ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/227369280

High-Dimensional Variable Selection for Survival Data

Article *in* Journal of the American Statistical Association · March 2010

DOI: 10.1198/jasa.2009.tm08622 · Source: RePEc

CITATIONS
267

READS
1,870

5 authors, including:

Udaya B Kogalur
Cleveland Clinic
13 PUBLICATIONS **2,321** CITATIONS
SEE PROFILE

Eiran Z Gorodeski
Cleveland Clinic
79 PUBLICATIONS **2,065** CITATIONS
SEE PROFILE

Andy Minn
University of Pennsylvania
103 PUBLICATIONS **19,419** CITATIONS
SEE PROFILE

Michael S Lauer
National Institutes of Health
455 PUBLICATIONS **38,069** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project miR-218 opposes an important RTK-HIF signaling pathway in Glioblastoma View project

All content following this page was uploaded by Michael S Lauer on 27 May 2014.

The user has requested enhancement of the downloaded file.

High-Dimensional Variable Selection for Survival Data

Hemant Ishwaran, Udaya B. Kogalur, Eiran Z. Gorodeski, Andy J. Minn, and Michael S. Lauer

The minimal depth of a maximal subtree is a dimensionless order statistic measuring the predictiveness of a variable in a survival tree. We derive the distribution of the minimal depth and use it for high-dimensional variable selection using random survival forests. In big p and small n problems (where p is the dimension and n is the sample size), the distribution of the minimal depth reveals a "ceiling effect" in which a tree simply cannot be grown deep enough to properly identify predictive variables. Motivated by this limitation, we develop a new regularized algorithm, termed RSF-Variable Hunting. This algorithm exploits maximal subtrees for effective variable selection under such scenarios. Several applications are presented demonstrating the methodology, including the problem of gene selection using microarray data. In this work we focus only on survival settings, although our methodology also applies to other random forests applications, including regression and classification settings. All examples presented here use the R-software package randomSurvivalForest.

KEY WORDS: Forest; Maximal subtree; Minimal depth; Random survival forest; Tree; VIMP.

1. INTRODUCTION

Motivated by challenging problems arising in modern biology, high-dimensional variable selection has become one of the hottest topics in statistics. High-dimensional survival analysis in particular has attracted much interest due to the popularity of microarray studies involving survival data. Traditionally, microarrays have been used to find gene expression values that predict phenotype, but a new emphasis is on finding genes predictive of survival. This is statistically challenging because the number of genes, p, is typically hundreds of times larger than the number of microarray samples, n. At the same time one has to contend with the nuances of survival data, such as dealing with right-censoring and deciding what (if any) distributional assumptions to make when modeling the data.

The most popular approaches by far are those based on Cox regression. Proposed methods include partial least squares (Nguyen and Rocke 2002; Li and Gui 2004), Cox regression under lasso-type penalization (Park and Hastie 2007a; Zhang and Lu 2007), and boosting using Cox-gradient descent (Li and Luan 2006; Ma and Huang 2006) (see Ridgeway 1999 for the first instance of boosting for Cox models). Some methods implicitly use Cox regression by making use of Cox scores; for example, "corrected" Cox scores were used by Bair and Tibshirani (2004) for semisupervised prediction using principal component regression and by Tibshirani et al. (2002) for semisupervised classification using nearest-neighbor shrunken centroid clustering.

Not all research has focused on Cox regression modeling, however. For example, Ma, Kosorok, and Fine (2006) proposed an additive risk model, Huang, Ma, and Xie (2006) considered lasso and gradient-directed regularization for an accelerated failure time model, Hothorn and Buhlmann (2006) described a general L_2-boosting procedure for right-censored survival data, and Clarke and West (2008) used a Bayesian model-averaging tree-based approach.

In this article we consider the general question of how to select variables in high-dimensional survival settings. Our approach is based on random survival forests (RSF), a new extension of Breiman's random forests (RF) (Breiman 2001a) to right-censored survival settings (Ishwaran et al. 2008). Similar to RF, trees in a survival forest are grown randomly using a two-step randomization process. First, each is grown using an independent bootstrap sample. Then, during the tree-growing process, a random subset of variables of size $m \le p$ is selected at each node, and the node is split using these candidate variables.

RSF enjoys all of the important properties of Breiman's RF and thus is well suited for high-dimensional variable selection. It adaptively discovers nonlinear effects and interactions and is fully nonparametric. Averaging over trees, and randomizing while growing a tree, enables RSF to approximate complex survival functions while maintaining low prediction error. Recently, Ishwaran and Kogalur (2008a) showed that RSF is uniformly consistent and that survival forests have a uniform approximating property in finite-sample settings—a property not possessed by survival trees. Note that RSF differs from the RF approach of Hothorn et al. (2006) and the conditional inference method described by Hothorn, Hornik, and Zeileis (2006) (see Ishwaran et al. 2008 for a discussion of differences among the various methods).

1.1 Contributions and Outline of Article

Although RF maintains good prediction performance even with a large number of variables, as dimension increases, some form of regularization is needed to ensure good variable selection properties.

The need to regularize forests has been recognized in the bioinformatics literature. For example, Diaz-Uriarte and Alvares (2006) described a stepwise procedure using RF for selecting genes from microarray data. Genes are ordered on the basis of variable importance (VIMP) (Sec. 2) and then removed from least informative to most informative until prediction error stabilizes. Using backward elimination, noise is systematically removed, and prediction error for refitted forests improves—a type of regularization.

Hemant Ishwaran is Staff (E-mail: *hemant.ishwaran@gmail.com*) and Udaya B. Kogalur is Adjunct Staff, Department of Quantitative Health Sciences, Cleveland Clinic, Cleveland, OH 44195. Eiran Z. Gorodeski is Fellow, Heart and Vascular Institute, Cleveland Clinic, Cleveland, OH 44195. Andy J. Minn is Assistant Professor, Department of Radiation Oncology, University of Pennsylvania, Philadelphia, PA 19104. Michael S. Lauer is Director, Division of Cardiovascular Sciences, National Heart, Lung, and Blood Institute, Bethesda, MD 20892. This work was partially funded by the National Heart, Lung, and Blood Institute (CAN 8324207) and by Department of Defense Breast Cancer Research Program Era of Hope Scholar Award (BC085325).

Journal of the American Statistical Association
March 2010, Vol. 105, No. 489, Theory and Methods
DOI: 10.1198/jasa.2009.tm08622

The variable selection method of Diaz-Uriarte and Alvares (2006) is just one example from the growing list of RF methods based on VIMP. But while VIMP is a useful idea, several limitations hamper the ability to develop a general methodology based on it: (1) VIMP is intimately tied to the type of prediction error used; (2) it seems impenetrable to detailed theoretical study (see Ishwaran 2007 for some attempts); and (3) regularization is crucial to success in high dimensions, but developing formal regularization methods based on VIMP seems impossible.

To address these issues, we take a different approach, using concepts more germane to the structure of trees. A key concept introduced in Section 2 is the *maximal subtree*. Related to this is a type of order statistic for trees that we call the *minimal depth*. We motivate the idea of a maximal subtree and discuss the minimal depth statistic, indicating why it represents a key quantity for assessing predictiveness of a variable. We derive its exact distribution (Thm. 1 in Sec. 2) and show how to use this distribution to select variables in settings where p is big, but does not dominate n. Section 3 looks at one such example. There we consider a large cohort of patients considered by their physicians to be at risk for cardiovascular disease. The patients were all referred for electrocardiography and exercise treadmill testing, two of the most common noninvasive diagnostic tests used to evaluate cardiovascular risk.

For the big p and small n problem (Sec. 6), careful analysis reveals a subtle relationship involving p, the depth of a tree, and the right tail of the distribution of the minimal depth. This shows that when the underlying model is sparse, it is impossible to grow a tree deep enough to properly select variables. (In Secs. 2 and 6 we discuss the required size of p relative to n for this to hold.) This motivates our new regularization algorithm, RSF-Variable Hunting (RSF-VH). We test RSF-VH on a collection of benchmark microarray data sets and compare its performance with that of several well-known methods (Sec. 6). Section 7 summarizes our main findings and discusses implications of our work to other settings, such as regression and classification problems.

2. MAXIMAL SUBTREES

2.1 Variable Importance

VIMP equals the amount that prediction error increases (or decreases) if a variable is noised up when predicting on test data (Breiman 2001a). Combined with the extremely adaptive nature of forests, VIMP has been found to be effective in many applied settings for filtering variables (Breiman 2001b; Lunetta et al. 2004; Bureau et al. 2005; Diaz-Uriarte and Alvares 2006; Weichselbaum et al. 2008; Ishwaran et al. 2009).

In Breiman's original definition, VIMP is calculated by permuting a variable (i.e., noising it up) and then calculating the change in prediction error (Breiman 2001a). A more effective method, which we use throughout this article, is to assign terminal node membership by random node assignment (Ishwaran 2007; Ishwaran et al. 2008). Random node assignment for a variable v works as follows. Cases (data) are dropped down a tree, and each case as it travels down the tree is randomly assigned to a daughter node whenever its parent node splits on v. The predictor for the noised-up data is calculated for each tree and then averaged over the forest. The VIMP for v is the prediction error for the original forest predictor subtracted from the prediction error for the noised-up forest predictor. A positive VIMP occurs if prediction error increases under noising up. Thus positive VIMP values, especially large ones, indicate predictive variables. Typically, VIMP is calculated by growing trees on bagged data and using out-of-bag (OOB) data for validation (Breiman 2001a); however, cross-validation methods also can be used.

2.2 Peering Inside the Black Box

With forests, one often finds variables tending to split close to the root node have a strong effect on prediction accuracy—and thus a strong effect on VIMP. Test data classified by random daughter assignments lead to poor prediction in such cases, because terminal node assignments will be distant from their original values. In contrast, variables that split farther down the tree have much less impact because terminal node assignments are not as perturbed.

These ideas were formalized by Ishwaran (2007) using a new tree concept called a maximal v-subtree. This structure allows quantification of the importance of a variable through its positioning in a tree. We recall this definition (see Figure 1 for an illustration).

Definition. For each variable v, call T_v a v-subtree of T if the root node of T_v is split using v. Call T_v a maximal v-subtree if T_v is not a subtree of a larger v-subtree.

Maximal subtrees provide a powerful way to explore forests. Just like VIMP, they can be used to quantify a variable's predictiveness; the closer a variable's maximal subtree is to the root node, the greater the variable's impact on prediction, and the more informative it is. Although it is possible in rare instances for a nonpredictive variable to split high in a tree and not impact prediction, such occurrences are rare in a large forest of trees, and their effects are minimized when averaging. Maximal subtrees also can be used to identify variable interactions. Interrelationships can be explored using what we call second-order

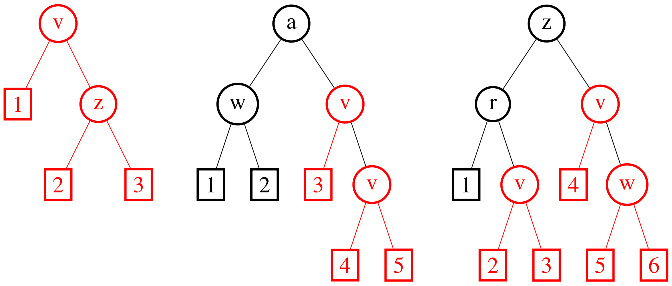

Figure 1. Illustration of maximal v-subtrees. In the first tree, v splits the root node, and the maximal v-subtree is the entire tree. In the second tree, the maximal v-subtree is the v-subtree with terminal nodes 3, 4, and 5 (contained within this is a smaller v-subtree with terminal nodes 4 and 5, but this subtree is not maximal). In the third tree, there are two maximal v-subtrees. The maximal subtree on the left side is the v-subtree with terminal nodes 2 and 3; that on the right side is the v-subtree with terminal nodes 4, 5, and 6. All maximal subtrees are highlighted in red. Letters in parent nodes (circles) identify the variable used to split the node.

maximal subtrees. A second-order maximal (w, v)-subtree is a maximal w-subtree within a maximal v-subtree for a variable v. By considering those variables with closest maximal subtrees to the root node of a maximal v-subtree, potential interactions with v can be identified.

In essence, the maximal subtree is a core concept that allows us to study forests and peer inside what is often thought of as a black box. VIMP is a powerful tool for analyzing forests, but there are compelling reasons suggesting that maximal subtrees can be used in place of (or in addition to) VIMP:

1. Maximal subtrees and their statistics are dimensionless and free of the specific measure of prediction error. Currently, there is much debate in the survival literature about what constitutes an appropriate measure of prediction performance. Removing the dependence on prediction error focuses issues on more fundamental tree concepts, such as splitting rules.
2. Although we focus on survival forests, maximal subtrees naturally apply to all forests. It is a tree concept independent of outcome. Thus our methodology automatically applies to popular applications like random forest regression (RF-R) and random forest classification (RF-C).
3. Unlike VIMP, which is a randomization procedure, maximal subtrees can be studied in detail.

The last point is especially important. In the next section we derive the exact distribution for the first-order statistic for a maximal subtree, what we call the minimal depth. This will lead to a new approach to high-dimensional variable selection.

2.3 Minimal Depth of a Maximal Subtree

Let D_v be the distance from the root node to the root of the closest maximal v-subtree for a given v. Then D_v is a nonnegative random variable taking values $\{0, \ldots, D(T)\}$, where $D(T)$ is the depth of T (the distance from the root to the farthest terminal node). We call D_v the minimal depth of v. It measures how far a case travels down T before encountering the first split on v, and indicates the predictiveness of v. The smaller the minimal depth, the greater the impact v has on prediction. If $D_v = 0$, then v splits the root node, and the maximal v-subtree is T itself. If $D_v = 1$, then the root node is split using a variable other than v, but the right or left (or both) daughters of the root node are split using v; thus one (or both) of these daughters is a maximal v-subtree. In general, if $D_v = d$, then v splits for the first time at depth d, and at least one of the ℓ_d nodes of depth d is a maximal v-subtree. Figure 2 illustrates these ideas.

To derive the distribution for D_v, we introduce the following notation. Recall that each tree in a random forest is grown by randomly selecting $m \le p$ candidate variables for splitting each node. Let $\pi_{v,j}(t)$ be the probability that v is selected as a candidate variable for splitting a node t of depth j, assuming that no maximal v-subtree exists at depth less than j. Let $\theta_{v,j}(t)$ be the probability that v splits a node t of depth j given that v is a candidate variable for splitting t and that no maximal v-subtree exists at depth less than j. In the next result, we assume that the depth of the tree, $D(T) \ge 1$, is fixed beforehand and that $\ell_d = 2^d$ for each d (i.e., the tree is "balanced").

Theorem 1. Assume that $\pi_{v,j}(t)$ and $\theta_{v,j}(t)$ depend only on the depth of the node t and not on the node t itself. Then

$$\mathbb{P}\{D_v = d\} = \left[\prod_{j=0}^{d-1}(1-\pi_{v,j}\theta_{v,j})^{\ell_j}\right][1-(1-\pi_{v,d}\theta_{v,d})^{\ell_d}], \qquad 0 \le d \le D(T)-1, \quad (1)$$

where $\pi_{v,j} := \pi_{v,j}(t)$ and $\theta_{v,j} := \theta_{v,j}(t)$.

A curious feature of Theorem 1 is that although the sum of probabilities over d is bounded between 0 and 1, there is no guarantee that this sum equals 1, because it is possible for no maximal v-subtree to exist. In such settings, D_v is set to the depth of the tree, $D(T)$. By convention, we normalize the probabilities by

$$\mathbb{P}\{D_v = D(T)\} = 1 - \sum_{d=0}^{D(T)-1} \mathbb{P}\{D_v = d\}. \qquad (2)$$

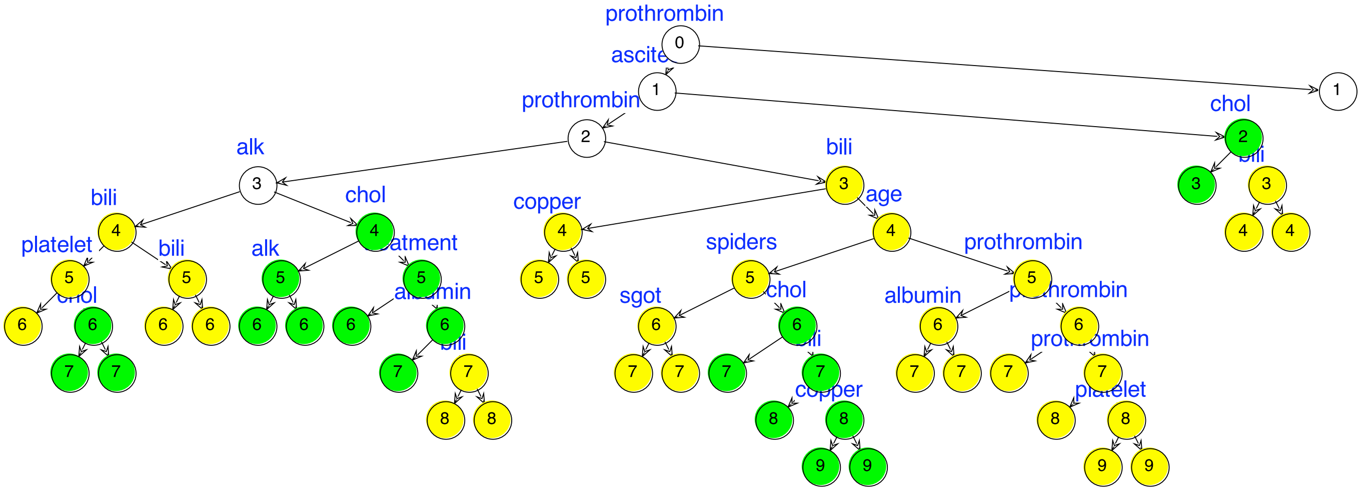

Figure 2. Illustration of minimal depth (based on PBC data described in Sec. 5). The yellow and green points are maximal subtrees for variables "bili" and "chol," respectively. Depth, d, of the tree is indicated by numbers $0, 1, \ldots, 9$ inside of each node [here $D(T) = 9$]. The minimal depth, D_v, for bili is 3 and chol is 2. Note that the tree is unbalanced, for example, $\ell_2 = 2$ and $\ell_3 = 4$ (where ℓ_d is the number of nodes of depth d). Unbalanced trees are discussed in Section 3.

For example, if $D(T) = 1$, which corresponds to a tree having only one split, then $\mathbb{P}\{D_v = 0\} = \pi_{v,0}\theta_{v,0}$ and $\mathbb{P}\{D_v = 1\} = 1 - \pi_{v,0}\theta_{v,0}$.

The assumption of a balanced tree in Theorem 1 is not essential to establishing a closed-form representation for the distribution of the minimal depth. An analog of (1) holds in general for unbalanced trees; however, it requires modification of the conditions for $\pi_{v,j}(t)$ and $\theta_{v,j}(t)$. Theorem 1 assumes that these values are independent of t, but this fails to hold for unbalanced trees. For an illustration, consider Figure 2. Let t be the right daughter node for the root node (the node on the extreme right with depth $d = 1$). Then, because t is a terminal node, $\theta_{v,1}(t) = 0$ for all v.

To accommodate terminal nodes appearing at different depths, we must allow $\theta_{v,j}(t)$ to depend on t. We assume that if t is a node of depth j, then $\theta_{v,j}(t) = 0$ for all v if t is a terminal node; otherwise $\theta_{v,j}(t) = \theta_{v,j}$ is independent of t. Conditioning on the number of nodes $\ell_j = \ell_j^*$, for $j = 0, 1, \ldots, D(T) - 1$, and assuming that $\pi_{v,j}(t) = \pi_{v,j}$, we have the following extension to Theorem 1 that holds for all unbalanced trees:

$$\mathbb{P}\{D_v = d \mid \ell_0^*, \ldots, \ell_{D(T)-1}^*\} = \left[\prod_{j=0}^{d-1} (1 - \pi_{v,j}\theta_{v,j})^{\ell_j^*}\right]\left[1 - (1 - \pi_{v,d}\theta_{v,d})^{\ell_d^*}\right]. \quad (3)$$

Unbalanced trees are discussed further in Section 3.

2.4 High-Dimensional Sparse Settings: Minimal Depth for Weak Variables

In this section we show that for weak variables in high-dimensional sparse settings, $\pi_{v,j}(t)$ and $\theta_{v,j}(t)$ have approximations that satisfy the conditions of Theorem 1. Using this, we obtain a simple closed-form expression for the minimal depth under the null that a variable is noninformative. We use this null distribution to select variables in high dimensions.

First, consider $\pi_{v,j}(t)$. This equals 1 minus the probability that v is not selected as a candidate variable for t, given v has not been split on yet. If p is large, then it is clear that $\pi_{v,j}(t)$ can be approximated by

$$\pi_{v,j} = 1 - \prod_{k=0}^{m-1}\left(1 - \frac{1}{p-k}\right) \approx \frac{m}{p}. \quad (4)$$

This approximation is independent of the depth d and the node t and holds if $m/p = o(1)$. In our applications, $m = \sqrt{p}$.

To estimate $\theta_{v,j}(t)$, let $0 < \tau_0 < 1$ be the fraction of strongly informative variables. We assume that the probability a strong variable is used to split a node is proportional to $W \geq 1$ relative to a weak variable. The approximation (4) shows that the m candidate variables used to split a node can be assumed to be randomly selected from the p variables; therefore, each node has approximately $m\tau_0$ strong variables and $m(1 - \tau_0)$ weak variables. Assuming a sparse setting in which $\tau_0 \ll 1$, then, if v is a weak variable, $\theta_{v,j}(t)$ can be approximated by

$$\theta_{v,j} \approx \frac{1}{(1 - \tau_0)m + W\tau_0 m} = \frac{1}{m}(1 - \tau_0(W - 1) + o(W\tau_0)) \approx \frac{1}{m}. \quad (5)$$

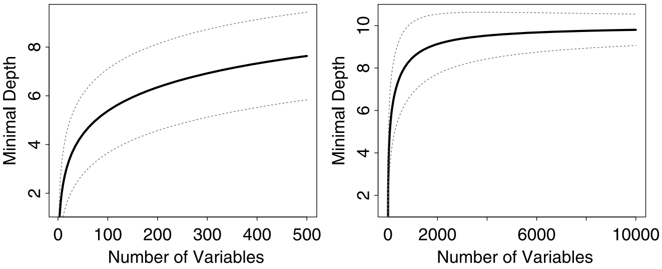

Figure 3. Mean ± standard deviation for minimal depth, D_v, under the null hypothesis that a variable is weak (6) assuming a tree with depth $D(T) = 10$. The thick line is the mean.

By (4) and (5),

$$\pi_{v,j}\theta_{v,j} \approx \frac{m}{p} \times \frac{1}{m} = \frac{1}{p}.$$

Because all probabilities are independent of t, deduce by Theorem 1 that

$$\mathbb{P}\{D_v = d \mid v \text{ is a weak variable}\} \approx \left(1 - \frac{1}{p}\right)^{L_d}\left[1 - \left(1 - \frac{1}{p}\right)^{\ell_d}\right], \quad (6)$$

where $L_d = 1 + 2 + \cdots + 2^{d-1} = \ell_d - 1$.

Figure 3 shows how the mean and standard deviation for D_v under (6) varies as a function of p for a tree of depth $D(T) = 10$. The mean increases as a function of p, but the increase is slow. When p is 500, the mean for D_v is roughly 7, and when p is as large as 10,000, the mean is roughly 9. The mean for D_v signifies a threshold value for identifying strong variables, and thus presents a method for selecting variables in high dimensions. This threshold should be robust given that it increases slowly with p.

Note, importantly, that the asymptote on the right side of Figure 3 arises as p becomes larger than $\ell_{D(T)}$. If $p \gg \ell_{D(T)}$, then (6) is of order

$$\left[1 - \frac{\ell_d - 1}{p} + o(\ell_d/p)\right]\left[\frac{\ell_d}{p} + o(\ell_d/p)\right] = \frac{\ell_d}{p}\left(1 - \frac{\ell_d - 1}{p}\right) + o(\ell_d/p),$$

and all probabilities are near 0. Therefore, D_v becomes degenerate at $D(T)$, because $\mathbb{P}\{D_v = D(T)\} \approx 1$ due to the normalizing constraint (2). This has important implications for big p and small n problems. If p is too large relative to n, then it may not be possible to grow a tree deep enough to properly use (6) to threshold variables. We return to this issue in Section 6.

2.5 Accuracy of Approximations

Here we provide further justification for the approximations (4) and (5) used to establish (6). Although our previous arguments specifically assumed a sparse high-dimensional setting, we show that the approximations may still be valid otherwise.

The accuracy of (4) depends on the number of variables available for splitting a node. If p variables are available, then the left side of (4) is exact, and the approximation is highly accurate, even if p is very small. On the other hand, because variables can get “used up” during the tree-growing process, not all variables will be available at each node. For example, if v' is a binary variable that splits the root node, then v' cannot split further nodes. When one or more variables get used up, $\pi_{v,j}$ becomes

$$\pi_{v,j} = 1 - \prod_{k=0}^{m-1}\left(1 - \frac{1}{p_{j,t} - k}\right),$$

where $p_{j,t} < p$ is the number of variables available for splitting a node t. The accuracy of (4) suffers if $p_{j,t}/p \ll 1$, but because it is difficult for a tree to use up a sizeable fraction of its variables, it is unlikely that $p_{j,t}$ will differ greatly from p. Furthermore, any serious disparity between $p_{j,t}$ and p is likely to occur near the bottom of the tree, and because the distribution of D_v has little mass when the depth is large, the effect is minimized.

There is another way to motivate (6) without the assumption of high-dimensional sparsity. Note that because (5) is independent of the node and the depth of the node, (5) says that the behavior of a weak variable mimics a random coin-tossing experiment. When combined with (4), this implies that the number of splits, S_v, for v is a binomial random variable

$$(S_v | v \text{ is a weak variable}) \sim \text{Binomial}\left(S_T, \frac{1}{p}\right), \qquad (7)$$

where S_T is the number of splits in a tree (if T is balanced, then $S_T = L_{D(T)}$).

Thus if the splitting behavior for a weak variable is such that splits are roughly independent and the number of splits within a tree is approximately Poisson-distributed with mean S_T/p, then (6) should hold. Importantly, this does not presume high-dimensional sparsity, nor is there any reason to believe that such an assumption is required for the Poisson property to hold. In fact, in Section 4 we show that our method performs excellently even in low-dimensional problems.

The Poisson behavior can be seen in a real example. Consider Figure 4, which shows the tree relative frequency of a variable being split, S_v/S_T, averaged over each tree T in a survival forest. The figure was derived from a RSF analysis used in Section 3. Note how most of the variables have relative frequencies near $1/p$, the mean value of S_v under (7) (see the thick dashed line). If the theory underlying (7) is correct, then these must be weak variables; there is strong evidence to suggest this is the case. Our analysis in Section 3 demonstrates that many of the variables in this data are weakly informative and that only a handful have a strong signal.

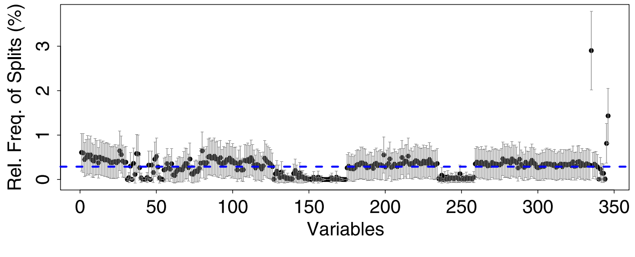

Figure 4. Relative frequency (in percent) that a variable is split (based on RSF analysis of electrocardiogram data; see Sec. 3). Gray bars are standard deviations. The dashed line is $1/p$ (in percent), the mean value under the null hypothesis that a variable is weak. A color version of this figure is available in the electronic version of this article.

Interestingly, two other types of variables appear in Figure 4. One group has relative frequencies near 0; for example, there is a cluster of such variables near 150 and 250 on the x-axis. These are near-degenerate variables with near-zero variance that are rarely split on; for example, some had >99% of their data concentrated at one value. They pose no problem because they have large minimal depths—far larger than that predicted by (6)—and are easily identified as being noninformative. The second group are those with large relative frequencies; for example, there are three variables on the extreme right side with especially large values. These are strong variables. We discuss these kinds of variables next.

2.6 Minimal Depth for Strong Variables

Our variable selection procedure is based on the premise that those variables with minimal depth less than the mean under the null must be strong variables. To quantify how accurately strong variables are identified by this method, we derive the distribution of the minimal depth under the alternative hypothesis that a variable is strong.

In deriving this distribution, we make the following assumptions:

1. The tree is balanced.
2. If v is a strong variable, then $\pi_{v,j} = m/p$.
3. The split for a node is always at the median of the value being split. Thus if there are N cases in a node, then $N/2$ cases are assigned to the left daughter node, and $N/2$ cases are assigned to the right daughter node.
4. If v is a strong variable, then $m\theta_{v,j} = \min(W\sqrt{n2^{-j}}, m)$.
5. $m = \sqrt{p}$.

Assumptions 1, 2, and 5 are the same as before. Assumption 1 could be removed similar to what was done in the extension of Theorem 1 to (3). Assumption 3 is for convenience and allows us to write out a closed-form expresssion for the distribution. This assumption is unrealistic in practice, but weakening it will not change the message that we are trying to convey. Assumption 4 says that if v is a strong variable and a candidate for splitting a node, then the probability that v splits the node equals W/m times the square root of the sample size of the node, $N = n2^{-j}$. (The size of the node is due to assumption 3.) This is realistic, because we would expect any good splitting rule to have a $\sqrt{N}$-asymptotic property.

Under assumptions 1–5, and invoking Theorem 1, it follows that

$$\mathbb{P}\{D_v = d | v \text{ is a strong variable}\} = \left(1 - \frac{W_d}{p}\right)^{L_d}\left[1 - \left(1 - \frac{W_d}{p}\right)^{\ell_d}\right], \qquad (8)$$

where $W_d = \min(W\sqrt{n2^{-d}}, \sqrt{p})$.

Figure 5 compares the mean of D_v under (8) to that under (6) for a tree of depth $D(T) = 10$, for $n = 2^{D(T)}$ and $W = 1$ (see the black lines in the figure). It is apparent that the mean minimal depth for a strong variable is substantially smaller than a weak variable as long as p is not too large; say $p < 1000$. In this

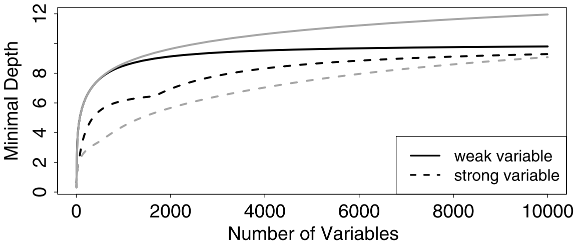

Figure 5. The black lines show mean of the minimal depth for weak and strong variables assuming a tree with depth $D(T) = 10$. For small p, strong variables have smaller minimal depths, but as p increases, the minimal depth converges to $D(T)$ for both types of variables. The gray lines are for a tree with depth $D(T) = 15$.

range, minimal depth thresholding is highly effective; however, as p increases, the tree becomes overwhelmed with variables, and eventually the distribution under both the null and alternative hypotheses degenerates to $D(T)$, and variable selection is no longer effective. This is the big p small n problem that we discuss in Section 6. At the same time, note that if n increases, then the depth of a tree increases, and the limit point in p where the minimal depth converges under the two hypotheses increases [see the gray lines in Figure 5 for $D(T) = 15$]. This shows that if n is very large, then minimal depth thresholding can be effective for very large p.

3. ELECTROCARDIOGRAPHIC ABNORMALITIES AND LONG–TERM SURVIVAL

Cardiovascular disease is the leading cause of death in the developed world. Many patients are asymptomatic or minimally symptomatic for many years before presenting with a life-threatening clinical event, such as a myocardial infarction or sudden cardiac death. Physicians often refer patients deemed to be at increased risk for routine noninvasive diagnostic tests, such as electrocardiography (ECG) and exercise treadmill testing, both of which involve collecting numerous variables. ECG is a vector-based recording of electrical currents within the heart over the course of the cardiac cycle, which involves depolarization and repolarization of the atria (upper chambers) and ventricles (lower chambers). Using digital technology, more than 500 variables are typically recorded, including the amplitudes, durations, and direction of different electrical signals corresponding to atrial and ventricular depolarization and repolarization. In the exercise test, patients walk on a treadmill with gradually increasing speed and grade until exhaustion, which typically takes about 8–12 minutes. One of the strongest predictors of risk is exercise capacity (corresponding to physical fitness). In addition, throughout the test and for several minutes thereafter, detailed data on ECG changes, heart rate, heart rhythm, and symptoms are obtained. Previous investigations have demonstrated that both ECG and exercise testing are powerful predictors of risk in patients with suspected cardiovascular disease.

Our cohort presented a unique opportunity and challenge in that all patients had a qualitatively normal ECG; that is, there were no gross abnormalities. We focused on this cohort because qualitatively normal ECGs are common in patients with suspected cardiovascular disease and because previous investigations have demonstrated that subtle quantitative differences based on computerized measures may be prognostically important. Our cohort comprised 19,530 patients. For each patient, 346 variables comprising both clinical and ECG measurements were recorded. Mean follow-up time was approximately 11 years. A total of 1742 patients died.

A survival forest of 1000 trees was fit to the data. Computations were implemented using the randomSurvivalForest R-package (Ishwaran and Kogalur 2007, 2008b). (All survival forests grown in this work were calculated using this software unless stated otherwise.) The number of candidate variables selected for each node was $m = \sqrt{p}$. For the splitting rule, we used random log-rank splitting with an "nsplit" value of 10. Trees were grown by choosing a maximum of nsplit split points randomly for each candidate variable when splitting a node. (This is in contrast to deterministic splitting, in which all possible split points for each candidate variable are considered.) Log-rank splitting was applied to these random split points, and the node was split using the variable whose random split point maximized the log-rank statistic. Random splitting greatly reduces computation (Ishwaran et al. 2008). Another advantage is that it mitigates tree bias favoring splits for continuous variables and factors with a large number of categorical labels. See Lo and Shih, Lo and Vanichsetakul (1997, 1988) for other approaches to unbiased splitting and for more background on unbiased tree splitting.

Figure 6 shows, for each variable, the forest-averaged minimal depth of the closest maximal subtree (i.e., minimal depth) versus the forest-averaged minimal depth of the second-closest maximal subtree (i.e., second-order depth). We focus on the minimal and second-order depths because these contain the most information; higher-order depths were nearly the same for most variables. A circle's diameter in the plot is proportional to the forest-averaged number of maximal subtrees for a variable. The three variables in the extreme bottom left of the plot have the smallest minimal and second-order depths. These

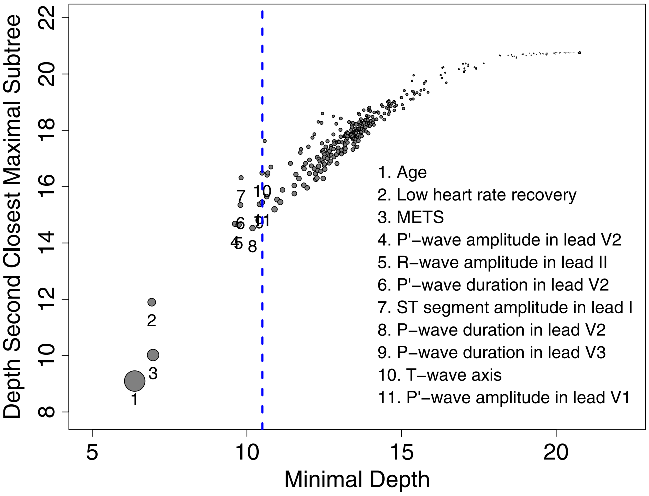

Figure 6. Distance of the closest maximal subtree (minimal depth) versus distance of the second-closest maximal subtree (second-order depth) from RSF analysis of ECG data ($n = 19{,}530$, $p = 346$, OOB error rate 18.6%). A circle's diameter is proportional to the average number of maximal subtrees for a given variable. The dashed line is the mean value of D_v^* and represents a threshold value for filtering variables. A color version of this figure is available in the electronic version of this article.

are by far the most predictive variables. In order, they are age, low heart rate recovery (i.e., slow postexercise attenuation of heart rate), and peak metabolic equivalents (i.e., the proportion of age- and gender-adjusted exercise capacity achieved). All are clinical variables, and all are known to be powerful predictors of cardiovascular disease.

To use the null distribution (6) to determine a threshold value for filtering variables, we had to make an adjustment to allow for unbalanced trees. Recall that (6) assumed a balanced tree (i.e., $\ell_d = 2^d$ for each d), but survival trees in forests are unbalanced, because they are grown to full size with little or no constraint. In randomSurvivalForest, trees are grown only under the constraint that a terminal node has a minimum of "nodesize" unique deaths.

To make this adjustment, in place of D_v we used D_v^*, a random variable with distribution (6), but with node counts ℓ_d replaced by forest-averaged estimates ℓ_d^*. This can be thought of as a conditioning argument similar to (3), but with the conditioning here with respect to ℓ_d^*. To normalize the distribution for D_v^*, let $\overline{D}$ be the average tree depth of the forest. Then $D_v^* \in \{0, 1, \ldots, \overline{D}\}$, and its distribution is normalized by

$$\mathbb{P}\{D_v^* = \overline{D}\} := 1 - \sum_{d=0}^{\overline{D}-1} \mathbb{P}\{D_v^* = d\}.$$

The mean value for D_v^* is indicated by the dashed line in Figure 6. In total, 11 variables (8 ECG variables in addition to the top 3 variables) had a value below this mean. The presence of the ECG variables is interesting because it confirms previous studies suggesting that ECG measurements play a role in long-term survival.

To investigate the effectiveness of our thresholding method, we ordered variables by minimal depth. Using the sorted variables, we considered the nested sequence of models starting with the top variable (smallest minimal depth), followed by the top two variables, then the top three variables, and so on. In each instance, a survival forest with 1000 trees was grown using random log-rank splitting with an nsplit value of 10.

Figure 7 compares the nested models using three different performance measures (all calculated using OOB data). The value C is 1 minus Harrell's C-index (Harrell et al. 1982). This estimates the probability of correctly ranking two individuals in terms of survival. Ranking of individuals was based on the RSF predicted value, defined as the sum of the forest cumulative hazard function, summed over all unique event (death) time points. (This is a predicted value for mortality; see Ishwaran et al. 2008 for further discussion.) CRPS is the continuous ranked probability score and is defined as the area under the prediction error curves (with curves evaluated at each unique event time). R^2 is the explained residual variation, an overall measure of accuracy adjusted relative to the Kaplan–Meier curve; here it is evaluated and averaged over each unique event time. Both CRPS and R^2 are based on the Brier score and were calculated using the pec R-software package (Gerds 2006, 2008). For more details on CRPS and R^2, see Graf et al. (2008).

Table 1. OOB performance measures for ECG data using models of various sizes

Variables	Model size	C	CRPS	R^2
Top 3 variables	3	0.172	0.049	0.158
Top 10 variables	10	0.168	0.048	0.177
All	346	0.185	0.051	0.143

Figure 7 shows that all performance measures improve as model size increases and that the pattern is near monotonic. Recall that models were ordered on the basis of minimal depth, but minimal depth is a quantity independent of the measure of prediction performance. Thus Figure 7 shows that minimal depth is capturing key information regarding a variable's predictiveness.

Figure 7 shows that our top three variables are highly informative. Performance continues to improve beyond these variables, eventually reaching a minimum (or maximum) when the model size is between 5 and 10. As more variables are added, performance eventually degrades. (Table 1 provides comparisons to the full model as an illustration.) The top model comprising 10 variables includes, in addition to the 3 clinical variables, 7 ECG measurements similar to those shown in Figure 6. The overlap in the two analyses is evidence of the effectiveness of minimal-depth thresholding.

4. SIMULATION STUDY OF PERFORMANCE IN LOW–DIMENSIONAL SETTINGS WITH AND WITHOUT CORRELATION

In this section we investigate the performance of our method in low-dimensional settings under different types of correlation. Variable selection was based on thresholding using the mean of D_v^* as in Section 3.

We used simulations to study performance. We set $n = 200$ and $p = 25$ and simulated survival times by drawing independent values from an exponential distribution with mean value $\mu = \exp(\sum_{k=1}^{p} \beta_k x_k)$. Censoring times were drawn independently from an an exponential distribution with mean set to the average of μ over all observations. The p-dimensional covariates $(x_1, \ldots, x_p)^T$ were simulated by drawing independent values from a multivariate normal distribution with mean 0 and

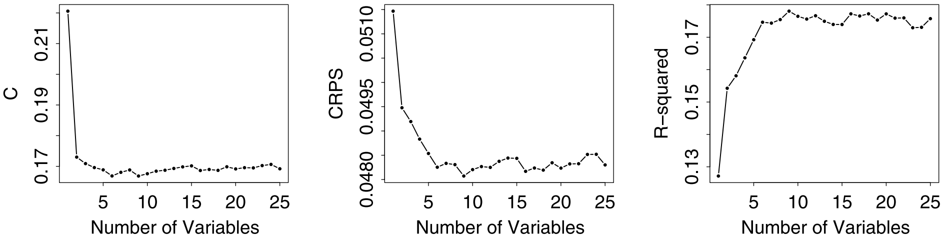

Figure 7. OOB performance measures for ECG data under sequentially fit models. Left to right: C, CRPS, and R^2.

covariance $\mathrm{corr}(x_j, x_k) = \rho^{|j-k|}$. We considered two correlation settings, (a) $\rho = 0$ and (b) $\rho = 0.9$. The regression parameter $\beta = (\beta_1, \ldots, \beta_p)$ was set to 0 everywhere except the five middle coordinates, which were set to the value $b_0 > 0$,

$$\beta = (\underbrace{0, \ldots, 0}_{10}, \underbrace{b_0, \ldots, b_0}_{5}, \underbrace{0, \ldots, 0}_{10}).$$

Five different values for b_0 were used: 0.2, 0.4, 0.6, 0.8, and 1.0.

The simulations were repeated 100 times independently. For each simulation, the false discovery rate (FDR), false nondiscovery rate (FNR), total number of incorrectly classified variables (Miss), and model size ($\hat{p}$) were recorded. The FDR equaled the false discovery rate for those coefficients identified as nonzero by the procedure (i.e., the number of truly zero coefficients identified as nonzero divided by the number of coefficients identified as nonzero). The FNR equaled the false nondiscovery rate for those coefficients identified as zero (i.e., the number of truly nonzero coefficients identified as zero divided by the number of coefficients identified as zero). Miss was defined as the total number of misclassified variables, that is, the total number of falsely identified nonzero coefficients (the numerator of FDR) and falsely identified zero coefficients (the numerator of FNR). Finally, $\hat{p}$ was the number of coefficients selected by the procedure. All quantities were averaged over the 100 simulations. Monte Carlo standard deviations were calculated for each performance measure (Table 2).

Two different Cox regression procedures were used for comparison. Because the data were simulated from a proportional hazards model, this put things on home turf for Cox modeling and made for challenging competition. The first procedure used Cox regression with the adaptive lasso, as described by Zhang and Lu (2007). Five-fold validation was used to tune the regularization parameter (see "Adapt-lasso" in Table 2). The second procedure used l_1-regularized Cox regression with the algorithm described by Park and Hastie (2007a) (see "Cox-path" in Table 2). Computations were implemented using the glmpath R-package (Park and Hastie 2007b). Five-fold validation was used to determine the optimal regularization parameter. For RSF, forests comprised 1000 trees grown under random log-rank splitting with an nsplit value of 10.

The results are impressive. In the uncorrelated variable simulations ($\rho = 0$), low FDR and FNR values are seen for RSF for almost all b_0. As b_0 increases, Miss decreases nearly to zero and $\hat{p}$ converges closely to the true dimension, $p_0 = 5$. Results using regularized Cox regression are also good, but there are noticeable differences. Cox-path overfits when the signal is low, whereas the adaptive lasso tends to overfit regardless of signal.

The results are even better in the correlated simulations ($\rho = 0.9$). RSF excels under all performance measures. Miss decreases to zero and $\hat{p}$ increases to p_0 as b_0 increases. Moreover, performance measures convergence faster than in the uncorrelated setting. For the Cox procedures, Cox-path tends to

Table 2. Low-dimensional simulations ($n = 200$, $p = 25$, and 5 nonzero coefficients). Performance measures are FDR, FNR, Miss (total number of misclassified variables), and $\hat{p}$ (number of coefficients selected by the procedure)

	RSF-minimal depth				Adapt-lasso				Cox-path			
b_0	FDR	FNR	Miss	$\hat{p}$	FDR	FNR	Miss	$\hat{p}$	FDR	FNR	Miss	$\hat{p}$
Uncorrelated variables ($\rho = 0$)												
Averaged values over 100 replications												
0.2	0.18	0.19	4.67	0.53	0.38	0.11	4.63	4.99	0.68	0.05	12.14	16.2
0.4	0.00	0.14	3.23	1.77	0.23	0.01	2.07	6.73	0.19	0.03	2.34	6.28
0.6	0.00	0.08	1.87	3.13	0.18	0.00	1.34	6.32	0.02	0.04	0.94	4.28
0.8	0.00	0.05	1.21	3.79	0.16	0.00	1.18	6.18	0.00	0.05	1.09	3.91
1.0	0.00	0.04	0.83	4.17	0.13	0.00	1.01	6.01	0.00	0.05	1.16	3.84
Monte Carlo standard deviations												
0.2	0.37	0.02	0.64	0.56	0.25	0.06	1.96	3.21	0.12	0.07	5.21	5.74
0.4	0.00	0.03	0.92	0.92	0.19	0.02	1.82	1.95	0.21	0.04	2.63	3.09
0.6	0.00	0.04	0.94	0.94	0.15	0.01	1.37	1.37	0.06	0.03	0.63	0.83
0.8	0.00	0.04	0.87	0.87	0.16	0.00	1.45	1.45	0.00	0.02	0.53	0.53
1.0	0.00	0.04	0.78	0.77	0.15	0.00	1.37	1.37	0.00	0.02	0.59	0.59
Correlated variables ($\rho = 0.9$)												
Averaged values over 100 replications												
0.2	0.01	0.14	3.33	1.71	0.48	0.14	5.47	4.27	0.39	0.09	4.43	5.81
0.4	0.01	0.05	1.11	3.95	0.42	0.11	4.56	5.20	0.09	0.05	1.66	4.42
0.6	0.00	0.02	0.50	4.52	0.35	0.07	3.86	6.02	0.03	0.05	1.32	4.04
0.8	0.00	0.01	0.29	4.73	0.30	0.05	3.00	6.24	0.01	0.06	1.33	3.77
1.0	0.00	0.01	0.16	4.84	0.27	0.03	2.49	6.53	0.01	0.06	1.32	3.78
Monte Carlo standard deviations												
0.2	0.05	0.04	1.15	1.18	0.24	0.03	1.89	2.15	0.22	0.04	2.27	2.28
0.4	0.03	0.04	0.80	0.81	0.19	0.04	1.99	1.77	0.14	0.04	1.07	1.34
0.6	0.02	0.03	0.56	0.54	0.20	0.04	2.19	2.33	0.08	0.03	0.75	0.91
0.8	0.02	0.02	0.48	0.49	0.18	0.04	1.82	1.77	0.04	0.04	0.86	0.89
1.0	1.00	0.02	0.37	0.37	0.18	0.03	1.82	1.76	0.04	0.04	0.88	0.91

underfit as the signal increases. Model sizes for the adaptive lasso are better, but FDR is high.

RSF does so well in the correlated case because of the grouping property of trees. Trees are able to borrow strength from correlation, such that if there is a cluster of correlated variables with a true signal, then a split on a variable from a member of the group will be followed closely by splits by other members of the group. As a result, each member of the group has a small minimal depth, so that the entire group becomes a candidate for selection in the final model. This grouping property is very similar to that exhibited by the elastic net (Zou and Hastie 2005) and is highly desirable when analyzing high-dimensional microarray data. This is because groups of genes, although highly correlated, often represent a biological pathway or process. As one example, prognostic gene signatures for breast cancer are highly correlated with genes involved in cell proliferation (Wirapati et al. 2003); however, unless enough genes directly linked to cell proliferation are selected in a final gene signature, this important biological insight will be lost.

5. PERFORMANCE UNDER HIGH–DIMENSIONAL NOISE

For our next illustration, we consider the well-known primary biliary cirrhosis (PBC) data set (Fleming and Harrington 1991). These data are from a randomized clinical trial, involving 312 individuals, of the effectiveness of the drug D-penicillamine on PBC. The data set contains 17 variables in addition to censoring information and survival times for each individual.

Random noise variables were added to the data. A noise variable was created by randomly selecting one of the original 17 variables and then randomly permuting its value. A total of 500 noise variables were simulated independently; 20% of the data was set aside for testing, and the remainder was used for training. Over the training data, a survival forest of 1000 trees was grown under random log-rank splitting with an nsplit value of 10. To ensure that trees were sufficiently rich, the minimum terminal node size was set to two (i.e., a nodesize of two).

Because of the small sample size and high dimensionality, we modified our previous approach to entertain regularized solution paths built from an initial model. We started with the model comprising variables whose minimal depth was less than the mean of D_v^* (calculated as in Section 3). Variables were ordered by their minimal depth and added sequentially to the model until joint VIMP no longer increased. Joint VIMP was calculated using random node assignment; daughter assignments were random for any parent node that split on any variable in a given group of variables. The point at which joint VIMP no longer increased signified the final model. A forest was refit using these variables. Two points should be emphasized: (a) Starting with variables meeting a minimal-depth criterion ensured that the algorithm started with a good candidate model and did not terminate early with an overly sparse solution, and (b) computing joint VIMP is computationally efficient because it does not require refitting the forest (Ishwaran and Kogalur 2008b).

Performance measures, as in Table 1, were calculated over the test data. The estimated model dimension and percentage of noise variables were also calculated. The experiment was repeated 100 times and the values averaged (Table 3). Table 3 shows that minimal depth thresholding is highly effective (row 1; RSF-minimal depth). All performance measures are better than the full model (i.e., the RSF analysis using all variables; row 3 of Table 3). The average model size was 8.5, and only a negligible fraction of the 500 noise variables were selected ($<0.1\%$). Furthermore, no noise variable appeared consistently; and of the original 17 variables, six appeared frequently (more than 90% of models). This shows minimal depth thresholding yields not only small, but also stable variable lists.

Included in Table 3 for comparison are results from a VIMP-based approach. The method applied was as follows. The data was expanded to include 50 additional noise variables independently simulated from a standard normal distribution. A RSF analysis was applied to the expanded data and the 99th percentile of the distribution of VIMP for the new noise variables was determined. Any variable exceeeding this threshold was selected as being informative, otherwise it was rejected as noise. A forest was refit using the selected variables only. A similar idea was recently described in Docksum, Tang, and Tsui (2008) under the name RFVS; however to avoid conflicting acronyms we refer to the method as RSF-VIMP.

The results are mixed. While the C, CRPS, and R^2 values for RSF-VIMP compare more favorably to RSF-minimal depth, the model size was substantially larger and the percentage of noise variables was higher (1.6%). Also, variable lists were not as stable; only 3 of the original 17 variables appeared more than 90% of the time. Additional experimentation revealed that RSF-VIMP also was sensitive to the selected threshold value; for example, using a 95th percentile threshold gave substantially larger models. The interested reader should consult Docksum, Tang, and Tsui (2008) for a detailed study of the method. It generally was successful, but its performance was enhanced when used in combination with other procedures.

Table 3. Test set performance using PBC data with 500 noise variables. Row 1 is RSF using minimal depth thresholding, row 2 is RSF using VIMP thresholding, row 3 is RSF using all variables, row 4 is cforest (unbiased conditional inference forests, Hothorn, Hornik, and Zeileis 2006) using all variables, and row 5 is gbm (generalized boosted regression models, Ridgeway 1999) using all variables. All values reported are averaged over 100 independent experiments. Ranking of a procedure within a column is indicated using superscripts

Method	Model size	C	CRPS	R^2	% Noise variables
RSF-minimal depth	$8.5^{(1)}$	$0.164^{(2)}$	$0.129^{(1)}$	$0.289^{(1)}$	$0.08^{(1)}$
RSF-VIMP	$16.4^{(2)}$	$0.172^{(4)}$	$0.135^{(2)}$	$0.257^{(2)}$	$1.6^{(2)}$
RSF	$517.0^{(3)}$	$0.165^{(3)}$	$0.159^{(4)}$	$0.140^{(4)}$	$100.0^{(3)}$
cforest	$517.0^{(3)}$	$0.157^{(1)}$	$0.164^{(5)}$	$0.099^{(5)}$	$100.0^{(3)}$
gbm	$517.0^{(3)}$	$0.176^{(5)}$	$0.144^{(3)}$	$0.236^{(3)}$	$100.0^{(3)}$

Also included in Table 3 is the unbiased conditional tree-based approach of Hothorn, Hornik, and Zeileis (2006). In this approach each tree was grown using unbiased splitting, a type of regularization in which a node is split only when the null hypothesis of no association between the response and each candidate variable is rejected. (We used an α-level of 0.10 for the rejection criterion.) Tuning parameters were set as in the RSF analysis. A total of 1000 trees were grown using bootstrap resampling. For each tree, the number of candidate variables, m, used to split a node equaled $\sqrt{p}$. Predicted values also were defined as in the RSF analysis; that is, each tree yielded a Kaplan–Meier survival curve for each individual, which was then averaged over the forest to yield an ensemble survival function. The resulting cumulative hazard function was summed over all unique event times. This was the predicted value for each individual. The results from the analysis are given in Table 3 under the entry "cforest." Computations were implemented using the party R-package (Hothorn, Hornik, and Zeileis 2006).

Cox-gradient descent boosting (Ridgeway 1999) was used for comparison. The analysis used a shrinkage (learning) parameter of 0.01, a tree depth of 5 (the base learner), and 10-fold validation to determine the optimal number of boosting iterations; a maximum of 1000 iterations were used. Using the predicted value (on the log-hazard scale), we computed the Breslow estimator of the baseline hazard function and from this computed the predicted survival function. The results from the analysis are given in Table 3 under the entry "gbm." Computations were implemented using the gbm R-package (Ridgeway 2007).

Overall, RSF-minimal depth performed the best of all procedures. Second best was RSF-VIMP. The procedures that use no thresholding (RSF, cforest, gbm) were clearly at a disadvantage. Even though each uses sophisticated regularization, this was not sufficient to compensate for the high dimensionality.

6. MICROARRAY DATA: DEALING WITH BIG p AND SMALL n SETTINGS

We have already remarked that in high-dimensional settings, there is an interplay between the dimension, p, and the depth of a tree, $D(T)$, that affects the use of (6) for thresholding variables. For proper implementation, p must not dominate $\ell_{D(T)}$; otherwise, trees will be too shallow, and variables will be assigned a default minimal depth of $D(T)$. Using the mean of D_v^* to threshold variables (as we have done) becomes ineffective when this occurs.

This becomes critical when trying to select genes from microarray data. In such settings $p \gg n$, and a tree simply cannot be deep enough to allow proper assessment of a gene's predictiveness. To ensure proper inference, we must reduce the number of genes so that the distribution of D_v is nondegenerate.

To do this, we merely need to ensure that p is of order $\ell_{D(T)}$. Because trees from forests are unbalanced, we approximate $D(T)$ by $\overline{D}$. Thus we propose selecting a number of genes, $P < p$, such that $\log_2(P) = O(\overline{D})$, and then applying the regularization algorithm of Section 5. This process is repeated several times independently. We call this the RSF-VH algorithm. A detailed description of this algorithm is given in the display that follows. Section 6.1 discusses some key points.

Algorithm 1 RSF-VH Algorithm

```
1:  for $b = 1$ to $B$ do
2:      Split the data into test and training data sets.
3:      Select $P < p$ genes. Call this set of genes $\mathcal{G}_P$.
4:      Fit a survival forest, $\mathcal{F}$, to the training data using $\mathcal{G}_P$.
5:      Calculate the mean for $D_v^*$ using $\mathcal{F}$. Let $\mathcal{G}$ be the subset
        of genes from $\mathcal{G}_P$ having minimal depth less than this
        threshold.
6:      Let $\mathcal{V}$ be the joint VIMP for $\mathcal{G}$ from $\mathcal{F}$. Set $\Delta = \mathcal{V}$.
7:      while $\Delta > 0$ do
8:          Augment $\mathcal{G}$ to include the next gene in $\mathcal{G}_P$ with smallest
            minimal depth (if there is no such gene, then $\mathcal{G}$ is
            unchanged). Call this new set $\mathcal{G}^+$.
9:          Let $\mathcal{V}^+$ be the joint VIMP for $\mathcal{G}^+$ from $\mathcal{F}$. Set $\Delta =
            \mathcal{V}^+ - \mathcal{V}$.
10:         if $\Delta > 0$ then
11:             Set $\mathcal{V} = \mathcal{V}^+$; $\mathcal{G} = \mathcal{G}^+$.
12:         end if
13:     end while
14:     Fit a survival forest $\mathcal{F}^*$ to the training data using $\mathcal{G}$.
15:     Calculate the prediction error of $\mathcal{F}^*$ over the test data.
16: end for
```

6.1 Some Key Points Concerning the RSF-VH Algorithm

1. The dimension-reduction step (line 3 in the algorithm) is generic. In our examples, we randomly sampled P genes without replacement, but other methods could be used. For example, a preliminary analysis could be used to assign weights to genes indicating their importance, with genes then randomly selected according to these weights (see Sec. 7 for an illustration).
2. When finished, the algorithm returns B independent estimates of prediction error. Averaging these yields an estimate of prediction error for the procedure.
3. Each iteration of the algorithm yields a list of significant genes, $\mathcal{G}$. The mean dimension is the average size of these lists, and the final estimated model is the sorted values of these genes, up to the size of the mean dimension. Genes can be sorted in different ways, including by frequency of occurrence (i.e., number of times a gene is selected over the B iterations) and by mean minimal depth.
4. A key parameter is the number of randomly selected genes, P; however, choosing P is relatively straightforward. A few preliminary trees can be grown, and $\log_2(P)$ can be selected to be roughly the same size as their average tree depth.
5. For P to be as large as possible, it is important that trees be grown to full size. This may reduce the speed of the algorithm; however, only the forest of line 4 (and not that of line 14) needs to be fit in this way.
6. In line 8, the current set of genes, $\mathcal{G}$, can be augmented by including the next $K > 1$ genes with smallest minimal depth. This enables more efficient model searching and also may protect against early termination of the algorithm (line 7).
7. The algorithm is reasonably fast. Joint VIMP is calculated without the need to regrow $\mathcal{F}$. Thus, lines 7–13, although

iterative, take little computational time. The two most expensive computations are the forests grown in lines 4 and 14. Line 14 is usually much faster because the forest is grown using a subset of $\mathscr{G}_P$; however, because n is usually relatively small, even line 4 is fast. This is especially true if a random splitting rule is used.

6.2 Results

We tested RSF-VH on five different benchmark microarray data sets: the diffuse large B-cell lymphoma (DLBCL) data set of Rosenwald et al. (2002), the breast cancer data set of van't Veer et al. (2002), the lung cancer data set of Beer et al. (2002), the acute myeloid leukemia (AML) data set of Bullinger et al. (2004), and the mantle cell lymphoma (MCL) data set of Rosenwald et al. (2003).

Each of these data sets was randomly split into an 80% training set and a 20% test set. Forests were grown over the training data using a randomly selected subset of $P = 500$ genes. When augmenting the gene list in line 8 of the algorithm, we used $K = 5$ (see remark 6 in Sec. 6.1). All forests comprised 1000 survival trees grown under random log-rank splitting with an nsplit value of 10. All trees were grown to full length (i.e., a nodesize of 1). This process was repeated $B = 100$ times. Table 4 gives the average test set prediction error and average model size for each data set.

For comparison, we used the nearest shrunken centroid method of Tibshirani et al. (2002), the supervised principal components method of Bair and Tibshirani (2004), and the L_2-boosting procedure of Hothorn et al., Hothorn and Buhlmann (2006, 2006). We chose these methods because they were specifically designed for the gene selection problem. The procedures were implemented using the R-software packages pamr (Hastie et al. 2002), superpc (Bair and Tibshirani 2004), and mboost (Hothorn et al. 2007), respectively. For pamr and superpc, 10-fold validation was used for tuning; this value was automatically adjusted if the sample size was too small. For mboost, number of boosting iterations was set at 100 (the default) and 1000. We refer to these latter procedures as mboost_{100} and mboost_{1000}, respectively.

Table 4. Test set performance over benchmark microarray data. Values averaged over 100 independent experiments. Procedures are RSF-VH ($P = 500$, $K = 5$), mboost_{100} and mboost_{1000} (L_2-boosting, Hothorn et al. 2006; Hothorn and Buhlmann 2006 with 100 and 1000 boosting iterations, respectively) PAM (nearest shrunken centroids, Tibshirani et al. 2002), and SuperPC (supervised principal components, Bair and Tibshirani 2004)

	AML	DLBCL	Lung cancer	MCL	Breast cancer
			Average prediction error (C)		
RSF-VH	40.1	39.3	32.6	29.8	31.5
mboost_{100}	38.4	37.3	47.4	31.9	35.5
mboost_{1000}	41.2	37.8	43.2	33.3	37.7
PAM	42.5	39.9	31.7	27.6	30.9
SuperPC	39.8	45.2	34.5	29.9	30.1
			Average model size		
RSF-VH	26.4	27.9	37.4	29.6	43.5
mboost_{100}	29.2	31.5	13.8	30.6	22.1
mboost_{1000}	91.4	131.1	41.1	81.9	59.7
PAM	2945.2	2856.4	5382.4	492.5	2484.7
SuperPC	1069.2	2176.3	498.4	2023.9	855.6
			Summary values		
p	6283	7399	7129	8810	4751
n	116	240	86	92	78
No. of deaths	67	138	24	64	34

Each comparison procedure was applied to the same training/test data as RSF-VH (but using all p genes). Note that for prediction error, only the concordance error rate, C, is reported. We do not report CRPS and R^2, because these measures require an estimated survival function. (Of the four methods, only RSF-VH provides such an estimate.) Table 4 presents the average test set prediction error and average model size over the 100 replicates for each procedure.

The results show that RSF-VH performed well, consistently yielding small gene lists and low prediction error. Boosting also performed well, although model sizes were sensitive to the number of boosting iterations; model sizes for mboost_{1000} were sometimes two or more times larger than for mboost_{100}. Also of concern was the finding that prediction error was tied to the censoring rate. Prediction error was relatively poor over the lung and breast cancer data sets, which had the highest censoring rates. Finally, prediction error was generally good under both PAM (nearest shrunken centroids) and SuperPC (supervised principal components), but that number of selected genes was hundreds of times larger than both RSF-VH and mboost.

7. DISCUSSION

Selecting variables in high-dimensional survival settings is challenging. In trying to overcome these challenges, simplifications and strong assumptions are often made. For example, proportional hazards is assumed in many of the approaches advocated for microarray data. Some approaches are univariate, with models fit to one gene at a time (thus potentially missing important multivariable effects). Another tendency is to assume linear relationships for variables. While linear combinations of gene expression values work adequately for some problems, this may not always be the case. For example, microarray studies can involve additional variables, such as clinical data, and including these may require gene interactions or other higher-order modeling. Outside of microarray data, nonlinear effects and interactions are a real concern, and methods that rely on simplistic modeling are at a serious disadvantage.

In contrast, using a nonparametric and data-adaptive method such as RSF automatically addresses these issues. Furthermore, because forests are known to be excellent predictors in high-dimensional settings, they are excellent candidates for use in high-dimensional variable selection.

The challenge in using RSF was the lack of rigorous theory for thresholding variables and for guiding regularization. Current strategies involve using VIMP, but, as we have outlined, this entails difficulties. To circumvent these problems, we introduced a new way to think about variable selection. This led us to maximal subtrees, theory for thresholding noise variables, and an approach to regularizing RSF in big p and small n problems.

Table 5. Test set performance over the ECG data. Values are averaged over 100 independent experiments

Method	Model size	C	CRPS	R^2
RSF-VH$_{50}$	9.0	0.314	0.057	0.034
RSF-VH$^{\star}_{50}$	10.1	0.203	0.049	0.106

Although we have considered only survival analysis in this article, our methods naturally apply to other RF applications as well. Because minimal depth of a variable is independent of outcome and choice of prediction error measure, it applies universally to all RF applications. Computationally, the methodology is easily implemented. In our applications, we simply stored the survival forest in a compressed format and then used recursive algorithms to mine the forest and extract the necessary data. These algorithms are fast. Assuming balanced trees, the number of calculations for each tree is $O(2^{\overline{D}+1})$, where $\overline{D}$ is the average tree depth. Assuming that tree-splitting does not terminate prematurely and that no one is censored, this can be expressed as $O(2n/M)$, where M is the average terminal node size. These expressions are independent of the dimension, p. The dimension p plays a role in the time taken to grow a tree, but not in the parsing of maximal subtrees.

Finally, we remark that while we did not investigate the performance of RSF-VH in Section 6 under more sophisticated dimension-reduction steps (line 3 of the algorithm), we are confident that its performance would have been enhanced by this. As one demonstration of this, consider Table 5, which presents the results from a reanalysis of the ECG data. Two different implementations of RSF-VH were used. In the first implementation, $P = 50$ variables were selected randomly from the $p = 346$ variables. This is similar to how RSF-VH was implemented in Section 6. In the second implementation, $P = 50$ variables were selected, but with variables selected with probability proportional to their VIMP from a preliminary forest fit to training data (see remark 1 of Sec. 6.1). We refer to these two methods as RSF-VH$_{50}$ and RSF-VH$^{\star}_{50}$, respectively. In both cases, 1000 trees were grown with a nodesize value of 2. In both cases, we set $K = 2$ (see remark 6 of Sec. 6.1).

The values reported in Table 5 were averaged over 100 independent replicates using a modified test set validation procedure. Because of the large sample size involved, each replicate used only 5% of the data for training and only 5% for testing. This was done to reduce the computational time.

RSF-VH$^{\star}_{50}$ had substantially better prediction performance than RSF-VH$_{50}$. (The values were not as good as in those reported in Sec. 3, because of the small sample sizes.) In terms of model size, both were roughly the same, and both selected final models with age, low heart rate recovery, and peak metabolic equivalents followed by several of the same ECG variables reported in Section 3; however, the variables found using RSF-VH$^{\star}_{50}$ had more overlap with those reported in Section 3.

APPENDIX: PROOF OF THEOREM 1

By the definition of $\pi_{v,j}$ and $\theta_{v,j}$, if t is a node of depth $j \le d$, then

$$\mathbb{P}\{v \text{ does not split } t | D_v \ge j\} = 1 - \pi_{v,j}\theta_{v,j}.$$

Furthermore, $\mathbb{P}\{D_v = 0\} = \pi_{v,0}\theta_{v,0}$. Therefore, the probability that no maximal v-subtree exists at depth less than $d \ge 1$ is

$$\begin{aligned}\mathbb{P}\{D_v \ge d\} &= \mathbb{P}\{D_v \ge 1\} \prod_{j=1}^{d-1} \mathbb{P}\{v \text{ is not split at depth } j | D_v \ge j\} \\ &= [1 - \mathbb{P}\{D_v = 0\}] \prod_{j=1}^{d-1} \mathbb{P}\{v \text{ is not split at depth } j | D_v \ge j\} \\ &= \prod_{j=0}^{d-1} (1 - \pi_{v,j}\theta_{v,j})^{\ell_j}, \qquad \text{(A.1)}\end{aligned}$$

where $\ell_j = 2^j$ equals the total number of nodes of depth j. Given that $D_v \ge d$, the probability that v splits a node of depth d is 1 minus the probability that each node of depth d is split on some other variable than v. This probability is

$$1 - (1 - \pi_{v,d}\theta_{v,d})^{\ell_d}. \qquad \text{(A.2)}$$

Using

$$\mathbb{P}\{D_v = d\} = \mathbb{P}\{v \text{ is split at depth } d | D_v \ge d\} \times \mathbb{P}\{D_v \ge d\},$$

the result for $d \ge 1$ follows on multiplying (A.1) and (A.2). The case where $d = 0$ holds from $\mathbb{P}\{D_v = 0\} = \pi_{v,0}\theta_{v,0}$.

[Received November 2008. Revised July 2009.]

REFERENCES

Bair, E., and Tibshirani, R. (2004), "Semi-Supervised Methods to Predict Patient Survival From Gene Expression Data," *PLoS Biology*, 2, 0511–0522. [205,215]

——— (2004), "superpc: Supervised Principal Components," R package version 1.05, available at *http://cran.r-project.org*. [215]

Beer, D. G. et al. (2002), "Gene Expression Profiles Predict Survival of Patients With Lung Adenocarcinoma," *Nature Medicine*, 8, 816–824. [215]

Breiman, L. (2001a), "Random Forests," *Machine Learning*, 45, 5–32. [205, 206]

——— (2001b), "Statistical Modeling: The Two Cultures," *Statistical Science*, 16, 199–231. [206]

Bullinger, L. et al. (2004), "Use of Gene-Expression Profiling to Identify Prognostic Subclasses in Adult Acute Myeloid Leukemia," *The New England Journal of Medicine*, 350, 1605–1616. [215]

Bureau, A., Dupuis, J., Falls, K., Lunetta, K. L., Hayward, B., Keith, T. P., and Eerdewegh, P. V. (2005), "Identifying SNPs Predictive of Phenotype Using Random Forests," *Genetic Epidemiology*, 28, 171–182. [206]

Clarke, J., and West, M. (2008), "Bayesian Weibull Tree Models for Survival Analysis of Clinico-Genomic Data," *Statistical Methodology*, 5, 238–262. [205]

Diaz-Uriarte, R., and Alvarez de Andres, S. (2006), "Gene Selection and Classification of Microarray Data Using Random Forest," *BMC Bioinformatics*, 7, 3. [205,206]

Docksum, K., Tang, S., and Tsui, K.-W. (2008), "Nonparametric Variable Selection: The EARTH Algorithm," *Journal of the American Statistical Association*, 103, 1609–1620. [213]

Fleming, T., and Harrington, D. (1991), *Counting Processes and Survival Analysis*, New York: Wiley. [213]

Gerds, T. A. (2008), "pec: Prediction Error Curves for Survival Models," R package version 1.0.7, available at *http://cran.r-project.org*. [211]

Gerds, T. A., and Schumacher, M. (2006), "Consistent Estimation of the Expected Brier Score in General Survival Models With Right-Censored Event Times," *Biometrical Journal*, 6, 1029–1040. [211]

Graf, E., Schmoor, C., Sauerbrei, W., and Schumacher, M. (1999), "Assessment and Comparison of Prognostic Classification Schemes for Survival Data," *Statistics in Medicine*, 18, 2529–2545. [211]

Harrell, F., Califf, R., Pryor, D., Lee, K., and Rosati, R. (1982), "Evaluating the Yield of Medical Tests," *Journal of the American Medical Association*, 247, 2543–2546. [211]

Hastie, T., Tibshirani, R., Narasimhan, B., and Chu, G. (2002), "pamr: Prediction Analysis for Microarrays," R package version 1.31, available at *http://cran.r-project.org*. [215]

Hothorn, T., and Buhlmann, P. (2006), "Model-Based Boosting in High-Dimensions," *Bioinformatics*, 22, 2828–2829. [205,215]

Hothorn, T., Buhlmann, P., Dudoit, S., Molinaro, A., and van der Laan, M. J. (2006), "Survival Ensembles," *Biostatistics*, 7, 355–373. [205,215]

Hothorn, T., Buhlmann, P., Kneib, T., and Schmid, M. (2007), "mboost: Model-Based Boosting," R package version 1.0-1, available at *http://cran.r-project.org*. [215]

Hothorn, T., Hornik, K., and Zeileis, A. (2006), "Unbiased Recursive Partitioning," *Journal of Computational and Graphical Statistics*, 15, 651–674. [205,213,214]

Huang, J., Ma, S., and Xie, H. (2006), "Regularized Estimation in the Accelerated Failure Time Model With High-Dimensional Covariates," *Biometrics*, 62, 813–820. [205]

Ishwaran, H. (2007), "Variable Importance in Binary Regression Trees and Forests," *Electronic Journal of Statistics*, 1, 519–537. [206]

Ishwaran, H., and Kogalur, U. B. (2007), "Random Survival Forests for R," *Rnews*, 7/2, 25–31. [210]

——— (2008a), "Consistency of Random Survival Forests." [205]

——— (2008b), "RandomSurvivalForest: Random Survival Forests," R package version 3.5.1, available at *http://cran.r-project.org*. [210,213]

Ishwaran, H., Blackstone, E. H., Hansen, C. A., and Rice, T. W. (2009), "A Novel Approach to Cancer Staging: Application to Esophageal Cancer," *Biostatistics*, 10, 603–620. [206]

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., and Lauer, M. S. (2008), "Random Survival Forests," *The Annals of Applied Statistics*, 2, 841–860. [205,206,210,211]

Li, H., and Luan, Y. (2006), "Boosting Proportional Hazards Models Using Smoothing Splines, With Applications to High-Dimensional Microarray Data," *Bioinformatics*, 21, 2403–2409. [205]

Li, H. Z., and Gui, J. (2004), "Partial Cox Regression Analysis for High-Dimensional Microarray Gene Expression Data," *Bioinformatics*, 20, 208–215. [205]

Loh, W.-Y., and Shih, Y.-S. (1997), "Split Selection Methods for Classification Trees," *Statistica Sinica*, 7, 815–840. [210]

Loh, W.-Y., and Vanichsetakul, N. (1988), "Tree-Structured Classification via Generalized Discriminant Analysis," *Journal of the American Statistical Association*, 83, 715–725. [210]

Lunetta, K. L., Hayward, L. B., Segal, J., and Eerdewegh, P. V. (2004), "Screening Large-Scale Association Study Data: Exploiting Interactions Using Random Forests," *BMC Genetics*, 5, 32. [206]

Ma, S., and Huang, J. (2006), "Clustering Threshold Gradient Descent Regularization: With Applications to Microarray Studies," *Bioinformatics*, 23, 466–472. [205]

Ma, S., Kosorok, M. R., and Fine, J. P. (2006), "Additive Risk Models for Survival Data With High-Dimensional Covariates," *Biometrics*, 62, 202–210. [205]

Nguyen, D., and Rocke, D. M. (2002), "Partial Least Squares Proportional Hazard Regression for Application to DNA Microarray Data," *Bioinformatics*, 18, 1625–1632. [205]

Park, M.-Y., and Hastie, T. (2007a), "L_1-Regularization Path Algorithm for Generalized Linear Models," *Journal of the Royal Statistical Society, Ser. B*, 69, 659–677. [205,212]

——— (2007b), "glmpath: L_1 Regularization Path for Generalized Linear Models and Cox Proportional Hazards Model," R package version 0.94, available at *http://cran.r-project.org*. [212]

Ridgeway, G. (1999), "The State of Boosting," *Computing Science and Statistics*, 31, 172–181. [205,213,214]

——— (2007), "Generalized Boosted Regression Models (gbm)," R package version 1.6-3, available at *http://cran.r-project.org*. [214]

Rosenwald, A. et al. (2002), "The Use of Molecular Profiling to Predict Survival After Chemotherapy for Diffuse Large B-Cell Lymphoma," *The New England Journal of Medicine*, 346, 1937–1947. [215]

——— (2003), "The Proliferation Gene Expression Signature Is a Quantitative Integrator of Oncogenic Events That Predicts Survival in Mantle Cell Lymphoma," *Cancer Cell*, 3, 185–197. [215]

Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (2002), "Diagnosis of Multiple Cancer Types by Shrunken Centroids of Gene Expression," *Proceedings of the National Academy of Sciences USA*, 99, 6567–6572. [205,215]

vant Veer, L. J. et al. (2002), "Gene Expression Profiling Predicts Clinical Outcome of Breast Cancer," *Nature*, 415, 530–536. [215]

Weichselbaum, R. R., Ishwaran, H., Yoon, T., Nuyten, D. S., Baker, S. W., Khodarev, N., Su, A. W., Shaikh, A. Y., Roach, P., Kreike, B., Roizman, B., Bergh, J., Pawitan, Y., van de Vijver, M. J., and Minn, A. J. (2008), "An Interferon-Related Gene Signature for DNA Damage Resistance Is a Predictive Marker for Chemotherapy and Radiation for Breast Cancer," *Proceedings of the National Academy of Sciences USA*, 105 (47), 18490–18495. [206]

Wirapati, P. et al. (2008), "Meta-Analysis of Gene Expression Profiles in Breast Cancer: Toward a Unified Understanding of Breast Cancer Subtyping and Prognosis Signatures," *Breast Cancer Research*, 10 (4), R65. [213]

Zhang, H. H., and Lu, W. (2007), "Adaptive Lasso for Cox's Proportional Hazards Model," *Biometrika*, 94, 691–703. [205,212]

Zou, H., and Hastie, T. (2005), "Regularization and Variable Selection via the Elastic Net," *Journal of the Royal Statistical Society, Ser. B*, 67 (2), 301–320. [213]

View publication stats