a discussion on future extensions.

2 Model and Method

2.1 Cox Proportional Hazards Model with Adaptive LASSO

Our discussion hereinafter will assume the context of survival data analysis, while the methodology applies to other censored regressions as well. Considering n random subjects, let T_i be the failure time, C_i be the censoring time, and $z_i = (z_{i1}, \dots, z_{ip})^T$ be the vector of covariates of the i -th subject, $i = 1, \dots, n$. Define the observed event time $\tilde{T}_i = \min(T_i, C_i)$ and the censoring indicator $\delta_i = I(T_i \leq C_i)$. Then the observed data consist of $\{(\tilde{T}_i, \delta_i, z_i), i = 1, \dots, n\}$. Furthermore, we assume conditional independent censoring, i.e., T_i and C_i are independent conditioning on z_i , throughout this paper.

The Cox proportional hazards model assumes that the hazard function $\lambda(t|z_i)$ of subject i with covariate z_i is given as:

$$\lambda(t|z_i) = \lambda_0(t) \exp(\beta^T z_i), \quad \text{for } i=1, \dots, n, \quad (1)$$

where $\lambda_0(t)$ is the unspecified baseline hazard function and $\beta = (\beta_1, \dots, \beta_p)^T$ is the vector of regression coefficients. Assuming there are no tied observations, then the log partial likelihood function (Cox, 1975) is given by:

$$\ell_n(\beta) = \sum_{i=1}^n \delta_i \left[\beta^T z_i - \log \left\{ \sum_{j=1}^n \exp(\beta^T z_j) I(\tilde{T}_j \geq \tilde{T}_i) \right\} \right]. \quad (2)$$

An estimate of β is obtained by minimizing $-\ell_n(\beta)$ over β .

To perform variable selection under this model, Zhang and Lu (2007) proposed the adaptive LASSO method, which minimizes the weighted L_1 penalized negative log partial likelihood,

$$-\frac{1}{n} \ell_n(\beta) + \lambda \sum_{j=1}^p |\beta_j| / |\tilde{\beta}_j|,$$

where $\tilde{\beta} = (\tilde{\beta}_1, \dots, \tilde{\beta}_p)^T$ is the maximizer of (2). It is shown that, asymptotically, this adaptive LASSO estimator enjoys the oracle properties, and numerically, the method performs competitively when $p < n$. However, when $p > n$, $\tilde{\beta}$ is not directly available, and thus the adaptive LASSO method cannot be applied to the Cox model when $p > n$.

2.2 A Modified Boosting Algorithm

Before we introduce our FOSSA method, we first present a version of boosting algorithm for the Cox model that will help the establishment of FOSSA. It is similar in spirit as the proposal of Li and Luan (2005) though not identical. The basic idea is to model the survival time in the form of $\lambda(t|z) = \lambda_0(t) \exp\{F(z)\}$, where the function $F(z)$ is updated by iteratively adding new terms. Since the terms added to $F(z)$ in precedent steps remain unchanged in future iterations, they are to be treated as an offset term in subsequent model fitting. We thus denote the log partial likelihood function with an offset term $\mu = (\mu_1, \dots, \mu_n)^T$ as

$$\ell_n(\boldsymbol{\beta}, \boldsymbol{\mu}) = \sum_{i=1}^n \delta_i \left[\mu_i + \boldsymbol{\beta}^T \mathbf{z}_i - \log \left\{ \sum_{j=1}^n \exp(\mu_j + \boldsymbol{\beta}^T \mathbf{z}_j) I(\tilde{T}_j \geq \tilde{T}_i) \right\} \right].$$

Meanwhile, we use $\ell_n(\beta_g, \boldsymbol{\mu})$ to denote the log partial likelihood function when only the g -th covariate z_g is fitted in the Cox model. That is,

$$\ell_n(\beta_g, \boldsymbol{\mu}) = \sum_{i=1}^n \delta_i \left[\mu_i + \beta_g z_{ig} - \log \left\{ \sum_{j=1}^n \exp(\mu_j + \beta_g z_{ij}) I(\tilde{T}_j \geq \tilde{T}_i) \right\} \right].$$

We next present the boosting algorithm.

Step 1. Set initial values: iteration $k = 0$, the offset $\boldsymbol{\mu}^{[0]} = (\mu^{[0]}_1, \dots, \mu^{[0]}_n)^T = 0$, and the fitted function $F^{[0]}(z) = 0$.

Step 2. Repeat for $k = 0, \dots, K$.

(a) Obtain the coefficients:

$$\tilde{\beta}_g = \underset{\beta_g}{\operatorname{argmin}} \left\{ -\ell_n(\beta_g, \boldsymbol{\mu}^{[k]}) \right\}, \quad g = 1, \dots, p. \quad (3)$$

That is, we fit p univariate Cox models with each covariate $z_1, \dots, z_p$ as the sole predictor plus an offset $\boldsymbol{\mu}^{[k]}$.

(b) Select the covariate with minimum negative log partial likelihood to add in the model, i.e., $g^* = \underset{g=1, \dots, p}{\operatorname{argmin}} \left\{ -\ell_n(\tilde{\beta}_g, \boldsymbol{\mu}^{[k]}) \right\}$.

(c) Update the fitted function $F^{[k+1]}(z) = F^{[k]}(z) + \nu \tilde{\beta}_{g^*} z_{g^*}$ where ν is a small learning rate (e.g. 0.05 or 0.1). Also update the offset $\boldsymbol{\mu}^{[k+1]} = (F^{[k+1]}(z_1), \dots, F^{[k+1]}(z_n))^T$.

Step 3. Output the final estimate as $F^{[K]}(z)$.

We first note that this version of the boosting algorithm is slightly different from the gradient descent boosting of Friedman (2001) and Li and Luan (2005) for Cox models. The difference is at Step 2(a), where this algorithm estimates β_g via maximum partial likelihood, whereas in the usual gradient descent boosting, β_g is estimated by fitting a regression model with the negative gradient as the response. As such, the two procedures may end up selecting different variables to add into the model. But if the same variable is selected, the two procedures would yield the same fitted function $F(z)$, because the gradient descent boosting (Li and Luan, 2005) has a line search step, which essentially fits a Cox model with the selected variable. On the other hand, if there is no response censoring and the loss function is chosen to be the usual squared loss, the two algorithms are equivalent, because both methods repeatedly fit the least squares residuals. We make the following remarks about this algorithm, which also apply to gradient boosting algorithms using component-wise base learners. First, the number of boosting iterations K is usually tuned through cross-

validation methods. Second, we note that, at each iteration, univariate regressions are carried out, and variable is added into the model one-at-a-time. This strategy is expected to be helpful for a very large p . Third, a small learning rate v is imposed to regularize the contribution of the newly selected term through proportional shrinkage. This is because once a coefficient is selected into the model, its coefficient remains unadjusted afterwards, and so adding a small learning rate would help reduce the effect of “incorrectly” estimated parameters to the final model. Finally, no explicit variable selection is performed. Usually it would take many steps to add a truly relevant variable into the model, whereas many irrelevant variables are to be included in the model with a tiny coefficient estimate.

2.3 Forward Stagewise Shrinkage and Addition

To capitalize on the advantages whereas to overcome the drawbacks of the adaptive LASSO shrinkage estimation and the boosting method, we propose to couple the two strategies for the purpose of variable selection in high dimensional Cox proportional hazards models. The basic idea is to introduce shrinkage estimation at each iteration of the boosting algorithm, meanwhile dropping the learning rate. As such it in effect replaces the proportional shrinkage dictated by the common learning rate v with an adaptive shrinkage at each iteration. We call the resulting algorithm the forward stagewise shrinkage and addition method (FOSSA). Specifically, the FOSSA algorithm is as follows:

Step 1. Set initial values: iteration $k = 0$, the offset $\boldsymbol{\mu}^{[0]} = (\mu_1^{[0]}, \dots, \mu_n^{[0]})^T = \mathbf{0}$, and the fitted function $F^{[0]}(z) = 0$.

Step 2. Obtain the coefficients:

$$\hat{\beta}_g \argmin_{\beta_g} \left\{ -\frac{1}{n} \ell_n(\beta_g, \boldsymbol{\mu}^{[k]}) + \frac{\lambda}{|\hat{\beta}_g|} |\beta_g| \right\}, \quad g=1, \dots, p, \quad (4)$$

where $\tilde{\beta}_g$ is the unpenalized estimate obtained in (3), and λ is a shrinkage parameter. So in effect, we fit p adaptive LASSO type Cox models with each covariate $z_1, \dots, z_p$ as the sole predictor plus an offset $\boldsymbol{\mu}^{[k]}$.

Step 3. Select the covariate with minimum negative log partial likelihood to add in the model, i.e., $g^* = \argmin_{g=1, \dots, p} \left\{ -\ell_n(\hat{\beta}_g, \boldsymbol{\mu}^{[k]}) \right\}$.

Step 4. Update the fitted function $F^{[k+1]}(z) = F^{[k]}(z) + \hat{\beta}_{g^*} z_{g^*}$. Also update the offset $\boldsymbol{\mu}^{[k+1]} = (F^{[k+1]}(z_1), \dots, F^{[k+1]}(z_n))^T$.

Step 5. Increment k by 1. Go back to Step 2 until convergence (i.e. $\{\ell_n(\mathbf{0}, \boldsymbol{\mu}^{[k+1]}) - \ell_n(\mathbf{0}, \boldsymbol{\mu}^{[k]})\}/n < tol$, where tol is a pre-specified value such as $1e-4$).

Comparing the FOSSA algorithm with the forward stagewise boosting algorithm, we note that the differences are in Step 2, where an adaptive LASSO penalty is introduced in estimating $\hat{\beta}_g$, and in Step 4, where the learning rate v is no longer applied. The consequences of those changes are the following. First, variable selection is now carried out explicitly; the truly inactive predictor would usually not enter the model due to the large

penalty, and the truly active predictor could enter the model faster due to the drop of the small learning rate. Second, the new algorithm converges very fast, and thanks to the regularization in coefficient estimation, also usually achieves a more accurate estimation compared to the usual boosting algorithm. This will later be verified in our simulations.

Implementation of FOSSA requires solving the minimization problem in (4) and tuning of the shrinkage parameter λ . For the first task, the optimization is carried out by adopting the least squares approximation (LSA) idea of Wang and Leng (2007) used for LASSO and adaptive LASSO implementations. More specifically, we aim to minimize the objective function

$$-\frac{1}{n}\ell_n(\beta_g, \boldsymbol{\mu}^{[k]}) + \frac{\lambda}{|\tilde{\beta}_g|}|\beta_g|, \quad (5)$$

where $\tilde{\beta}_g$ is the maximum partial likelihood estimate obtained in (3). Applying the Taylor series expansion of the negative log partial likelihood at $\tilde{\beta}_g$, we obtain

$$-\frac{1}{n}\ell_n(\beta_g, \boldsymbol{\mu}^{[k]}) \approx -\frac{1}{n}\ell_n(\tilde{\beta}_g, \boldsymbol{\mu}^{[k]}) + \frac{1}{n}G(\tilde{\beta}_g)(\beta_g - \tilde{\beta}_g) + \frac{1}{2n}H(\tilde{\beta}_g)(\beta_g - \tilde{\beta}_g)^2,$$

where $G(\tilde{\beta}_g)$ and $H(\tilde{\beta}_g)$ denote the first and second derivative of $-\ell_n(\beta_g, \boldsymbol{\mu}^{[k]})$ with respect to β_g that is evaluated at $\tilde{\beta}_g$. Furthermore, we note that $G(\tilde{\beta}_g) = 0$. Then, by ignoring the constant, we can rewrite the objective function in (5) as

$$\frac{1}{2n}H(\tilde{\beta}_g)(\beta_g - \tilde{\beta}_g)^2 + \frac{\lambda}{|\tilde{\beta}_g|}|\beta_g|.$$

Its minimizer is then given by:

$$\hat{\beta}_g = \text{sign}(\tilde{\beta}_g) \left(|\tilde{\beta}_g| - \frac{n\lambda}{H(\tilde{\beta}_g)|\tilde{\beta}_g|} \right)_+,$$

which can be equivalently written as

$$\hat{\beta}_g = \begin{cases} \tilde{\beta}_g - n\lambda \{H(\tilde{\beta}_g)\tilde{\beta}_g\}^{-1} & \text{if } \tilde{\beta}_g^2 > n\lambda H(\tilde{\beta}_g)^{-1} \\ 0 & \text{otherwise} \end{cases}$$

For the task of tuning λ , we employ a Bayesian information criterion (BIC)

$$-2\hat{\ell}_n + \log(n) d_\lambda, \quad (6)$$

where $\hat{\ell}_n$ denotes the final log partial likelihood function after the algorithm converges, and d_λ is the number of nonzero covariates in the final model. We then search over a grid values of λ and choose the one that minimizes (6). In this algorithm, we have chosen a common shrinkage parameter λ for all covariates across all iterations. We also note that, despite this choice of common λ , the amount of regularization is adaptively dependent upon the magnitude of the individual covariate $|\tilde{\beta}_g|$.

Alternatively, we may also specify a different tuning parameter $\lambda_g^{[k]}$ in (4) for each covariate g at each iteration k . We can continue to use BIC to tune this parameter, except that $\hat{\ell}_n$ in (6) is replaced with the log partial likelihood function at the current state, and d_λ only takes two values: 0 or 1. Consequently, there are only two effective choices of λ , either

$(\tilde{\beta}_g)^2 H(\tilde{\beta}_g) / n$ or 0, which yields either $\hat{\beta}_g = 0$ or $\hat{\beta}_g = \tilde{\beta}_g$, respectively. This way of tuning is computationally simpler. We have compared the two tuning strategies, and found that (results not reported here), the common λ with a grid search achieves a comparable performance as the varying λ when p is small, and shows an edge when p is large. For this reason, we will adopt the common λ strategy in the rest of the article.

A natural question that might be raised is whether we could combine the shrinkage method with a small learning rate in this stagewise approach. We have in fact investigated this possibility by combining the FOSSA method with a small learning rate of 0.1 when adding each coefficient in the model in Step 4 of the algorithm, i.e. multiplying the adaptive LASSO coefficient by 0.1 at each iteration. For computational simplicity, a different tuning parameter $\lambda_g^{[k]}$ in (4) for each covariate g at each iteration k with BIC tuning was employed in this combined approach. The results (not shown) suggest that this combined method in general is not a good choice, as combining the adaptive LASSO with a small learning rate usually leads to over-shrinkage of the coefficient estimates, and as a result greatly reduces the estimation accuracy. Therefore, combining the shrinkage method with a small learning rate is not recommended.

3 Numerical Study

We conducted an intensive simulation study to evaluate the empirical performance of FOSSA. Three scenarios were considered: a usual setup of a linear Cox model with $p < n$, a linear Cox model with $p = n$, and a full quadratic Cox model where $p < n$ but the total number of terms in the model, including the linear terms, the quadratic terms and all the two-way interaction terms, far exceeds the sample size. We also compare FOSSA with several existing methods that apply to Cox model, including the boosting method, LASSO, adaptive LASSO (aLASSO), forward stepwise selection (FR), SIS and iterative SIS (ISIS) followed by SCAD estimation, denoted by SIS-SCAD and ISIS-SCAD, respectively. As a benchmark, we include the oracle estimator in comparison as well.

3.1 Linear Model with $p < n$

We first generated the covariates z_i , $i = 1, \dots, n$, according to a multivariate normal distribution with mean zero, variance one, and an order one autoregressive correlation

structure, where $\text{corr}(z_{ij}, z_{ik}) = 0.5^{|j-k|}$, $j, k = 1, \dots, p$. The failure time T_i 's were then generated following the Cox proportional hazards model given in (1) with $\lambda_0(t) = 1$ and $\beta = (1, 0.8, 0, \dots, 0, 0.6)^T$. In other words, only the first two and the last predictors are active. The censoring time C_i 's were generated independently from an exponential distribution with mean of $1/\exp(C_0 + 0.3z_2 + 0.5z_3)$, where C_0 was varied to achieve overall censoring rate of 25%, 50% and 75%, separately. Sample size was chosen as $n = 150$, and the number of predictors was chosen as $p = 20$ and $p = 30$. A total of 100 data replications were performed. We have noted in our simulations that the FR algorithm sometimes fails to converge. For that reason, the maximum number of iterations in FR was limited to be $\min([n/\log(n)], p)$, and the final model was chosen based on BIC selection. For boosting, we used the mboost package in R (Bühlmann and Hothorn, 2007), and performed gradient boosting using component-wise univariate linear models. The number of iterations was chosen via five fold cross-validation. Least squares approximation together with BIC selection was used for FOSSA, LASSO and aLASSO implementations in this scenario. SIS-SCAD and ISIS-SCAD were both implemented using the SIS package in R. For this small p setting, the number of covariates recruited by the (I)SIS step was set as $d = [n/\log(n)/4]$, which is the default setting for SIS in R for Cox model. We note that this is in fact in favor of (I)SIS as $[n/\log(n)/4] = 7$ for $n = 150$ and the number of true active parameters was only 3 in this setting.

We evaluate and compare methods by three categories of criteria. The first is the average mean squared error (MSE), $(\hat{\beta} - \beta)^T \Sigma (\hat{\beta} - \beta)$, where Σ is taken to be the population covariance matrix of the covariates z in this setup. This criterion measures the estimation accuracy of the model parameter β . The second category examines the performance in terms of variable selection accuracy. Criteria in this category include the average size of the selected models (Size), the frequency of selecting all truly active predictors (Cover) out of 100 data replications, the frequency of selecting the exact model (Exact), the percentage of correct zeros being identified (Corr0), and the percentage of incorrect zeros being identified (Incorr0). The third category compares the boosting algorithm and the new FOSSA algorithm in terms of the average iterations (Iter) required.

Table 1 reports the results. At 25% censoring rate, in terms of both parameter estimation accuracy and variable selection accuracy, FOSSA, aLASSO, FR, ISIS-SCAD and SIS-SCAD perform similarly well and provide smallest MSEs, with SIS-SCAD has slightly lower coverage than others. At 50% censoring rate, FOSSA and SIS-SCAD gave smallest MSEs, with SIS-SCAD again having lower coverage percentage. At 75% censoring rate, the performance of all methods deteriorate. However, FOSSA together with boosting and SIS-SCAD still provide smallest MSE among all methods. In addition, we note that LASSO appears to be less accurate in estimating β , as reflected by a larger MSE, partly due to the fact that its estimation is biased. The usual boosting estimator always selects more variables than necessary, as indicated by a much larger model size than the truth ($p_0 = 3$). Consequently, boosting rarely picks up the exact model. In addition, comparing the number of iterations, FOSSA requires far fewer steps than boosting. Overall, the results show that FOSSA achieves a fairly competitive performance among other methods in this $p < n$ setup.

3.2 Linear Model with $p = n$

In the second example, the data were generated in the same way as the $p < n$ scenario, except that $\beta = (1, 1, 0, \dots, 0, 1)^T$, with $p = 100, 500$, and 1000 . The sample size was chosen as $n = 100$ and censoring rate was controlled as 25%, 50% and 75%. So now we have a linear Cox model with $p = n$. Since adaptive LASSO cannot be directly applied to this $p = n$ case, it is not included in the comparison. As a result, we compare the performance of FOSSA, boosting, LASSO, FR, ISIS-SCAD and SIS-SCAD. In this $p = n$ scenario, we employ the `glmnet` function in R (Friedman et al, 2010) for the implementation of LASSO. In addition, since p is much larger, we set $d = \lfloor n / \log(n) \rfloor$ for SIS-SCAD as suggested by Fan and Lv (2008), while d was remained as $\lfloor n / \log(n) / 4 \rfloor$ for ISIS-SCAD because the algorithm became unstable and too many warnings/errors were reported for large d , especially for the higher censoring rates.

Table 2 reports the results. At 25% censoring rate, FOSSA and SIS-SCAD provide smallest MSEs among all six methods, while the performance of SIS-SCAD deteriorates more as p increases. At 50% and 75% censoring rates, FOSSA gives smallest MSE among all methods, with relatively high to medium coverage. We note that FR yields a substantially large MSE, indicating a very inaccurate parameter estimation. The performance of SIS-SCAD decreases dramatically as censoring rate increases. Although boosting and LASSO tend to have higher coverage percentages, they result in a much larger model ($p_0 = 3$). In addition, in this large p case, FOSSA and SIS-SCAD are the only methods that were able to select the exact model, which was rarely picked by other methods. Moreover, the FOSSA algorithm converges quickly even with a very large p , and requires much less iteration steps compared to boosting. Besides, we note that both SIS-SCAD and ISIS-SCAD are two-step procedures, which include a screening step, followed by a regularized model fitting step. By contrast, FOSSA incorporates model selection and estimation into a single stroke, and conceptually speaking the estimation accuracy of FOSSA might be improved if additional model fitting steps were added. Overall, the results suggest that although no method is superior to all others in every aspect and every circumstance, FOSSA demonstrated a very competitive performance in terms of achieving a good balance of estimation accuracy and model selection ability among others in this $p = n$ scenario.

3.3 Quadratic Model with More Terms than Sample Size

In reality, the true association between the survival time and the covariates is often likely to be more complicated than the linear Cox model depicted in (1). One option to incorporate such a complex association is to include higher order polynomials of the covariates, e.g., the quadratic and interaction terms. In principle, the selection methods that work for the linear model work for the polynomial model too, by treating all those higher order terms as additional covariates. However, practically, the effective number of predictors in a polynomial model grows rapidly with the number p of the original predictors, and based upon the observations in Section 3.2, the existing solutions are expected to suffer from a deteriorating accuracy in both model estimation and variable selection. In this section, we consider a simulation example of this type. The original covariates z_i 's were generated in the same way as in Section 3.1. Next the failure time T_i 's were generated based on a quadratic

Cox model, where $\lambda(t|z) = \exp(z_1 - z_2 + z_2^2 - 0.8z_1z_5 + 0.8z_6z_9)$. The censoring time C_i 's were generated independently from a uniform distribution with 20% censoring proportion for this more challenging scenario. The sample size was chosen as $n = 100$, and p took values of 20 and 30. In addition to the original covariates, all their quadratic and two-way interaction terms were included as candidate variables for all methods, resulting in an effective number $\tilde{p}$ of covariates equal to $p + p(p + 1)/2 = 230$ and 495, respectively. Since $\tilde{p}$ is much larger than n , we again compare our FOSSA solution with boosting, LASSO, FR, SIS-SCAD and ISIS-SCAD.

Table 3 reports the results. Here, to simplify the computation of MSE, we take Σ equal to an identity matrix in MSE. FOSSA, boosting and ISIS-SCAD achieve smallest MSEs in this scenario, while boosting has a much larger than necessary model size and a large iteration number, and ISIS-SCAD has a much lower coverage percentage. LASSO achieves slightly higher MSE, and selects many more variables than necessary. We also note that SIS-SCAD has very low coverage in this scenario. FR appears to have an unstable parameter estimation with a large MSE, and also a relatively large model size.

3.4 Real Data Analysis

We illustrate the application of our FOSSA method in the analysis of two microarray datasets. The first is a study of diffuse large-B-cell lymphoma (Rosenwald et al, 2002), which consists of $n = 240$ lymphoma patients and $p = 7399$ candidate genes. The response is the patient's survival time after the chemotherapy, among which 102 are censored, yielding a 42.5% censoring proportion. The second is a breast cancer study (van Houwelingen et al, 2006), where there are $n = 295$ female patients with breast cancer and $p = 4919$ candidate genes. The response is again the survival time, and among 295 patients, 216 have censored response, which results in a relatively high censoring rate of 73%. In both studies, the goal is to identify important genes that affect the survival phenotype. Given the large value of p , we consider a linear Cox model for both data. Both datasets used in our analysis were previous post-processed and analyzed in published papers (Bair et al (2006) and van Houwelingen et al (2006)).

For the lymphoma data, the FOSSA algorithm converged in 12 iterations, yielding a selection of 11 genes out of 7399 candidates. Among them, six were also selected by Li and Luan (2005) in their analysis of the same dataset using a boosting method. We also note that, to apply the boosting algorithm, Li and Luan (2005) conducted a preliminary screening using a univariate Cox model to first bring the number of candidates from 7399 to 50. By contrast, our method does not require any pre-screening step. Those 11 genes are listed in Table 4, by the order that they enter the model in FOSSA, along with their gene ID's and the estimated coefficients by FOSSA. We also fit a linear Cox model with only those 11 selected genes, and report the coefficient estimates and the corresponding p -values. It is seen that the estimates obtained from the final Cox model with high significance and those from FOSSA are quite compatible. In order to evaluate the predictive performance of the FOSSA method, and to compare with other methods, we generated 100 randomly splits of the data into training and testing datasets with a 2:1 ratio. For each split, a model was first built with the training dataset using each method. A cutoff value of high-risk and low-risk groups was

set as the median of $\hat{\beta}_{train}^T z_{train}$. Subjects in both the training and testing datasets were further classified into high-risk and low-risk groups based on the comparison of $\hat{\beta}_{train}^T z_{train}$ and $\hat{\beta}_{train}^T z_{test}$ with the same cutoff value, respectively. Log-rank test was then performed on both the training dataset and the testing dataset based on the risk group determination. p -values on training datasets were highly significant for almost all splits for all methods. Box plots of the log-rank test p -values of the 100 testing datasets are summarized in Figure 1(a). Although not highly differentiable, it is seen that FOSSA achieved lowest overall p -values between these two risk groups among all methods compared. We note that ISIS-SCAD appeared to give highest overall p -values among these five methods for this dataset. In addition, boosting and LASSO tend to yield a less sparse model than FOSSA. Separation of the risk groups by FOSSA is graphically shown in Figure 2(a) via the Kaplan-Meier estimates of survival curves together with the log-rank test results from one random split.

For the breast cancer data, FOSSA converged in 8 iterations, and selected 7 genes out of 4919 candidates. Table 4 lists those selected genes and their FOSSA coefficients. We again fit a Cox model using only those 7 genes, which all found significant based their p -values, and the two models yielded compatible estimates. Predictive performance of FOSSA was also evaluated by randomly splitting the dataset into training set and testing set with a 2:1 ratio 100 times. The results are shown in Figure 1(b) and Figure 2(b). For this dataset, all methods except ISIS-SCAD achieved statistically significant p -values at $\alpha = 0.05$ level for majority of the splits, with boosting appeared to give the lowest overall p -values. However, boosting again tends to select many more variables than FOSSA. These results indicate that FOSSA performs competitively well compared to others, and can be useful in building predictive models for censored survival data with high dimensional predictors.

4 Discussion

In this article, we have proposed a forward stagewise shrinkage and addition (FOSSA) method for simultaneously model estimation and variable selection in Cox proportional hazards models with high dimensional covariates. It carries out an additive stagewise modeling while introducing shrinkage estimation at each iteration. Compared with the existing variable selection methods, our method performs very competitively in both $p < n$ and $p = n$ setups. Compared with the existing boosting method, our solution explicitly conducts variable selection, and substantially reduces the number of iterations and thus computing time by dropping the small learning rate in the usual boosting algorithm. Therefore, it provides a useful addition to the statistical toolbox for the analysis of high dimensional survival data. In FOSSA, adaptive LASSO penalty is added at each iteration step of the boosting algorithm, which makes the derivation of the theoretical properties of FOSSA very challenging. It is an interesting topic that needs further investigation.

In addition, it is interesting to develop a boosting method with an adaptive learning rate. Specifically, we may consider a set of learning rate values, say from 0.1 to 1. Then, in a modified boosting algorithm, we use 5-fold cross-validation method to adaptively select the optimal learning rate value, as well as the number iterations needed. However, such a method may be computationally intensive since it requires a two-dimensional tuning on both

the learning rate value in each iteration and the total number of iterations needed. This is another interesting topic that needs further investigation.

When the association between the response and the covariates is more complicated than a linear relation, a common remedy is to introduce higher order polynomial terms, e.g., the quadratic and interaction terms, into the model. Although the same methodology for linear model selection can be applied here, our results in Section 3.3 indicate that the problem becomes harder than merely having a larger number of covariates. This can be partly attributed to the high correlations among the covariates and their polynomial terms. Our solution exhibits a competitive performance under this scenario. Moreover, our method can be straightforwardly extended to nonlinear Cox models by considering more flexible base learner, e.g., the component-wise cubic smoothing splines (Bühlmann and Yu, 2003; Li and Luan, 2005; Lu and Li, 2008), and employing the group LASSO type penalty (Yuan and Lin, 2006). This extension is certainly of interest and is currently under investigation.

Acknowledgments

This work was supported by the National Institute of Health [grant RO1 CA140632] to W. Lu; and the National Science Foundation [grant DMS-1106668] to L. Li. We also thank the associate editor and three referees for their valuable comments and suggestions.

References

- Bair E, Hastie T, Paul D, Tibshirani R. Prediction by supervised principal components. *Journal of the American Statistical Association*. 2006; 101(473)
- Bühlmann P, Hothorn T. Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*. 2007; 22:477–505.
- Bühlmann P, Yu B. Boosting with the L2 loss: Regression and classification. *Journal of the American Statistical Association*. 2003; 98(462):324–339.
- Cox DR. Regression models and life-tables. *Journal of the Royal Statistics Society*. 1972; 34:187–220. Series B
- Cox DR. Partial likelihood. *Biometrika*. 1975; 62(2):269–276.
- Fan J, Li R. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*. 2001; 96:1348–1360.
- Fan J, Li R. Variable selection for Cox's proportional hazards model and frailty model. *Annals of Statistics*. 2002; 30:74–99.
- Fan J, Lv J. Sure independence screening for ultrahigh dimensional feature space. *Journal of The Royal Statistical Society*. 2008; 70(5):849–911. [PubMed: 19603084] Series B
- Fan, J.; Feng, Y.; Wu, Y. *Borrowing Strength: Theory Powering Applications-A Festschrift for Lawrence D Brown*. Institute of Mathematical Statistics; Beachwood, OH: 2010. High-dimensional variable selection for coxs proportional hazards model; p. 70-86.
- Freund Y. Boosting a weak learning algorithm by majority. *Information and Computation*. 1995; 121(2):256–285.
- Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*. 1997; 55(1):119–139.
- Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*. 2000; 28:337–374.
- Friedman JH. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*. 2001; 29:1189–1232.
- Friedman JH, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*. 2010; 33(1):1–22. [PubMed: 20808728]

- van Houwelingen HC, Bruinsma T, Hart AAM, van't Veer LJ, Wessels LFA. Cross-validated Cox regression on microarray gene expression data. *Statistics in Medicine*. 2006; 25:3201–3216. [PubMed: 16143967]
- Ishwaran H, Kogalur UB, Gorodeski EZ, Minn AJ, Lauer MS. High-dimensional variable selection for survival data. *Journal of the American Statistical Association*. 2010; 105(489):205–217.
- Li H, Luan Y. Boosting proportional hazards models using smoothing splines, with applications to high-dimensional microarray data. *Bioinformatics*. 2005; 21(10):2403–2409. [PubMed: 15713732]
- Lu W, Li L. Boosting method for nonlinear transformation models with censored survival data. *Biostatistics*. 2008; 9(4):658–667. [PubMed: 18344565]
- Ridgeway G. The state of boosting. *Computing Science and Statistics*. 1999; 31:172–181.
- Rosenwald A, Wright G, Chan WC, et al. The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *New England Journal of Medicine*. 2002; 346:1937–1947. [PubMed: 12075054]
- Schapire RE. The strength of weak learnability. *Machine Learning*. 1990; 5:197–227.
- Tibshirani R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*. 1996; 58:267–288.Series B
- Tibshirani R. The lasso method for variable selection in the Cox model. *Statistics in Medicine*. 1997; 16(4):385–395. [PubMed: 9044528]
- Wang H. Forward regression for ultra-high dimensional variable screening. *Journal of the American Statistical Association*. 2009; 104(488):1512–1524.
- Wang H, Leng C. Unified LASSO estimation by least squares approximation. *Journal of the American Statistical Association*. 2007; 102:1039–1048.
- van Wieringen WN, Kun D, Hampel R, Boulesteix AL. Survival prediction using gene expression data: A review and comparison. *Computational Statistics & Data Analysis*. 2009; 53:1590–1603.
- Witten DM, Tibshirani R. Survival analysis with high-dimensional covariates. *Statistical methods in medical research*. 2010; 19(1):29–51. [PubMed: 19654171]
- Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. *Journal of The Royal Statistical Society*. 2006; 68(1):49–67.Series B
- Yuan M, Lin Y. On the nonnegative garrote estimator. *Journal of the Royal Statistical Society*. 2007; 69:143–161.Series B
- Zhang HH, Lu W. Adaptive Lasso for Cox's proportional hazards model. *Biometrika*. 2007; 94(3): 691–703.
- Zou H. The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*. 2006; 101:1418–1429.
- Zou H, Hastie T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society*. 2005; 67:301–320.Series B

Fig. 1.
Real data analysis: p -values of log-rank test on testing datasets over 100 random splits.

(a) Diffuse large-B-cell lymphoma dataset

(b) Breast cancer dataset

Fig. 2.

Real data analysis: Kaplan-Meier estimates of survival curves for high and low risk patients (of one random split). Log-rank tests p values are given.

Table 1

Simulation results for a linear Cox model with $p < n$ ($n=150$). Results are averaged over 100 replications. Numbers in parentheses are standard errors.

Censoring Rate=25%								
p	Method	MSE	Size	Cover	Exact	Corr0	Incorr0	Iter
$p=20$	Oracle	0.05 (0.01)						
	FOSSA	0.10 (0.01)	3.55 (0.08)	1.00	0.63	0.968	0.000	5.1
	Boosting	0.14 (0.01)	8.29 (0.29)	1.00	0.02	0.689	0.000	177.4
	LASSO	0.25 (0.02)	4.53 (0.14)	1.00	0.27	0.910	0.000	
	aLASSO	0.10 (0.01)	3.55 (0.09)	1.00	0.66	0.968	0.000	
	FR	0.10 (0.01)	3.47 (0.06)	1.00	0.60	0.972	0.000	
	ISIS-SCAD	0.11 (0.01)	3.73 (0.13)	1.00	0.68	0.957	0.000	
	SIS-SCAD	0.09 (0.01)	3.29 (0.06)	0.97	0.71	0.981	0.010	
$p=30$	Oracle	0.05 (0.00)						
	FOSSA	0.11 (0.01)	3.71 (0.09)	1.00	0.54	0.974	0.000	5.1
	Boosting	0.15 (0.01)	9.39 (0.33)	1.00	0.00	0.763	0.000	158.3
	LASSO	0.35 (0.01)	4.27 (0.11)	1.00	0.29	0.953	0.000	
	aLASSO	0.12 (0.01)	3.61 (0.08)	1.00	0.57	0.977	0.000	
	FR	0.13 (0.01)	3.78 (0.10)	1.00	0.48	0.971	0.000	
	ISIS-SCAD	0.13 (0.01)	4.10 (0.16)	1.00	0.59	0.959	0.000	
	SIS-SCAD	0.09 (0.01)	3.34 (0.07)	0.95	0.68	0.986	0.017	
Censoring Rate=50%								
p	Method	MSE	Size	Cover	Exact	Corr0	Incorr0	Iter
$p=20$	Oracle	0.07 (0.01)						
	FOSSA	0.14 (0.01)	3.59 (0.08)	1.00	0.56	0.965	0.000	4.9
	Boosting	0.19 (0.01)	7.63 (0.27)	1.00	0.00	0.728	0.000	215.4
	LASSO	0.36 (0.02)	4.26 (0.12)	1.00	0.29	0.926	0.000	
	aLASSO	0.16 (0.01)	3.53 (0.09)	1.00	0.66	0.969	0.000	
	FR	0.15 (0.02)	3.52 (0.07)	1.00	0.57	0.969	0.000	
	ISIS-SCAD	0.18 (0.02)	3.85 (0.13)	1.00	0.58	0.950	0.000	
	SIS-SCAD	0.13 (0.02)	3.34 (0.07)	0.98	0.71	0.979	0.007	
$p=30$	Oracle	0.09 (0.01)						
	FOSSA	0.17 (0.01)	3.66 (0.10)	0.96	0.52	0.974	0.013	4.9
	Boosting	0.21 (0.01)	9.22 (0.34)	1.00	0.03	0.770	0.000	218.2
	LASSO	0.53 (0.03)	4.11 (0.12)	0.96	0.30	0.957	0.013	
	aLASSO	0.23 (0.02)	3.65 (0.10)	0.97	0.58	0.975	0.010	
	FR	0.23 (0.02)	3.93 (0.11)	1.00	0.47	0.966	0.000	
	ISIS-SCAD	0.28 (0.03)	4.51 (0.17)	1.00	0.44	0.944	0.000	
	SIS-SCAD	0.17 (0.02)	3.48 (0.09)	0.92	0.59	0.979	0.027	

Censoring Rate=75%								
p	Method	MSE	Size	Cover	Exact	Corr0	Incorr0	Iter
$p=20$	Oracle	0.13 (0.01)						
	FOSSA	0.33 (0.02)	3.22 (0.11)	0.76	0.51	0.969	0.100	4.3
	Boosting	0.32 (0.02)	7.14 (0.26)	0.95	0.03	0.754	0.017	298.1
	LASSO	0.80 (0.06)	3.29 (0.15)	0.62	0.25	0.955	0.160	
	aLASSO	0.51 (0.05)	2.79 (0.10)	0.62	0.46	0.982	0.173	
	FR	0.39 (0.06)	3.26 (0.09)	0.81	0.57	0.972	0.070	
	ISIS-SCAD	0.45 (0.06)	3.67 (0.13)	0.84	0.50	0.951	0.057	
	SIS-SCAD	0.28 (0.02)	3.17 (0.08)	0.78	0.58	0.976	0.077	
$p=30$	Oracle	0.17 (0.02)						
	FOSSA	0.40 (0.03)	3.29 (0.12)	0.72	0.40	0.977	0.113	4.4
	Boosting	0.37 (0.02)	7.79 (0.30)	0.92	0.03	0.820	0.027	290.7
	LASSO	1.46 (0.08)	2.11 (0.14)	0.34	0.21	0.989	0.393	
	aLASSO	0.92 (0.08)	2.37 (0.13)	0.48	0.33	0.990	0.297	
	FR	0.53 (0.05)	3.51 (0.09)	0.76	0.34	0.971	0.087	
	ISIS-SCAD	0.66 (0.06)	4.23 (0.16)	0.84	0.33	0.949	0.053	
	SIS-SCAD	0.38 (0.04)	3.31 (0.08)	0.75	0.45	0.979	0.083	

Table 2

Simulation results for a linear Cox model with p n ($n=100$). Results are averaged over 100 replications. Numbers in parentheses are standard errors.

Censoring Rate=25%								
p	Method	MSE	Size	Cover	Exact	Corr0	Incorr0	Iter
$p=100$	Oracle	0.09 (0.01)						
	FOSSA	0.30 (0.02)	4.31 (0.17)	1.00	0.39	0.986	0.000	6.0
	Boosting	0.39 (0.02)	13.96 (0.50)	1.00	0.00	0.887	0.000	168.7
	LASSO	0.46 (0.02)	10.98 (0.38)	1.00	0.00	0.918	0.000	
	FR	3.78 (0.58)	9.63 (0.54)	1.00	0.05	0.932	0.000	
	ISIS-SCAD	0.39 (0.02)	4.84 (0.05)	1.00	0.08	0.981	0.000	
	SIS-SCAD	0.24 (0.04)	4.40 (0.23)	1.00	0.55	0.986	0.000	
$p=500$	Oracle	0.07 (0.01)						
	FOSSA	0.44 (0.02)	5.07 (0.24)	1.00	0.29	0.996	0.000	6.6
	Boosting	0.60 (0.02)	19.67 (0.83)	1.00	0.00	0.966	0.000	147.5
	LASSO	0.75 (0.03)	13.42 (0.55)	1.00	0.01	0.979	0.000	
	FR	37.4 (1.74)	20.95 (0.04)	1.00	0.00	0.964	0.000	
	ISIS-SCAD	0.50 (0.03)	5.00 (0.00)	1.00	0.00	0.996	0.000	
	SIS-SCAD	0.36 (0.04)	6.11 (0.31)	0.99	0.29	0.994	0.003	
$p=1000$	Oracle	0.08 (0.01)						
	FOSSA	0.46 (0.02)	6.01 (0.33)	1.00	0.26	0.997	0.000	7.4
	Boosting	0.60 (0.02)	24.53 (1.05)	1.00	0.00	0.978	0.000	160.8
	LASSO	0.79 (0.03)	14.82 (0.64)	1.00	0.00	0.988	0.000	
	FR	59.4 (2.22)	20.77 (0.08)	1.00	0.00	0.982	0.000	
	ISIS-SCAD	0.66 (0.04)	5.00 (0.00)	1.00	0.00	0.998	0.000	
	SIS-SCAD	0.51 (0.07)	6.58 (0.34)	0.98	0.22	0.996	0.007	
Censoring Rate=50%								
p	Method	MSE	Size	Cover	Exact	Corr0	Incorr0	Iter
$p=100$	Oracle	0.13 (0.01)						
	FOSSA	0.42 (0.03)	4.68 (0.17)	1.00	0.34	0.983	0.000	6.1
	Boosting	0.55 (0.02)	12.95 (0.46)	1.00	0.01	0.897	0.000	202.1
	LASSO	0.59 (0.03)	11.16 (0.41)	1.00	0.00	0.916	0.000	
	FR	19.6 (2.47)	13.15 (0.63)	0.97	0.03	0.895	0.010	
	ISIS-SCAD	0.66 (0.04)	4.82 (0.06)	1.00	0.09	0.981	0.000	
	SIS-SCAD	0.71 (0.10)	5.26 (0.26)	0.97	0.37	0.976	0.010	
$p=500$	Oracle	0.12 (0.01)						
	FOSSA	0.62 (0.03)	4.87 (0.25)	0.96	0.35	0.996	0.017	6.1
	Boosting	0.84 (0.03)	19.31 (0.74)	1.00	0.00	0.967	0.000	181.8
	LASSO	0.97 (0.04)	14.03 (0.56)	1.00	0.00	0.978	0.000	
	FR	94.9 (4.04)	18.95 (0.21)	0.93	0.00	0.968	0.023	

Censoring Rate=50%								
<i>p</i>	Method	MSE	Size	Cover	Exact	Corr0	Incorr0	Iter
<i>p</i> =1000	ISIS-SCAD	0.86 (0.05)	5.00 (0.00)	1.00	0.00	0.996	0.000	
	SIS-SCAD	1.14 (0.10)	7.69 (0.32)	0.91	0.10	0.990	0.033	
	Oracle	0.12 (0.01)						
	FOSSA	0.60 (0.03)	6.11 (0.35)	0.97	0.25	0.997	0.013	7.3
	Boosting	0.86 (0.03)	22.73 (1.08)	1.00	0.00	0.980	0.000	198.9
	LASSO	1.01 (0.04)	14.57 (0.65)	0.98	0.00	0.988	0.007	
	FR	99.0 (3.17)	17.97 (0.24)	0.93	0.00	0.985	0.027	
	ISIS-SCAD	1.08 (0.07)	5.00 (0.00)	0.99	0.00	0.998	0.003	
	SIS-SCAD	1.55 (0.12)	8.73 (0.29)	0.94	0.08	0.994	0.030	
Censoring Rate=75%								
<i>p</i>	Method	MSE	Size	Cover	Exact	Corr0	Incorr0	Iter
<i>p</i> =100	Oracle	0.33 (0.04)						
	FOSSA	0.93 (0.06)	4.12 (0.19)	0.73	0.24	0.985	0.110	5.3
	Boosting	1.06 (0.06)	10.94 (0.42)	0.93	0.00	0.917	0.023	256.7
	LASSO	1.14 (0.06)	10.13 (0.46)	0.92	0.00	0.926	0.027	
	FR	75.61 (5.05)	12.78 (0.40)	0.74	0.01	0.896	0.090	
	ISIS-SCAD	1.94 (0.15)	4.92 (0.04)	0.83	0.02	0.978	0.063	
	SIS-SCAD	3.94 (0.54)	6.55 (0.33)	0.71	0.13	0.960	0.106	
<i>p</i> =500	Oracle	0.25 (0.03)						
	FOSSA	1.47 (0.07)	5.46 (0.36)	0.52	0.05	0.994	0.217	6.5
	Boosting	1.64 (0.07)	15.76 (0.79)	0.81	0.01	0.974	0.073	239.7
	LASSO	1.67 (0.07)	14.21 (0.67)	0.77	0.00	0.977	0.087	
	FR	88.13 (3.18)	10.90 (0.20)	0.37	0.00	0.982	0.283	
	ISIS-SCAD	2.99 (0.18)	5.00 (0.00)	0.60	0.00	0.995	0.157	
	SIS-SCAD	7.94 (0.72)	9.37 (0.29)	0.44	0.01	0.986	0.222	
<i>p</i> =1000	Oracle	0.39 (0.07)						
	FOSSA	1.42 (0.08)	7.03 (0.42)	0.60	0.08	0.995	0.163	8.1
	Boosting	1.62 (0.07)	17.23 (0.83)	0.80	0.00	0.985	0.080	231.6
	LASSO	1.76 (0.07)	12.35 (0.62)	0.72	0.00	0.990	0.110	
	FR	88.49 (3.21)	9.95 (0.19)	0.33	0.00	0.992	0.283	
	ISIS-SCAD	4.38 (0.61)	5.00 (0.00)	0.63	0.00	0.998	0.145	
	SIS-SCAD	9.78 (0.95)	9.93 (0.29)	0.42	0.00	0.992	0.265	

Table 3

Simulation results for a quadratic Cox model with more terms than the sample size. Results are averaged over 100 replications. Numbers in parentheses are standard errors.

$\tilde{p}$	Method	MSE	Size	Cover	Exact	Corr 0	Incorr 0	Iter
230	ORACLE	0.17 (0.02)						
	FOSSA	1.74 (0.08)	9.71 (0.39)	0.70	0.01	0.977	0.110	13.1
	Boosting	1.65 (0.08)	21.62 (0.73)	0.81	0.00	0.925	0.056	222.3
	LASSO	1.93 (0.08)	17.01 (0.62)	0.73	0.00	0.945	0.086	
	FR	14.47 (0.80)	19.12 (0.33)	0.90	0.00	0.937	0.024	
	ISIS-SCAD	1.53 (0.11)	5.00 (0.00)	0.27	0.27	0.995	0.235	
	SIS-SCAD	2.03 (0.11)	8.03 (0.26)	0.14	0.04	0.981	0.260	
495	ORACLE	0.14 (0.01)						
	FOSSA	2.01 (0.07)	11.45 (0.49)	0.64	0.02	0.986	0.116	14.3
	Boosting	2.05 (0.08)	25.27 (1.01)	0.74	0.00	0.958	0.070	190.7
	LASSO	2.50 (0.09)	16.81 (0.74)	0.54	0.00	0.974	0.162	
	FR	27.89 (1.65)	20.93 (0.06)	0.90	0.00	0.967	0.034	
	ISIS-SCAD	1.85 (0.14)	5.00 (0.00)	0.27	0.27	0.997	0.300	
	SIS-SCAD	2.56 (0.10)	9.55 (0.32)	0.06	0.01	0.988	0.313	

Table 4

Genes selected by FOSSA for the lymphoma data and the breast cancer data. Reported are the order of selection, the gene ID, the estimated coefficient by FOSSA, the estimated coefficient by the Cox model with only the selected genes and the corresponding p -values.

Diffuse large-B-cell lymphoma dataset				
Order	Gene ID	$\hat{\beta}$ (FOSSA)	$\hat{\beta}$ (Cox)	p -value (Cox)
1	AA805575	-0.134	-0.105	2.31e-02
2	AA262133	0.764	1.497	4.01e-08
3	W46566	-0.276	-0.320	2.78e-02
4	AA830781	0.259	0.323	4.47e-02
5	AA243583	-0.274	-0.644	5.42e-06
6	AA293559	-0.106	-0.258	5.66e-03
7	LC_32424	0.176	0.544	1.36e-02
8	AA721746	0.102	0.148	3.27e-01
9	N48691	-0.078	-0.383	1.04e-01
10	AI219836	0.027	0.387	1.37e-02
11	AI391470	0.009	0.222	1.16e-01

Breast cancer dataset				
Order	Gene ID	$\hat{\beta}$ (FOSSA)	$\hat{\beta}$ (Cox)	p -value (Cox)
1	NM_006607	2.502	2.741	1.69e-05
2	AL110226	1.516	2.562	2.43e-05
3	Contig58368_RC	0.788	1.136	2.73e-02
4	NM_002811	1.153	2.760	8.37e-04
5	NM_006399	-0.366	-1.410	1.11e-02
6	NM_006054	0.366	2.263	4.01e-03
7	NM_013290	0.159	1.304	2.35e-02