Year 2005 *Paper* 189

Cross-validated Bagged Prediction of Survival

Sandra E. Sinisi[*] Romain Neugebauer[†]
Mark J. van der Laan[‡]

[*] Division of Biostatistics, School of Public Health, University of California, Berkeley, sinisi54@alum.berkeley.edu

[†] Division of Biostatistics, School of Public Health, University of California, Berkeley

[‡] Division of Biostatistics, School of Public Health, University of California, Berkeley, laan@berkeley.edu

Cross-validated Bagged Prediction of Survival

Sandra E. Sinisi, Romain Neugebauer, and Mark J. van der Laan

Abstract

In this article, we show how to apply our previously proposed Deletion/Substitution/Addition algorithm in the context of right-censoring for the prediction of survival. Furthermore, we introduce how to incorporate bagging into the algorithm to obtain a cross-validated bagged estimator. The method is used for predicting the survival time of patients with diffuse large B-cell lymphoma based on gene expression variables.

Contents

1 Introduction

In some medical studies, data is collected on newly diagnosed cancer patients in hopes of finding significant prognostic factors. Treating cancer may negatively lead to disease recurrence or death from disease in which case the time to event can be measured along with many covariates. Covariates usually include epidemiological and histological variables, but now it is common to include measurements on expression levels for thousands of genes as additional covariates. Gene expression profiling is being used in the prognosis of breast cancer, colon cancer, ovarian cancer, and lymphoma to name a few. For instance, there have been a number of studies using gene expression to predict cancer survival in patients with non-Hodgkin's lymphoma.

Non-Hodgkin's lymphoma is a cancer of the lymphatic system. The lymphatic system, which is part of the body's immune system, is a complex system made up of lymph organs, such as the bone marrow, the thymus, the spleen, and the lymph nodes (or lymph glands). These are connected by a network of tiny lymphatic vessels. Lymph nodes are found all over the body. Lymph is a colourless fluid that circulates through the lymphatic system. It contains cells known as lymphocytes which are a type of white blood cell and an essential part of the body's defense against infection and disease. There are two main types of lymphocyte: B-cells and T-cells. Most lymphocytes start growing in the bone marrow. The B-cells continue to develop in the bone marrow, while the T-cells go from the bone marrow to the thymus gland and mature there. When they are mature, both B-cells and T-cells help to fight infections.

There are more than 20 different types of non-Hodgkin's lymphoma. Diffuse large B-cell lymphoma (DLBCL) is a common type, making up about 40 percent of all cases. It is a cancer of the B-lymphocytes. With chemotherapy, about 40 percent of patients with DLBCL can have long-term, disease-free survival; some may even be cured. Untreated, however, it may lead to death in one to two years (Rosenwald et al., 2002). Risk of disease recurrence and an idea of overall survival usually is determined by an international prognostic index (IPI) which takes into account age, stage of disease, general health (also known as performance status), number of extra nodal sites, and presence or absence of an elevated serum enzyme called lactate dehydrogenase (LDH). As an alternative to the IPI, gene expression in lymphocytes can be used to get a sense of overall survival. In this article, we are interested in modeling survival time of patients with diffuse large B-cell lymphoma (DLBCL) by their gene expression levels. We will propose a method to predict time to event by the measured covariates and apply it to a DLBCL data set.

1.1 Methods for Prediction of Survival

Survival analysis is concerned with the distribution of lifetimes, and the major distinguishing feature of survival analysis is *censoring*. A subject may not be observed for its entire *lifetime*, so that we may only know, for example, that the subject survived to the end of the trial. At the time of a study, a patient may have dropped out of the study, been lost to follow-up, or not had the particular event, in which case the last date of follow-up is recorded and referred to as the censored time to event. Let T denote a lifetime random variable. Right-censoring occurs when the subject leaves the trial at time C_i where we know either T_i if $T_i \leq C_i$ or that $T_i > C_i$.

Proportional hazards models (*Cox regression*) are commonly used to estimate covariate effects in survival analysis. A number of other methods have been proposed for nonparametric regression of survival outcomes. Kooperberg et al. (1995) developed an adaptive hazard regression (HARE) methodology for estimating the conditional log-hazard function based on (censored) survival data with one or more covariates. Hastie and Tibshirani (1990) fit additive proportional hazards models where covariate effects are modeled through sums of univariate smooth functions. There are also many modifications to classification and regression trees (CART) (Breiman et al., 1984) that are specific to censored survival data (Gordon and Olshen, 1985; Davis and Anderson, 1989; LeBlanc and Crowley, 1992; Segal, 1988). In particular, Molinaro et al. (2004) introduced a procedure for tree-structured estimation of censored data based on the unified loss-based estimation methodology of van der Laan and Dudoit (2003).

An important result of the loss-based estimation approach for right-censored data (van der Laan and Dudoit, 2003) applied by Molinaro et al. (2004) is the following. Suppose that we have complete (i.e., uncensored) data. In that case, we would calculate a *complete data survival predictor* by applying a particular data-driven model selection criterion to select a single model out of a set of candidate models. We would then report the efficient survival predictor computed under this selected model. A methodology for building a predictor and assessing its performance based on censored data should satisfy the following two properties. First, the censored data methodology when applied to uncensored data should predict the same survival as the *complete data survival predictor*. The survival tree methods (Segal, 1988; Davis and Anderson, 1989) appear to lack this property because when applied to the complete (i.e., uncensored) data, they do not reduce to a complete data methodology. In other words, the split functions used in survival trees are choices that are convenient for handling censored data but do not reduce to the choices suggested for uncensored data. Second, none of the methods mentioned incorporate external covariate processes to allow for informative censoring and gain in efficiency. We like to retain these two properties in our application.

1.2 Model

In this article, we develop a new prediction algorithm by building upon our previously proposed D/S/A algorithm (Sinisi and van der Laan, 2004) to allow right-censoring, and we propose an aggregation scheme for bagging the D/S/A algorithm.

Ideally, we would like to observe the true survival time for each patient. Let the full-data structure (of interest) be $X = \bar{X}(T) \equiv (X(t) : t \leq T)$ indexed by time t where T can denote a random survival time, and let $Z = \log T$. Denote the distribution of the full data structure X by $F_{X,0}$. The full-data structure incorporates covariates which may contain both time-dependent and time-independent covariates.

However, in realistic settings the observed data structure is given by

$$O \equiv \left(\tilde{T} = \min(T, C),\ \Delta = I(T \leq C),\ \bar{X}(\tilde{T})\right).$$

We observe the full data process $X(t)$ up to the minimum $\tilde{T}$ of the survival time T and a right-censoring variable C, with conditional distribution $G_0(\cdot|X)$ given the full data structure X. The missing, or *censored*, survival data can be due to drop out or the end of follow-up, for example. By convention, if T occurs prior to C $(T < C)$, then we set $C = \infty$. Thus, C is always observed and one can rewrite the observed data structure as $O = (C, \bar{X}(C))$. The distribution, $P_0 = P_{F_{X,0},G_0}$, of the observed data structure O is indexed by the full data distribution $F_{X,0}$ and the conditional distribution $G_0(\cdot|X)$ of the censoring variable C. $G_0(\cdot|X)$ is referred to as the *censoring* or *coarsening mechanism*. The survival function for the censoring mechanism is denoted by $\bar{G}_0(c|X) = Pr_0(C \geq c|X)$. We assume that G_0 satisfies the *coarsening at random* (CAR) assumption:

$$Pr_0(C = t \mid C \geq t, \bar{X}(T)) = Pr_0(C = t \mid C \geq t, \bar{X}(t)), \qquad \text{for } t < T.$$

If X does not include time-dependent covariates (e.g., $X = (W, Z)$), then CAR is equivalent to assuming that C is conditionally independent of the survival time T, given baseline covariates W. For details, we refer to van der Laan and Robins (2003).

1.3 Loss Function

When observing right-censored data, we have a learning set of n independent and identically distributed (i.i.d.) observations, $O_1, \ldots, O_n$, from the right-censored data structure, $O_i \sim P_0 = P_{F_{X,0},G_0}$. Let the empirical distribution of $O_1, \ldots, O_n$ be denoted by P_n. Our goal is to find a predictor of log survival time Z based on covariates W, i.e., an estimator of the parameter ψ_0 defined in terms of the risk for a full data loss function $L(X, \psi)$. The methodology presented in van der Laan and Robins (2003) suggests replacing the full (uncensored) data loss function with an observed (censored) data loss function.

Inverse Probability of Censoring Weighted Loss Function

A way to define the observed data loss function is the application of *inverse probability of censoring weights* (IPCW) (van der Laan and Robins, 2003). IPCW estimation derives its name from the fact that the full data function is weighted by the inverse of a censoring probability. For univariate prediction, our parameter of interest is the conditional expectation, $\psi(W) = E(Z|W)$, corresponding to the squared error loss function $L(X, \psi) = (Z - \psi(W))^2$. The IPCW observed data loss function for the squared error loss is (van der Laan and Robins, 2003):

$$\begin{aligned} L(O, \psi \mid G_0) &= L(X, \psi)\frac{\Delta}{\bar{G}_0(T|X)} \\ &= (Z - \psi(W))^2 \frac{\Delta}{\bar{G}_0(T|X)} \qquad (1) \end{aligned}$$

where $\Delta = I(T \leq C)$ and $\bar{G}_0$ is the conditional survival function for the censoring time C given full data X. The full data loss function is weighted by the inverse probability of being censored after time $\tilde{T}$ given the covariates. Under the CAR assumption, $\bar{G}_0(\cdot|X)$ is a function of only the observed data structure O.

Note that the conditional censoring survivor function $\bar{G}_0$ is typically unknown and needs to be replaced by an estimate $\bar{G}_n$. The simplest choice is the Kaplan-Meier estimate, but other procedures (e.g., Cox model) are available. Other choices of the observed data loss function are possible as well, such as the optimal doubly robust inverse probability of censoring weighted (DR-IPCW) loss function (van der Laan and Robins, 2003). For finite sample and asymptotic results regarding the cross-validation selector based on these loss functions, we refer to van der Laan and Dudoit (2003) and Keleş et al. (2003). For now, we assume that $G_n = \hat{G}(P_n)$ is given, and we discuss this in Section 2.2.

This article is organized such that it presents the D/S/A algorithm for prediction of survival in Section 2. After having described the algorithm, we show how to obtain a cross-validated bagged estimator in Section 3. Section 4 provides some simulated results, and we illustrate its application to the prediction of survival in patients with diffuse large B-cell lymphoma in Section 5.

2 Cross-validated D/S/A Algorithm

The *Deletion/Substitution/Addition algorithm*, or *D/S/A algorithm* (Sinisi and van der Laan, 2004), is a data-adaptive learning methodology which can be used to predict the conditional expectation of an outcome or response Z given a set of inputs or explanatory variables W.

It is helpful to review our estimation road map in the context of censoring which can be summarized in three steps. 1) Our parameter of interest is $\psi_0(W) = E_{P_0}(Z|W)$ and can be defined as the risk minimizer of an observed data loss function:

$$\psi_0 = \arg\min_{\psi \in \Psi} E_0 L(O, \psi | G_0).$$

2) We will generate a sequence of candidate estimators by minimizing the empirical risk over subspaces of increasing dimension approximating the com-

plete parameter space Ψ. We define a collection of subspaces $\Psi_s \subset \Psi$, indexed by s. For each choice of subspace s, we denote our candidate estimators with $\hat{\Psi}_s(P_n)$. 3) Select s with cross-validation.

Parameterization

Define a set of basis functions, $\{\phi_{\vec{p}} : \vec{p} \in \mathbb{N}^d\}$. These basis functions are polynomials and denoted by $\phi_{\vec{p}}(W) = W_1^{p_1} \ldots W_d^{p_d}$ given a d-vector $\vec{p} = (p_1, \ldots, p_d) \in I$, where I is an index set, $I \in \mathcal{I}$.

Every parameter $\psi \in \Psi$ can be approximated as a linear combination of tensor products of polynomial basis functions:

$$\psi_{I,\beta}(W) \equiv \sum_{\vec{p} \in I} \beta_{\vec{p}} \phi_{\vec{p}}(W).$$

The complete parameter space $\boldsymbol{\Psi}$ can be written as the collection of basis functions and represented by

$$\boldsymbol{\Psi} \equiv \{\psi_{I,\beta}(W) = \sum_{\vec{p} \in I} \beta_{\vec{p}} \phi_{\vec{p}}(W) : \beta,\, I \in \mathcal{I}\}.$$

Selection of Sieve

Define a collection of subspaces $\Psi_s \subset \boldsymbol{\Psi}$ of increasing dimension approximating the complete parameter space $\boldsymbol{\Psi}$, such as,

$$\Psi_s \equiv \left\{ \psi_{I,\beta}(W) = \sum_{\vec{p} \in I} \beta_{\vec{p}} \phi_{\vec{p}}(W) : \beta,\, I,\, m(I) \leq s \right\}.$$

The fine-tuning parameters, $s = (s_0, s_1, s_2, s_3)$, define the allowed index sets I. s_1 represents the number of tensor products; s_2 represents the maximal order of interaction of tensor products:

$$\max_{\vec{p} \in I} \sum_{j=1}^{d} I(p_j \neq 0)$$

(i.e., the number of non-zero components in $\vec{p}$); s_3 represents the maximal sum of powers of tensor products:

$$\max_{\vec{p} \in I} \sum_{j=1}^{d} p_j.$$

The dimension of W can be reduced to s_0 and is described below. Now for every s we want to find the estimator which minimizes the empirical risk over the subspace Ψ_s. Define $\hat{\Psi}_s(P_n)$ as the minimizer of the empirical mean of the IPCW loss function in $\psi_{I,\beta}$. This minimization can be done by first minimizing over β for a fixed I. Then it is left to minimize a function of I, $f_E(I)$, for which we propose the D/S/A algorithm.

To summarize, for each choice of s, the algorithm computes an estimator which is a regression with s_1 terms of maximal order s_2. These s-specific estimators are referred to as candidate estimators, and cross-validation will be used to select the fine-tuning parameters.

Estimator Construction

A set of constraints over which to search is specified: $s_0 = \{1, 2, \ldots\}$, $s_1 = \{1, 2, \ldots\}$, and $s_2 = \{1, 2, \ldots\}$. Assume a fixed ordering of $W = W_1, \ldots, W_d$. We proposed to obtain this ordering data-adaptively using ordered T-statistics based on marginal regressions, but other orderings (and transformations) are possible such as dimension reduction using principal components.

To reduce the dimension:

1. compute each T-statistic corresponding to the main effects of W_j, $j = 1, \ldots, d$ by fitting d univariate regressions

2. rank these statistics (in absolute value) in decreasing order $\hat{R}(1), \ldots, \hat{R}(d) \subset \{1, \ldots, d\}$ yielding the ordered covariates $W_{\hat{R}(1)}, W_{\hat{R}(2)}, \ldots, W_{\hat{R}(d)}$ which we will refer to as $W_1, \ldots, W_d$

3. input the set $(W_1, \ldots, W_{s_0}\colon s_0 \leq d)$ of length s_0 as the vector of searchable covariates

The algorithm starts by fitting a model with the main term, W_1 or W_2 or $\ldots W_{s_0}$, which minimizes our empirical observed loss function $f_E(I)$. Next, the algorithm cycles through a set of *deletion*, *substitution*, and *addition* moves. The process can be summarized as follows:

Algorithm: *dsa*

```
1. Perform dimension reduction
2. For $s_0 = 1, 2, \ldots$
    (a) Input ordered covariates of length $s_0$
    (b) For $s_2 = 1, 2, \ldots$
        i. Initiate algorithm
        ii. (*) Denote current working model by an index set $I_0$ of size $s_1 = |I_0|$.
        iii. Try to improve upon our current fit with an index set $I^-$ of size $s_1 - 1$ among all allowed deletion moves. If this provides an improvement, then set $I_0 = I^-$ and go back to (*).
        iv. Otherwise, find an optimal updated index set $I^=$ of the same size $s_1$ as $I_0$, among all allowed substitution moves. If this update improves on $I_0$, then set $I_0 = I^=$ and go back to (*).
        v. Otherwise, find an optimal updated index set $I^+$ of size $s_1 + 1$ among all allowed addition moves. Set $I_0 = I^+$ and go back to (*).
        vi. Stopping rule. Run the algorithm until the current index set size $s_1 = |I_0|$ is larger than a user-supplied maximum size.
```

Throughout this process, the algorithm is keeping track of the *best* estimators for all choices of s, $\hat{\Psi}_s(P_n)$.

Deletion/Substitution/Addition moves

The deletion, substitution, and addition moves can be described with the following notation.

Deletion moves. Simply try to remove one of the terms in the current fit

and fit a regression model of size $s_1 - 1$.
Substitution moves. Given an index set I of size $s_1 = |I|$, define a set $SUB(I)$ of index sets of same size s_1 by replacing individual elements $\vec{p} \in I$ by one of the $2d$ vectors created by adding or subtracting 1 to any of the d components of $\vec{p}$:

$$SUB(I) \rightarrow \begin{cases} (p_1+1, p_2, p_3, \ldots, p_d) \\ (p_1, p_2+1, p_3, \ldots, p_d) \\ \vdots \\ (p_1, p_2, p_3, \ldots, p_d+1) \\ (p_1-1, p_2, p_3, \ldots, p_d) \\ (p_1, p_2-1, p_3, \ldots, p_d) \\ \vdots \\ (p_1, p_2, p_3, \ldots, p_d-1) \end{cases}$$

for each $\vec{p} \in I$.
Addition moves. Given an index set I of size $s_1 = |I|$, define a set $ADD(I)$ of index sets of size $s_1 + 1$, by adding to I an element of $SUB(I)$ or one of the d unit vectors $\vec{u}_j$, $j = 1, \ldots, d$:

$$ADD(I) \rightarrow \begin{cases} (1, 0, \ldots, 0) \\ \vdots \\ (0, \ldots, 0, 1) \\ (p_1+1, p_2, p_3, \ldots, p_d) \\ \vdots \\ (p_1, p_2, p_3, \ldots, p_d+1) \\ (p_1-1, p_2, p_3, \ldots, p_d) \\ \vdots \\ (p_1, p_2, p_3, \ldots, p_d-1) \end{cases}$$

In addition, alternate-substitution moves are used when trying a term with more than s_2 non-zero components. In that case, $\vec{p}$ is replaced with the s_2 vectors obtained from setting one of the original non-zero components to zero (Sinisi and van der Laan, 2004).

Cross-validation Selector

Cross-validation divides the available *learning* set into a *training* set and a *validation* set. Observations in the training set are used to construct (or *train*) the estimators, and observations in the validation set are used to assess the performance of (or *validate*) these estimators. The cross-validation selector is chosen to have the best performance on the validation sets.

The algorithm uses v-fold cross-validation to select the fine-tuning parameters. To derive a general representation for cross-validation, let $B_n \in \{0,1\}^n$ be a random vector whose observed value defines a split of the observed data $O_1, \ldots, O_n$, the learning sample, into a validation sample and a training sample. If $B_n(i) = 0$ then observation i is placed in the training sample and if $B_n(i) = 1$, it is placed in the validation sample. With v-fold cross-validation, we have v different B_n split vectors. The empirical distribution of the data in the training sample and validation sample are denoted by P^0_{n,B_n} and P^1_{n,B_n}, respectively. The proportion of observations in the validation sample is denoted by $p = \sum_i B_n(i)/n$.

The cross-validation selector of s is now defined as

$$\begin{aligned}\hat{s}(P_n) &\equiv \text{argmin}_s E_{B_n} \int L(O, \hat{\Psi}_s(P^0_{n,B_n}) | \hat{G}(P^0_{n,B_n})) dP^1_{n,B_n}(O) \\ &= \text{argmin}_s E_{B_n} \frac{1}{np} \sum_{i=1}^{n} I(B_n(i) = 1) L(O_i, \hat{\Psi}_s(P^0_{n,B_n}) | \hat{G}(P^0_{n,B_n})).\end{aligned}$$

Final Estimator

The algorithm builds estimators $\hat{\Psi}_s$ for all choices of s_0, s_1, and s_2 on each of the v training sets. It evaluates the cross-validated risk of these estimators on the corresponding validation set. This results in a three-dimensional matrix of cross-validated risks. The values of (s_0, s_1, s_2) that correspond to the minimal cross-validated risk are selected: $\hat{s}(P_n)$. The algorithm is now run on the learning set for $\hat{s}_0$ and $\hat{s}_2$ and the *best* estimator of size $\hat{s}_1$ is reported. The final estimator is denoted by $\hat{\Psi}_{\hat{s}(P_n)}(P_n)$. A summary of this process is:

Algorithm: $cv - survdsa$

1. Estimate $\hat{G}(P^0_{n,B_n})$ and form weights for respective training sets; insert into the `weight` argument

2. Run *dsa* (for all $s = (s_0, s_1, s_2)$) on training sample to obtain $\hat{\Psi}_s(P^0_{n,B_n})$

3. Compute empirical risk over the validation sample (repeat for all v training/validation sets)

4. Choose the cross-validation selector, $\hat{s}(P_n)$

5. Estimate $\hat{G}(P_n)$ and form weights for learning set; insert into the `weight` argument

6. Run dsa on learning sample

7. Final estimator is given by $\hat{\Psi}_{\hat{s}(P_n)}(P_n)$

2.1 Variable Importance Measures

In addition to reporting an optimal predictor, the algorithm produces an importance measure for each variable. As before, let the data be n observations of (Y, W), where Y is the outcome of interest and W is a d-dimensional vector of covariates for which we would like a measure of importance.

We want to compute an importance measure for each variable W_j, $j = 1, \ldots, d$. Let W_{-j} represent all variables other than W_j. We can write $m_j(w) = E_{W_{-j}} E(Y|W_j = w, W_{-j})$. The variable importance measure is given by $|m_j(w) - m_j(0)|$ and can be plotted as a function of w. In the case of binary variables, it simply is $|m_j(1) - m_j(0)|$.

2.2 Estimating the Survival Function for the Censoring Mechanism

A new component of the D/S/A algorithm for predicting survival is the need to estimate $\bar{G}_n$ and thus estimate the inverse probability of censoring *weights*. As it is written, the user can supply any set of desired weights to be read in by the algorithm. In the case of non-informative censoring, one can simply use Kaplan-Meier to estimate $\bar{G}_n$. Otherwise, a possible approach is to estimate the weights with a Cox proportional hazard model.

A question that arises when estimating $\bar{G}_n$ is to decide which covariates to include in the estimate. A model selection technique for hazard regression is available from the `R` function `hare` in the `polspline` library (Kooperberg et al., 1995). `Hare` fits a hazard regression model by using linear splines to model the baseline hazard, covariates, and interactions. The function `phare` estimates the conditional probabilities from the fitted hazard regression model and yields an estimate of $\bar{G}_n$. Recall that the observed data structure is given by

$$O \equiv \Big(\tilde{T} = \min(T, C),\ \Delta = I(T \leq C),\ \bar{X}(\tilde{T})\Big).$$

Let $\Delta_c = 1 - \Delta$. The weights are given by:

$$\frac{\Delta_i}{\bar{G}_n(T_i|\bar{X}(C))}.$$

The following `R` code will yield the weights for the learning set using `hare`:

```
hareFit <- hare(ttilde,deltac,w)
gBar <- phare(ttilde,w,hareFit)
 wtsLearningSet <- delta/gBar
```

The estimate above is of $\hat{G}(P_n)$. Similarly, one can form weights for the training set by first estimating $\hat{G}(P^0_{n,B_n})$. A final note is that the weights are truncated at a user-defined truncation level. For example, in the simulations, we used a 5% truncation level ($0.05 \times n$) so that a single observation will never represent more than 5% of the learning sample, but for the data analysis, we used an absolute truncation level of five.

3 Cross-validated Bagged D/S/A Algorithm

3.1 Brief Review of Bagging

Bagging, or "**b**ootstrap **agg**regat**ing**", was introduced by Breiman (1996a) as a tool for reducing the variance of a predictor. The general idea is to generate multiple versions of a predictor and then using these to get an aggregated predictor. The multiple predictors are obtained by using bootstrap replicates of the data, and bagging is meant to yield gains in accuracy. The stability of the procedure that constructs each predictor is related to whether bagging will improve accuracy (Breiman, 1996a). Breiman (1996b) studied instability and found that bagging works well for unstable methods such as subset selection in linear regression.

Several approaches have been offered to combine different classifiers (LeBlanc and Tibshirani, 1996; Breiman, 1996c; Hothorn and Lausen, 2003). In addition, modifications of bagging have been proposed: "nice" bagging (Skurichina and Duin, 1998), sub-bagging or sub-sample aggregating (Buhlmann and Yu, 2002), iterated bagging or de-biasing (Breiman, 2001). Friedman and Hall (2000) show that bagging reduces variability when applied to highly nonlinear estimators such as decision trees and neural networks and can also reduce bias for certain types of estimators. Breiman (2001) show that iterated bagging is effective in reducing both bias and variance. Bagging has been viewed from its ability to reduce instability (Buhlmann and Yu, 2002) and its success with nonlinear features of statistical method (Friedman and Hall, 2000; Buja and Stuetzle, 2002). Hall and Samworth (2005) address the way its performance depends on re-sample size. Finally, ensemble methods have been used in the presence of censoring: bagging survival trees (Hothorn et al., 2003) and random forests for censored data (Hothorn et al., 2005).

3.2 CV-Bagged D/S/A Algorithm

van der Laan et al. (2005) proposed a general method for cross-validated bagging such that the cross-validation is performed external to the aggregation. This is suggested in order to achieve the correct trade-off between bias and variance for the aggregated estimators.

Our motivation for developing a cross-validated bagged (D/S/A) estimator arised from data applications. For example, we were interested in predicting viral load in a population of patients with HIV based on genotype and treatment history. Applying the D/S/A algorithm to this data rich in covariates resulted in a low dimensional fit. Although such an estimator is based on a sensible trade off between bias and variance, the resulting fit is disappointing in two respects. First, in many applications the true regression is believed to be a function of nearly all variables where many variables make a small contribution. Second, a clinician would like to obtain a measure of importance for each variable considered (van der Laan et al., 2005). But such a low dimensional fit reflects zero importance for all the variables that do not appear in the final fit. Based on these concerns, we propose to construct (D/S/A) estimators that 1) are high dimensional, so that the majority of variables contribute to the obtained regression, and 2) maintain a sensible trade-off between bias and variance. In this section, we will show how we employ “bagging” into our D/S/A algorithm and detail how cross-validation enters our approach.

Let *dsa* refer to the process outlined in Section 2 used to generate s-specific candidate estimators. We will view each bagged *dsa* (indexed by s) as candidates, and use cross-validation to select amongst these candiate bagged estimators.

Algorithm: $bagged - dsa$

1. For $b = 1$ to B
    (a) Draw bootstrap sample $P_{nb}^{\#}$ from the empirical probability distribution P_n
    (b) Run *dsa* (for all s) on $P_{nb}^{\#}$ to obtain $\hat{\Psi}_s(P_{nb}^{\#})$
2. Average these estimators:

$$\tilde{\Psi}_s(P_n) = \frac{1}{B}\sum_{b=1}^{B}\hat{\Psi}_s(P_{nb}^{\#})$$

For each choice of s, we now have an aggregated predictor. In other words, this results in a set of candidate bagged estimators $\tilde{\Psi}_s(P_n)$ indexed by s. Our goal is to data-adaptively select the s which minimizes the risk of $\tilde{\Psi}_s(P_n)$, and we need to use cross-validation appropriately to do so.

Recall that we are using B_n to define the v-fold cross-validation scheme. P_{n,B_n}^0 denotes the empirical distribution of the observations in the training set, and P_{n,B_n}^1 denotes the empirical distribution of the observations in the validation set.

Algorithm: $cv - bagged - dsa$

1. For $b = 1$ to B
    (a) Draw bootstrap training sample $P_{n,B_n,b}^{0\#}$ from the training sample P_{n,B_n}^0
    (b) Run *dsa* (for all s) on $P_{n,B_n,b}^{0\#}$ to obtain $\hat{\Psi}_s(P_{n,B_n,b}^{0\#})$

2. Average these to obtain:

$$\tilde{\Psi}_s(P^0_{n,B_n}) = \frac{1}{B}\sum_{b=1}^{B}\hat{\Psi}_s(P^{0\#}_{n,B_n,b})$$

3. Compute empirical risk over the validation sample (repeat for all v training/validation sets)

4. Choose the cross-validation selector:

$$\hat{S}(P_n) = \arg\min_s E_{B_n} \sum_{i,B_n(i)=1} L(O_i, \tilde{\Psi}_s(P^0_{n,B_n})) \qquad (2)$$

5. For $b = 1$ to B

   (a) Draw bootstrap learning sample $P^{\#}_{n,b}$ from the learning sample P_n

   (b) Run *dsa* on $P^{\#}_{n,b}$ to obtain $\hat{\Psi}_s(P^{\#}_{n,b})$

6. Average these to obtain:

$$\tilde{\Psi}_s(P_n) = \frac{1}{B}\sum_{b=1}^{B}\hat{\Psi}_s(P^{\#}_{n,b})$$

The final fit is the bagged estimator corresponding to $\hat{s}$, and the *cross-validated bagged estimator* is defined as:

$$\hat{\Psi}(P_n) = \hat{\Psi}_{\hat{S}(P_n)}(P_n).$$

This is then used to estimate the variable importance measures (VIM) described in Section 2.1.

For the prediction of survival, two steps need to be expanded in order to estimate the IPCW's:

(a) Draw bootstrap training sample $P^{0\#}_{n,B_n,b}$ from the training sample P^0_{n,B_n}

   i. Estimate (e.g., Kaplan-Meier or hazard regression) weights using the drawn bootstrap sample; input to `weight` argument

Similarly:

(a) Draw bootstrap learning sample $P^{\#}_{n,b}$ from the learning sample P_n

    i. Estimate (e.g., Kaplan-Meier or hazard regression) weights using the drawn learning sample; input to `weight` argument

Proceed with $cv-bagged-dsa$ as defined above using weights estimated from the bootstrap sample when appropriate. The process with the expanded steps that allow for estimation of weights will be referred to as $cv-bagged-survdsa$. A schematic of $cv-bagged-dsa$ for a fixed s_0 is shown in Figure 1.

4 Simulated Examples

The following illustrates the D/S/A un-bagged and bagged estimators on simulated right-censored data sets.

The first two simulated examples are based upon the following full data model: $Z = \sum_j \frac{1}{j} W_j + \varepsilon$ where W and ε are independent random variables with $W_j \sim U(0,1)$, $\varepsilon \sim N(0,\sigma^2)$ and $\sigma^2 = 0.25$. Censoring times were simulated using an exponential distribution: $C \sim E(\lambda)$ with $\lambda = 0.05$ and about 18% censoring.

A learning set with 250 observations was generated from the above model where $j = 1,\ldots,10$ for the first dataset and $j = 1,\ldots,5$ for the second dataset such that in addition to the 5 uniform variables that form the true model, there are 5 additional noise variables, $W \sim N(1,1)$.

The fine-tuning parameters, s_1 and s_2, were chosen using 5-fold cross-validation and the bagged estimate is based on 1000 bootstrap replications. $cv-bagged-survdsa$ was applied to each dataset with s_1 ranging from 1 to 10, s_2 ranging between 1 and 2, and the maximum allowed sum of powers on each tensor product is set at 2. This yields the un-bagged estimator, the bagged estimator, and variable importance curves based on the selected un-bagged and bagged fits. Note that $cv-bagged-survdsa$ specifies to form weights on $P^{\#}_n$. The simulated and real data examples were run with weights formed on P_n for simplicity.

The results are summarized in Table 1 and Figures 2 and 3. In the figures, the true variable importance curve along the with the variable importance curve calculated from the corresponding un-bagged or bagged estimator is

Table 1: *Simulated Data.* Summary Measures; RSS, R^2 and $\hat{s}$ are reported for the un-bagged (left col) and bagged (right col) estimators

	RSS		R^2		$(\hat{s}_1, \hat{s}_2)$	
sim 1	53.02	51.69	0.359	0.375	(7,1)	(3,2)
sim 2	51.40	51.04	0.380	0.384	(3,1)	(3,1)

displayed for the range of each variable. It is clear that the un-bagged and bagged estimators are comparable in terms of prediction, as in each case the bagged estimator has a slight improvement in RSS over the un-bagged estimator. In each simulation, the cross-validation selector chooses a model smaller than the truth. As a result, the bagged estimators are not that high-dimensional which results in roughly similar variable importance plots given by the un-bagged and bagged estimators. For the first simulated data example, three of the variables (W_6, W_7 and W_{10}) have an importance of zero based on the un-bagged estimator because they were not selected ($\hat{s}_1 = 7$) while the bagged estimator gives these variables a very low importance. The un-bagged estimator for the second simulated data example has three terms ($\hat{s}_1 = 3$). Therefore, only the first three variables have an importance measure greater than zero. For these three variables, the importance curves are similar for both estimators, again because the bagged fit is relatively low dimensional.

The above results were based only on a single data set. A simulation was done using 25 data sets where the true data model is simulated in the second manner described earlier. This yielded variable importance curves for the un-bagged and bagged estimator. A slope was estimated by drawing a line through each curve that passes through the intercept. These slopes were averaged across the 25 repetitions and reported in Table 3, along with the estimated variance, bias, and mean square error (MSE). The final column reports the ratio of the MSE based on the un-bagged estimator to the MSE based on the bagged estimator.

A third simulation was done where the true model involves interaction terms. The full data model is: $Z = 5W_1W_2W_3 - 4W_5 + 3W_6W_8 + 2W_{10}W_{12}W_{15} + W_{13}W_{14} + \varepsilon$ where W and ε are independent random variables with $W_j \sim U(0,1)$, $\varepsilon \sim N(0, \sigma^2)$ and $\sigma^2 = 1$. Censoring times were simulated using an exponential distribution: $C \sim E(\lambda)$ with $\lambda = 0.05$ and about 14% censoring.

A learning set with 250 observations was generated from the above model. The fine-tuning parameters, s_1 and s_2, were chosen using 5-fold cross-validation and the bagged estimate is based on 1000 bootstrap replications. $cv - bagged - survdsa$ was applied to each dataset with s_1 ranging from 1 to 10, s_2 ranging from 1 to 3, and the maximum allowed sum of powers on each tensor product is set at 3. The un-bagged ($R^2 = 0.76$, $RSS = 241.4$) and bagged ($R^2 = 0.78$, $RSS = 214.2$) estimators are again comparable in terms of prediction. The un-bagged fit only includes two-way interactions ($\hat{s}_1 = 5, \hat{s}_2 = 2$) while the bagged fit correctly allows for three-way interactions ($\hat{s}_1 = 4, \hat{s}_2 = 3$). This resulted in a bagged fit of 239 terms.

Instead of plotting the variable importance curves, Table 2 lists the variable importance at a fixed w, $|m_j(0.5) - m_j(0)|$ for the true fit, the un-bagged fit, and the bagged fit. Six of the variables have an estimated importance measure of zero based on the un-bagged fit. This is because based on the cross-validation selector these terms are not part of the final fit. It is not necessarily a reflection of the role they played in the search algorithm. On the other hand, the bagged estimator allows an estimate of importance measure for every variable.

5 Real Data Example

Data Description

Diffuse large B-cell lymphoma (DLBCL) is the most common type of lymphoma in adults (Lossos et al., 2004). Anthracycline-based chemotherapy can successfully treat only 35 to 40 percent of patients with DLBCL (Rosenwald et al., 2002). A well-established predictor of outcome in DLBCL is the International Prognostic Index (IPI) which is based on five clinical characteristics: age, tumor stage, serum lactate dehydrogenase concentration, Eastern Cooperative Oncology Group (ECOG) performance status, and number of extranodal disease sites, but the outcome in patients with DLBCL who have identical IPI values varies considerably (Lossos et al., 2004). As a result, Rosenwald et al. (2002) hypothesized that gene-expression profiles of DLBCL could be used independently of the IPI to predict survival after chemotherapy. Alizadeh et al. (2000); Shipp et al. (2002); Nguyen and Rocke (2002); Lossos et al. (2004); Bair and Tibshirani (2004); Li and Li (2004) also

Table 2: *Simulated Data.* Data Set 3. Variable Importance Measures (VIM) are reported for $w = 0.5$

W	True VIM	Un-bagged VIM	Bagged VIM
1	1.11	1.02	0.94
2	0.62	0.38	0.59
3	0.63	0.58	0.48
4	0	0.37	0.15
5	2.00	2.20	2.08
6	0.75	0.40	0.61
7	0	0.00	0.09
8	0.71	0.51	0.49
9	0	0.00	0.01
10	0.25	0.21	0.19
11	0	0.16	0.21
12	0.25	0.00	0.14
13	0.24	0.00	0.05
14	0.24	0.00	0.11
15	0.26	0.00	0.16

considered an analysis of censored survival time based on microarray gene expression profiles, and they found that it is possible to identify subgroups of patients with different survival rates based on gene expression data.

The DLBCL dataset of Rosenwald et al. (2002) consists of 7399 gene expression measures from 240 patients with untreated DLBCL who had no previous history of lymphoma. A survival time ranging between 0 and 21.8 years was recorded for each patient, where 138 of the patients died during the study (uncensored) and 102 patients were alive (censored) at the end of the study. Further description of the data is in Rosenwald et al. (2002).

The data used in our analysis was obtained directly from Bair and Tibshirani (2004) (see http://www-stat.stanford.edu/˜tibs/superpc/staudt.html). Bair and Tibshirani (2004) used the `R` function `pamr.knnimpute` to impute the missing expression data. Five of the patients had a recorded survival time of 0.0 years. The median survival time was 2.8 years, and the mean survival time was approximately 4.4 years.

Gene Signatures

Alizadeh et al. (2000) designed a specialized microarray, the *Lymphochip*, to answer questions in normal and malignant lymphocyte biology by selecting genes that are preferentially expressed in lymphoid cells and genes with known or suspected roles in processes important in cancer or immunology. For example, Alizadeh et al. (2000) hypothesized that DLBCL derives from normal B cells located within the germinal centers (GCs) of lymphoid organs and customized the Lymphochip array by enriching it with genes related to the GCs (Shipp et al., 2002). Clusters of coordinately expressed genes from the Lymphochip array were operationally defined as gene expression *signatures*. A gene expression signature is a group of genes expressed in a specific cell lineage or stage of differentiation or during a particular biologic response (Rosenwald et al., 2002). A gene expression signature was named by either the cell type in which its component genes were expressed (e.g., the *T-cell* signature) or the biological process in which its component genes were known to function (e.g., the *proliferation* signature). This allows the overall gene expression profile of a DLBCL lymph-node biopsy to be understood, at first, as a collection of gene expression signatures revealing different biological features of the sample (Alizadeh et al., 2000). Some known gene-expression signatures include the germinal-center B-cell signature, MHC class II signature, lymph-node signature, and proliferation signature (Alizadeh et al., 2000; Shaffer et al., 2001; Rosenwald et al., 2002).

Some molecular analyses of clinical heterogeneity in DLBCL have focused on individual genes. Examples include adhesion molecules (which influence the trafficking of normal activated B cells and tumor cells), proteins (which regulate apoptosis in other B cell lymphomas and normal B cell subpopulations), and angiogenic peptides (which promote the development of an effective tumor vasculature) (Shipp et al., 2002). *BAL* (B-aggressive lymphoma) has been identified based on its differential expression in fatal high-risk DLBCL and treated low-risk tumors (Shipp et al., 2002).

5.1 Previous Analyses

This section summarizes the various analyses done by four authors (Rosenwald et al., 2002; Bair and Tibshirani, 2004; Li and Li, 2004; Gui and Li, 2004) to analyze the DLBCL data set. Their approaches can be divided into two categories: methods for survival prediction and methods for identifying subgroups. Rosenwald et al. (2002) used hierarchical clustering to group the genes into *gene-expression signatures*, as described earlier. Based on those gene clusters, they built a Cox proportional hazards model to predict time to death in the patients with DLBCL. Bair and Tibshirani (2004) applied various semi-supervised methods to this data. Li and Li (2004) reduced the dimension space and then built a Cox proportional hazards model on this reduced space. Gui and Li (2004) applied a procedure they call LARS-Lasso. Note that Segal (2005) evaluates the methods that have been applied to this data set. The raw data came from Rosenwald et al. (2002) where tumor-biopsy specimens were obtained from 240 patients with untreated DLBCL who had no previous history of lymphoma. The patients were randomly divided into two groups: 160 patients in the "preliminary group" (Rosenwald et al., 2002), or training set, and 80 patients in the validation group. For each patient, the survival time was recorded along with the censoring indicator and gene-expression measures for 7399 features.

5.1.1 Identifying Subgroups

The end goal of Rosenwald et al. (2002) is to construct a predictor of survival, but their proposed method is to form subgroups with distinctive gene-expression profiles. They applied hierarchical clustering to group genes that were correlated with the outcome into gene-expression signatures. Many of the genes identified fell within known gene-expression signatures: 151 features belonged to the signature that characterizes germinal-center B-cells, 37 features were in the MHC class II signature, 357 were in the lymph-node signature, and 1333 were in the proliferation signature. The authors then combined the genes which were significantly associated with survival within each signature. To reduce the number of genes in the model to be fitted, they selected 16 genes that were highly variable in expression: three germinal-center B-cell genes, four MHC class II genes, six lymph-node genes, and three proliferation genes, and averaged the expression values for genes belonging to the same signature. Meanwhile, the authors looked at the univariate analysis, based on the training group, between survival and the individual genes that were not in the four signatures. They found that *BMP6* was univariately significant. Finally, they formed a predictor of survival and fitted a Cox proportional hazards model combining the four gene-expression signatures and *BMP6* (Rosenwald et al., 2002):

$$\begin{aligned}\lambda(t|w) &= (0.241 \times \text{proliferation signature average}) + (0.310 \times BMP6) \\ &\quad -(0.290 \times \text{germinal center B-cell signature average}) \\ &\quad -(0.311 \times \text{MHC class II signature average}) \\ &\quad -(0.249 \times \text{lymph-node signature average})\end{aligned}$$

Bair and Tibshirani (2004) developed some procedures to identify subtypes of cancer and applied them to the DLBCL data of Rosenwald et al. (2002). Their goal was to identify subtypes of cancer that are clinically relevant and biologically meaningful. The main idea of their method is to use clinical data to produce a list of genes which correlate with the outcome of interest and then apply unsupervised clustering techniques to this subset of genes. In the DLBCL dataset for example, patients' survival times are known but tumor subtypes have not been formally identified. In this case, the authors can calculate a *Cox score* which measures the correlation between the gene expression level and patient survival, and then only consider the genes with a Cox score exceeding a particular threshold. The authors proceed to describe a number of different methods such as a *supervised principal components* method and a *semi-supervised clustering* routine, and they applied these different methods as summarized in the next paragraph.

The first approach of Bair and Tibshirani (2004) was to apply an unsupervised two-means clustering procedure to the DLBCL data and compare the survival times of the two subgroups. The subgroups identified in this manner and by using hierarchical clustering did not differ with respect to survival. A separate analysis was to assign the 160 patients in the training set to a low or high risk subgroup based on survival time. They selected a model using 249 of the genes after applying nearest shrunken centroids with cross-validation. The next step was to use this model to assign each patient in the test set to a low or high risk group. Their next approach was to apply a supervised clustering procedure to the data. They ranked all of the genes based on their univariate Cox proportional hazards score and performed clustering using the 25 top-scoring genes. Lastly, they applied a *supervised principal components* method (described in the next subsection). Using the 160 training observations, they computed Cox scores for each gene and kept the 17 genes with a Cox score of 2.39 or higher. They then calculated the principal components of the training data using these 17 genes.

The survival curves for the low-risk versus high-risk group obtained by each of their methods was statistically different ($p < 0.05$) (Bair and Tibshirani, 2004).

In addition, Bair and Tibshirani (2004) discuss that sometimes a continuous predictor of survival is desired.

5.1.2 Survival Prediction

Bair and Tibshirani (2004) used a form of principal components to predict survival. They describe *supervised principal components* as a generalization of principal components regression. The first (or first few) principal components are the linear combinations of the features that capture the directions of largest variation within a dataset. However, these directions may or may not be related to an outcome variable of interest, and to find linear combinations that are related to an outcome variable, they compute univariate scores (in the case of survival, these are obtained from a proportional hazards model) for each gene and then retain only those features whose score exceeds a threshold. A principal components analysis is carried out using only the data from these selected features. Finally, these "supervised principal components" are used in a regression model to predict the outcome. Refer to Bair and Tibshirani (2004) for further description of all the methods used in their analyses.

Li and Li (2004) also analyzed the DLBCL dataset of Rosenwald et al. (2002) by using principal components analysis (PCA) and sliced inverse regression (SIR) with a Cox proportional hazards model built on the extracted linear combinations of genes. The first step is to reduce the dimension of the gene-expression data to a low-dimensional space so that a predictive survival model can be built on reduced space. Li and Li (2004) proposed to use SIR (available in `R`) to reduce the dimension, but they had to modify SIR to accomodate censoring. The covariance matrix of X needs to be non-singular in order to implement SIR, however, for microarray data, the number of genes is much larger than the number of samples n in which case the covariance matrix of X is singular. To address this problem, Li and Li (2004) adopted the idea of Chiaromonte and Martinelli (2002) to combine SIR with PCA. They obtained q principal components (PCs) based on correlations among all genes with $q < n$. They then applied SIR with principal components as input which takes into account the predictor variability and correlates the extracted linear combinations of genes with the response. Li and Li (2004) chose $q = 40$ PCs and applied SIR to these PCs. The first SIR linear combination was chosen to be sufficient in capturing the response information, and they denoted this extracted linear combination of gene expression levels with s. They then fit the following Cox proportional hazards model with s (on the 160 training patients):

$$\lambda_i(t|s_i) \quad = \quad \lambda_0(t)\,\exp(0.2418s_i - 0.0046s_i^2),$$

where $\lambda_0(t)$ is an unspecified baseline hazard function, and $\lambda_i(t|s_i)$ is the hazard function for the i-th patient. Li and Li (2004) give Kaplan-Meier survival curves for the 160 training patients and for the 80 test patients where their scores are computed based on the model for the training set. In each case, the difference between the low and high risk patients is statistically significant. This process is repeated with 5-fold cross-validation as well. The authors conclude that their method works well in distinguishing between patient survival risks.

Gui and Li (2004) used the L_1 penalized estimation for the Cox model to select genes that are relevant to survival and to build a predictive model. They call their procedure LARS-Lasso. Gui and Li (2004) also used the single data split of Rosenwald et al. (2002), 160 patients in the training sample and 80 patients in the validation sample, to analyze this data. The top ten features selected by the LARS-Lasso procedure based on the training set is given by Gui and Li (2004). Seven of these features belong to either the MHC Class II, lymph node, or germinal center signature.

Regardless of the goal, prediction of survival or identifying cancer subtypes, the general idea in all of these methods is first to reduce the data dimension (in a manner that takes into account right-censoring) and then build a predictor using the reduced data. The number of features is a tunable parameter of the D/S/A algorithm. Like some of the other methods, we can use principal components to transform and reduce the data instead of using univariate regressions at the start of the algorithm. The choice of s_0 would then correspond to the cut-off of principal components used in our search algorithm. The principles of our approach allow us to mimic other methods. The data reduction technique can be modified, and the search through polynomial regressions can be replaced by histogram regressions, for example (see Molinaro and van der Laan (2004)).

5.2 D/S/A Results

The D/S/A algorithm was used to estimate $E(Z|W)$ where W is the 7399-dimensional vector of gene expression measurements. The outcome in our analysis is $Z = \log(T+1)$ because five of the 240 patients had a recorded survival time of zero. We assumed that there was non-informative censoring in this data and used the Kaplan-Meier estimator to compute $\bar{G}_n(T|X)$. The fine-tuning parameters $s = (s_0, s_1, s_2, s_3)$ were set at different levels and $s_0, s_1,$ and s_2 were selected via 5-fold cross-validation.

- $|W| \leq s_0$ represents the number of initial gene features to be used as input in the model (dimension of vector of covariates)

- $|I| \leq s_1$ represents the number of tensor products or the size of the index sets

- $\max_{\vec{p} \in I} \sum_{j=1}^{d} I(p_j \neq 0) \leq s_2$ represents the maximum order of interaction of tensor products (the number of non-zero components in $\vec{p}$)

- $\max_{\vec{p} \in I} \sum_{j=1}^{d} p_j \leq s_3$ represents the maximum sum of powers of tensor products

Instead of selecting s_0 data-adaptively with our algorithm, we reduced the 7399 features based on a multiple testing procedure to control false discovery rate (FDR), for computational considerations. We used Cox regression models to obtain the 7399 unadjusted p-values, computed the adjusted p-values controlling FDR, and retained the 78 features with an adjusted p-value less than 0.05. Table 4 lists the adjusted p-values for $W_1, \ldots, W_{78}$.

$cv - bagged - survdsa$ was then applied using these 78 variables. The fine-tuning parameters (s_0, s_1, s_2, s_3) were set at (78,10,3,3), where s_1 and s_2 were selected via cross-validation, allowing s_1 to range between 1 and 10 and s_2 to range between 1 and 3. The bagged estimator was based on 100 bootstrap replications.

The following un-bagged estimator was selected: $1.045 - 0.989[W_{46}^3] + 0.196[W_{35}] + 0.142[W_{71}]$ $(s_0 = 78, \hat{s}_1 = 3, \hat{s}_2 = 1, s_3 = 3)$. This fit has an R^2 of 0.59 and RSS of 67.19. The corresponding bagged estimator consisted of the average across 100 bootstrap replications of the best predictor of size nine (and maximum order of interactions one) in each bootstrap replication

($\hat{s}_1 = 9, \hat{s}_2 = 1$). As a result of the constraint on interactions, this produced a relatively low-dimensional aggregated predictor with 137 terms. The terms are either main terms (e.g., w_1) or powers of up to three (e.g., w_1^2, w_1^3) because $s_3 = 3$. The bagged estimator produced a fit with an R^2 of 0.68 and RSS of 52.81. The cross-validated risks for the estimators corresponding to different choices of s_1 and s_2 are given in Table 5 (un-bagged) and 6 (bagged). These numbers provide a rough estimate of the standard error of the predictor, e.g., $\sqrt{0.35} = 0.6$. The outcome $log(T + 1)$ ranges from 0 to 3.13. Comparing Table 6 to Table 5, it is clear that the bagging forms more *stable* estimators. However, bagging did not result in an improvement in cross-validated risk. In addition to using cross-validation to select the fine tuning parameters s, we can use cross-validation to select between an un-bagged and bagged estimator. In this case, we would select the un-bagged estimator as the optimal estimator (0.281 versus 0.351).

The variable importance curve, based on the bagged estimator, is plotted for W_{35} (Figure 4). Table 7 summarizes these by displaying the slope of the importance curve for each respective variable. The five variables having the highest negative slope are W_{46}, W_{58}, W_{69}, W_{55}, and W_{71}, and the five variables having the highest positive slope are W_{37}, W_{35}, W_3, W_{54}, and W_{21}. W_{35}, W_{46}, and W_{71} were selected by the un-bagged estimator. W_{35} is in the lymph node signature; W_{55}, W_{69} and W_{71} are in the MHC class II signature; W_3 and W_{58} are in the proliferation signature of Rosenwald et al. (2002). W_{35} was one of the top ten features selected by Segal (2005) and Gui and Li (2004). W_{46} has a similar description ($|U28918|H65676|Hs.119222|suppression$) to a feature found in the proliferation signature.

6 Discussion

In this paper, we presented an algorithm that extends the previously proposed D/S/A algorithm (Sinisi and van der Laan, 2004) to handle censored data problems and extended it further as an aggregation technique (bagging). The proposed method is applied to the DLBCL data of Rosenwald et al. (2002).

Inspiration for developing the D/S/A algorithm came from the loss-based estimation methodology of van der Laan and Dudoit (2003). van der Laan and Robins (2003) handle censored data by mapping the full (uncensored) data loss function into an observed (censored) data loss function. Molinaro et al. (2004) apply these ideas to survival trees demonstrating the relationship between full data and censored data estimators. Furthermore, the IPCW loss function allows for informative censoring. This allowed for a straight-forward extension of the D/S/A algorithm for the prediction of survival.

One of the options of the D/S/A algorithm is to reduce the data based on univariate regressions to have no more than s_0 candidate covariates, where s_0 can be chosen with cross-validation. When looking at other methods used to analyze the DLBCL data, we found that often the data was reduced using principal components (PCs) and then a hazards model was estimated with the first (few) PCs. s_0 could represent the number of PCs rather than the number of original variables. It is easy to foresee that the dimension reduction can be done in other ways. Our approach is general as discussed in (van der Laan and Dudoit, 2003; Sinisi and van der Laan, 2004; Molinaro and van der Laan, 2004; Durbin et al., 2005) and based on the choice of loss function, basis functions, and sets of deletion, substitution, and addition moves. It might be worthwhile to pursue more general implementations that allow the user to decide on more options such as the strategy for dimension reduction.

van der Laan et al. (2005) introduces how to apply the cross-validation selector external to candidate bagged estimators when selecting fine-tuning parameters. This led to our proposal of the cross-validated bagged dsa estimator. When bagging is applied to survival data, it is necessary to estimate the conditional censoring survivor function with the observed data. We recommended estimating this for each bootstrap sample, but we did not do this in our analyses for computational considerations. The impact of this should be investigated with simulated experiments. A further exploration into the comparisons of un-bagged and bagged estimators is under consideration.

Acknowledgments

We would like to thank E. Bair and R. Tibshirani for making the DLBCL data set readily available; A. Hubbard and A. Molinaro for helpful discussions; M. Petersen for creating Figure One.

References

A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson Jr., L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403: 503–511, 2000.

E. Bair and R. Tibshirani. Semi-supervised methods to predict patient survival from gene expression data. *PLOS Biology*, 2(4), 2004.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996a.

L. Breiman. Heuristics of instability and stabilization in model selection. *Annals of Statistics*, 24:2350–2383, 1996b.

L. Breiman. Stacked regressions. *Machine Learning*, 24:49–64, 1996c.

L. Breiman. Using iterated bagging to debias regressions. *Machine Learning*, 45:261–277, 2001.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. The Wadsworth Statistics/Probability series. Wadsworth International Group, 1984.

P. Buhlmann and B. Yu. Analyzing bagging. *Annals of Statistics*, 30:927–961, 2002.

A. Buja and W. Stuetzle. Observations on bagging, 2002. URL `http://www-stat.wharton.upenn.edu/~buja/PAPERS/paper-bag-wxs.pdf/`.

F. Chiaromonte and J. Martinelli. Dimension reduction strategies for analyzing global gene expression data with a response. *Mathematical Biosciences*, 176:123–144, 2002.

R. B. Davis and J. R. Anderson. Exponential survival trees. *Statistics in Medicine*, 8:947–961, 1989.

B. Durbin, S. Dudoit, and M. J. van der Laan. Optimization of the architecture of neural networks using a Deletion/Substitution/Addition algorithm. Technical Report 170, Division of Biostatistics, UC Berkeley, March 2005. URL http://www.bepress.com/ucbbiostat/paper170/.

J. H. Friedman and P. Hall. On bagging and nonlinear estimation, 2000. URL http://www-stat.stanford.edu/~jhf/ftp/bag.ps/.

L. Gordon and R. Olshen. Tree-structured survival analysis. *Cancer Treatment Reports*, 69:1065–1069, 1985.

J. Gui and H. Li. Penalized Cox regression analysis in the high-dimensional and low-sample size settings with applications to microarray gene expression data. Technical report, Center for Bioinformatics and Molecular Biostatistics, University of California, San Francisco, 2004. URL http://repositories.cdlib.org/cbmb/L1Cox/.

P. Hall and R. J. Samworth. Properties of bagged nearest-neighbour classifiers. *Journal of the Royal Statistical Society: Series B*, 2005. To appear.

T. J. Hastie and R. J. Tibshirani. *Generalized Additive Models*. Chapman and Hall, 1990.

T. Hothorn, P. Buhlmann, S. Dudoit, A. M. Molinaro, and M. J. van der Laan. Survival ensembles. Technical Report 174, Division of Biostatistics, University of California, Berkeley, April 2005. URL http://www.bepress.com/ucbbiostat/paper174/.

T. Hothorn and B. Lausen. Bundling classifiers by bagging trees. *Computational Statistics and Data Analysis*, 2003.

T. Hothorn, B. Lausen, A. Benner, and M. Radespiel-Troger. Bagging survival trees. *Statistics in Medicine*, 23:77–91, 2003.

S. Keleş, M. J. van der Laan, and S. Dudoit. Asymptotically optimal model selection method with right censored outcomes. Technical Report 124, Division of Biostatistics, University of California, Berkeley, Sept. 2003. URL http://www.bepress.com/ucbbiostat/paper124/.

C. Kooperberg, C. J. Stone, and Y. K. Truong. Hazard regression. *Journal of the American Statistical Association*, 90:78–94, 1995.

M. LeBlanc and J. Crowley. Relative risk trees for censored survival data. *Biometrics*, 48:411–425, 1992.

M. LeBlanc and R. Tibshirani. Combining estimates in regression and classification. *Journal of the American Statistical Association*, 91:1641–1650, 1996.

L. Li and H. Li. Dimension reduction methods for microarrays with application to censored survival data. *Bioinformatics*, 20(18):3406–3412, 2004.

I. S. Lossos, D. K. Czerwinski, A. A. Alizadeh, M. A. Wechser, R. Tibshirani, D. Botstein, and R. Levy. Prediction of survival in diffuse large-B-cell lymphoma based on the expression of six genes. *The New England Journal of Medicine*, 350(18), 2004.

A. M. Molinaro, S. Dudoit, and M. J. van der Laan. Tree-based multivariate regression and density estimation with right-censored data. *Journal of Multivariate Analysis*, 90(1):154–177, 2004.

A. M. Molinaro and M. J. van der Laan. Deletion/Substitution/Addition algorithm for partitioning the covariate space in prediction. Technical Report 162, Division of Biostatistics, University of California, Berkeley, Nov. 2004. URL `http://www.bepress.com/ucbbiostat/paper162/`.

D. V. Nguyen and D. M. Rocke. Partial least squares proportional hazard regression for application to DNA microarray survival data. *Bioinformatics*, 18(12):1625–1632, 2002.

A. Rosenwald, G. Wright, W. C. Chan, J. M. Connors, E. Campo, R. I. Fisher, R. D. Gascoyne, H. K. Muller-Hermelink, E. B. Smeland, and L. M. Staudt. The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *The New England Journal of Medicine*, 346(25), 2002.

M. R. Segal. Regression trees for censored data. *Biometrics*, 44:35–47, 1988.

M. R. Segal. Microarray gene expression data with linked survival phenotypes: Diffuse large-B-cell lymphoma revisited. Technical report, Center for Bioinformatics and Molecular Biostatistics, University of California, San Francisco, 2005. URL `http://repositories.cdlib.org/cbmb/dlbcl/`.

A. L. Shaffer, A. Rosenwald, E. M. Hurt, J. M. Giltnane, L. T. Lam, O. K. Pickeral, and L. M. Staudt. Signatures of the immune response. *Immunity*, 15:375–385, 2001.

M. A. Shipp, K. N. Ross, P. Tamayo, A. P. Weng, J. L. Kutok, R. C. T. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G. S. Pinkus, T. S. Ray, M. A. Koval, K. W. Last, A. Norton, T. A. Lister, J. Mesirov, D. S. Neuberg, E. S. Lander, J. C. Aster, and T. R. Golub. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, 8(1):68–74, 2002.

S. E. Sinisi and M. J. van der Laan. Deletion/Substitution/Addition algorithm in learning with applications in genomics. *Statistical Applications in Genetics and Molecular Biology*, 3(1), 2004. URL `http://www.bepress.com/sagmb/vol3/iss1/art18/`. Article 18.

M. Skurichina and R. P. W. Duin. Bagging for linear classifiers. *Pattern Recognition*, 31:909–930, 1998.

M. J. van der Laan and S. Dudoit. Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: Finite sample oracle inequalities and examples. Technical Report 130, Division of Biostatistics, University of California, Berkeley, Nov. 2003. URL `http://www.bepress.com/ucbbiostat/paper130/`.

M. J. van der Laan and J. Robins. *Unified Methods for Censored Longitudinal Data and Causality.* Springer Series in Statistics. Springer, 2003.

M. J. van der Laan, S. E. Sinisi, and M. L. Petersen. Cross-validated bagged learning. Technical Report 182, Division of Biostatistics, University of California, Berkeley, June 2005. URL `http://www.bepress.com/ucbbiostat/paper182/`.

Table 3: *Simulated Data.* Data Set 2 (25 Repetitions). Variable Importance Summary (mean, variance, bias, and mean square error) Measures are reported for un-bagged (left col) vs. bagged (right col) estimator; 'ratio' is the ratio of un-bagged MSE to bagged MSE

w	true	mean		var		bias		MSE		ratio
1	1	0.975491	0.950232	0.034100	0.024090	-0.024509	-0.049768	0.034700	0.026566	1.31
2	0.50	0.430924	0.397433	0.017968	0.016736	-0.069076	-0.102567	0.022740	0.027256	0.83
3	0.33	0.245109	0.219208	0.032635	0.022762	-0.088224	-0.114126	0.040418	0.035786	1.13
4	0.25	0.131324	0.121650	0.029833	0.017976	-0.118676	-0.128350	0.043918	0.034450	1.27
5	0.20	0.067187	0.083951	0.015280	0.010214	-0.132813	-0.116049	0.032919	0.023681	1.39
6	0	0.004689	0.011620	0.000301	0.000222	0.004689	0.011620	0.000323	0.000357	0.91
7	0	0.000000	0.006331	0.000000	0.000093	0.000000	0.006331	0.000000	0.000133	0
8	0	0.005988	0.013254	0.000523	0.000415	0.005988	0.013254	0.000559	0.000591	0.95
9	0	0.006452	0.009928	0.000462	0.000431	0.006452	0.009928	0.000504	0.000530	0.95
10	0	0.005928	0.004478	0.000406	0.000091	0.005928	0.004478	0.000441	0.000111	3.98

Table 4: *DLBCL data analysis.* Adjusted p-values for Top 78 Variables. Unadjusted p-values are obtained from Cox regressions, and adjusted p-values are controlling FDR. ID refers to the unique Lymphochip identification number.

W	ID	Adj p-value	W	ID	Adj p-value	W	ID	Adj p-value
1	31242	0.0107	27	30191	0.0229	53	27872	0.0360
2	24376	0.0107	28	16832	0.0232	54	33358	0.0360
3	31981	0.0107	29	34783	0.0242	55	16988	0.0360
4	32679	0.0114	30	17591	0.0256	56	33310	0.0360
5	25116	0.0175	31	29710	0.0256	57	19255	0.0360
6	27267	0.0176	32	27612	0.0256	58	29176	0.0362
7	27774	0.0176	33	27587	0.0256	59	29222	0.0364
8	19373	0.0176	34	25054	0.0256	60	34680	0.0371
9	24396	0.0176	35	28641	0.0263	61	31681	0.0371
10	27592	0.0176	36	24220	0.0263	62	33166	0.0371
11	34805	0.0181	37	34344	0.0263	63	27718	0.0386
12	33014	0.0216	38	32238	0.0269	64	17646	0.0405
13	27573	0.0218	39	17236	0.0269	65	25092	0.0410
14	24394	0.0218	40	24725	0.0269	66	33644	0.0422
15	27585	0.0218	41	17482	0.0269	67	33585	0.0424
16	24432	0.0218	42	23872	0.0279	68	28532	0.0426
17	17259	0.0223	43	31669	0.0284	69	28197	0.0430
18	30634	0.0223	44	30272	0.0284	70	17517	0.0449
19	27766	0.0223	45	24203	0.0284	71	17273	0.0449
20	27415	0.0223	46	15937	0.0304	72	26884	0.0472
21	28328	0.0229	47	34546	0.0304	73	17154	0.0478
22	30669	0.0229	48	17241	0.0327	74	28325	0.0478
23	27731	0.0229	49	15841	0.0342	75	34500	0.0478
24	26940	0.0229	50	24377	0.0342	76	24205	0.0482
25	26528	0.0229	51	27270	0.0360	77	27218	0.0482
26	28377	0.0229	52	31806	0.0360	78	16359	0.0495

Table 5: *DLBCL data analysis.* Cross-validated risks for un-bagged estimators (v=5, np=48)

s_1/s_2	1	2	3
1	0.407	0.460	0.462
2	0.386	0.443	0.453
3	**0.281**	0.405	0.462
4	0.319	0.458	0.489
5	0.353	0.453	0.519
6	0.350	0.501	0.595
7	0.389	0.554	0.657
8	0.382	0.562	0.745
9	0.367	0.553	0.777
10	0.379	0.589	1.012

Table 6: *DLBCL data analysis.* Cross-validated risks for bagged estimators (v=5, np=48)

s_1/s_2	1	2	3
1	0.370	0.395	0.400
2	0.376	0.380	0.391
3	0.366	0.379	0.399
4	0.362	0.361	0.400
5	0.360	0.370	0.401
6	0.353	0.367	0.403
7	0.353	0.361	0.409
8	0.357	0.367	0.418
9	**0.351**	0.378	0.402
10	0.365	0.377	0.401

Table 7: *DLBCL data analysis.* Sorted Variable Importance Slopes based on Bagged Estimator

W	VIM	W	VIM	W	VIM
46	-0.1378	43	-1.38E-03	77	0.0015
58	-0.0429	76	-1.09E-03	59	0.0015
69	-0.0295	23	-1.01E-03	12	0.0018
55	-0.0197	16	-8.93E-04	2	0.0019
71	-0.0196	20	-8.75E-04	60	0.0019
73	-0.0137	39	-8.27E-04	45	0.0020
8	-0.0108	67	-4.42E-04	17	0.0024
31	-0.0104	53	-4.12E-04	56	0.0028
24	-0.0103	75	-4.07E-04	32	0.0035
34	-0.0089	9	-1.80E-04	68	0.0035
15	-0.0075	26	-1.03E-04	62	0.0037
72	-0.0061	36	-1.49E-05	10	0.0042
25	-0.0052	44	1.76E-05	40	0.0045
19	-0.0050	27	2.39E-04	6	0.0048
52	-0.0048	28	3.17E-04	63	0.0051
38	-0.0047	4	3.31E-04	11	0.0060
22	-0.0042	7	5.17E-04	5	0.0070
48	-0.0030	78	5.91E-04	49	0.0073
18	-0.0030	70	6.73E-04	50	0.0076
1	-0.0029	66	7.10E-04	47	0.0084
64	-0.0027	42	7.84E-04	29	0.0087
61	-0.0026	30	7.95E-04	21	0.0100
51	-0.0025	41	8.36E-04	54	0.0224
33	-0.0022	74	8.88E-04	3	0.0294
13	-0.0018	65	1.02E-03	35	0.0305
14	-0.0017	57	1.06E-03	37	0.0332

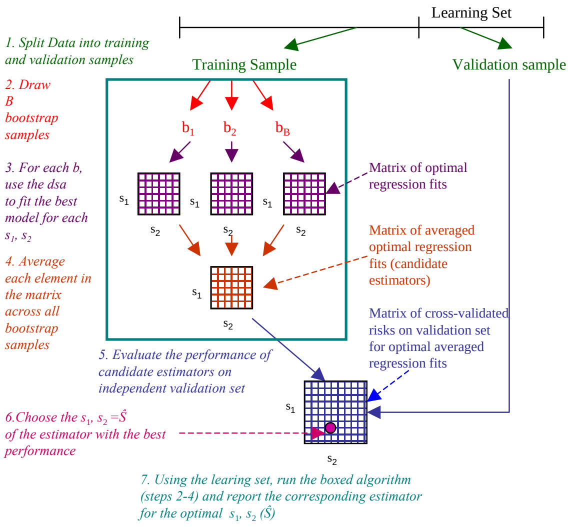

Learning Set
1. Split Data into training and validation samples
Training Sample
Validation sample
2. Draw B bootstrap samples
b1
b2
bB
3. For each b, use the dsa to fit the best model for each s1, s2
s1
s2
Matrix of optimal regression fits
4. Average each element in the matrix across all bootstrap samples
Matrix of averaged optimal regression fits (candidate estimators)
Matrix of cross-validated risks on validation set for optimal averaged regression fits
5. Evaluate the performance of candidate estimators on independent validation set
6.Choose the s1, s2 =Ŝ of the estimator with the best performance
7. Using the learing set, run the boxed algorithm (steps 2-4) and report the corresponding estimator for the optimal s1, s2 (Ŝ)

Figure 1: *cv-bagged-dsa.* Schematic of cross-validated bagged D/S/A algorithm for s_1 and s_2

Figure 2: *Simulation 1.* Variable Importance Measure against Range of w for $W_1, \ldots, W_{10}$

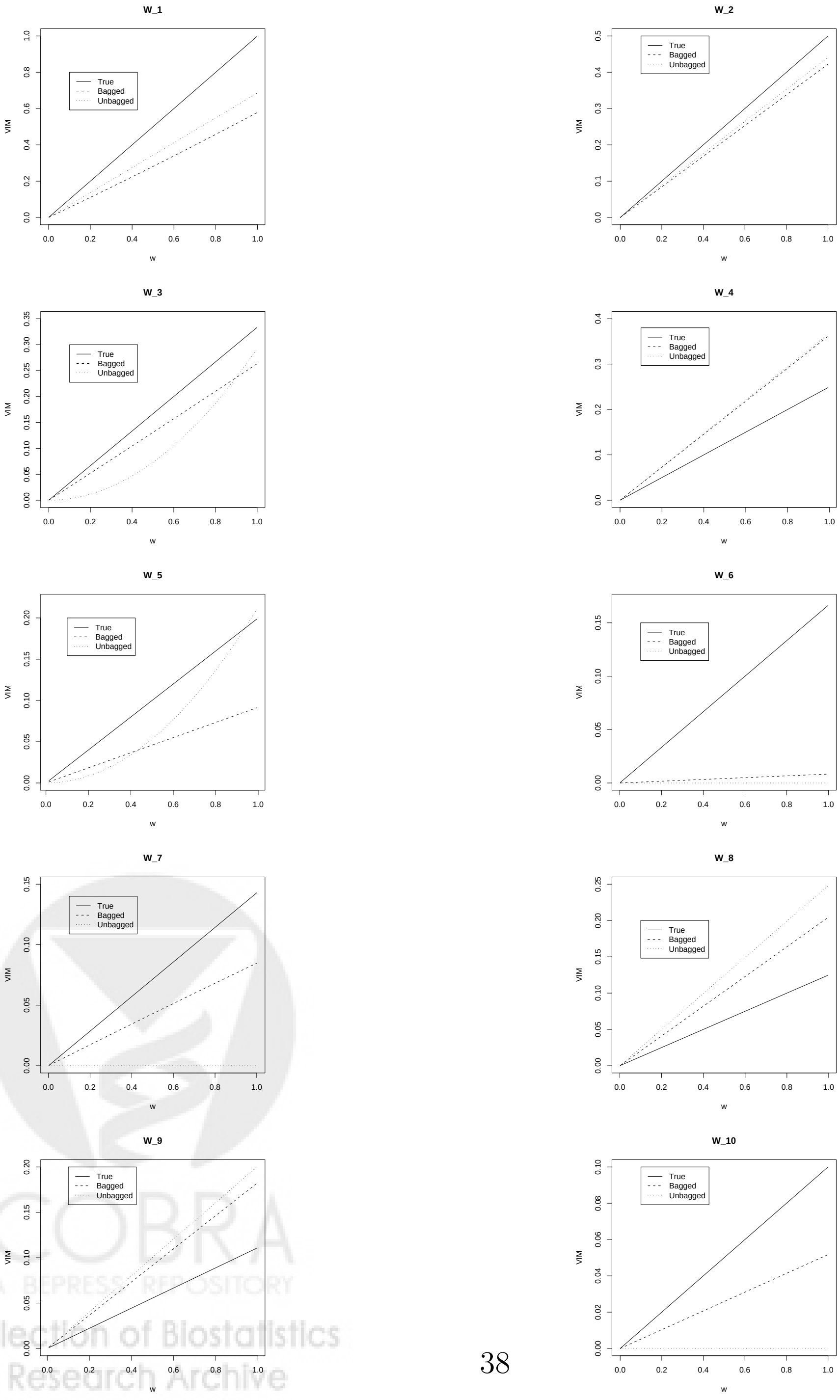


Figure 3: *Simulation 2.* Variable Importance Measure against Range of w for $W_1, \ldots, W_{10}$

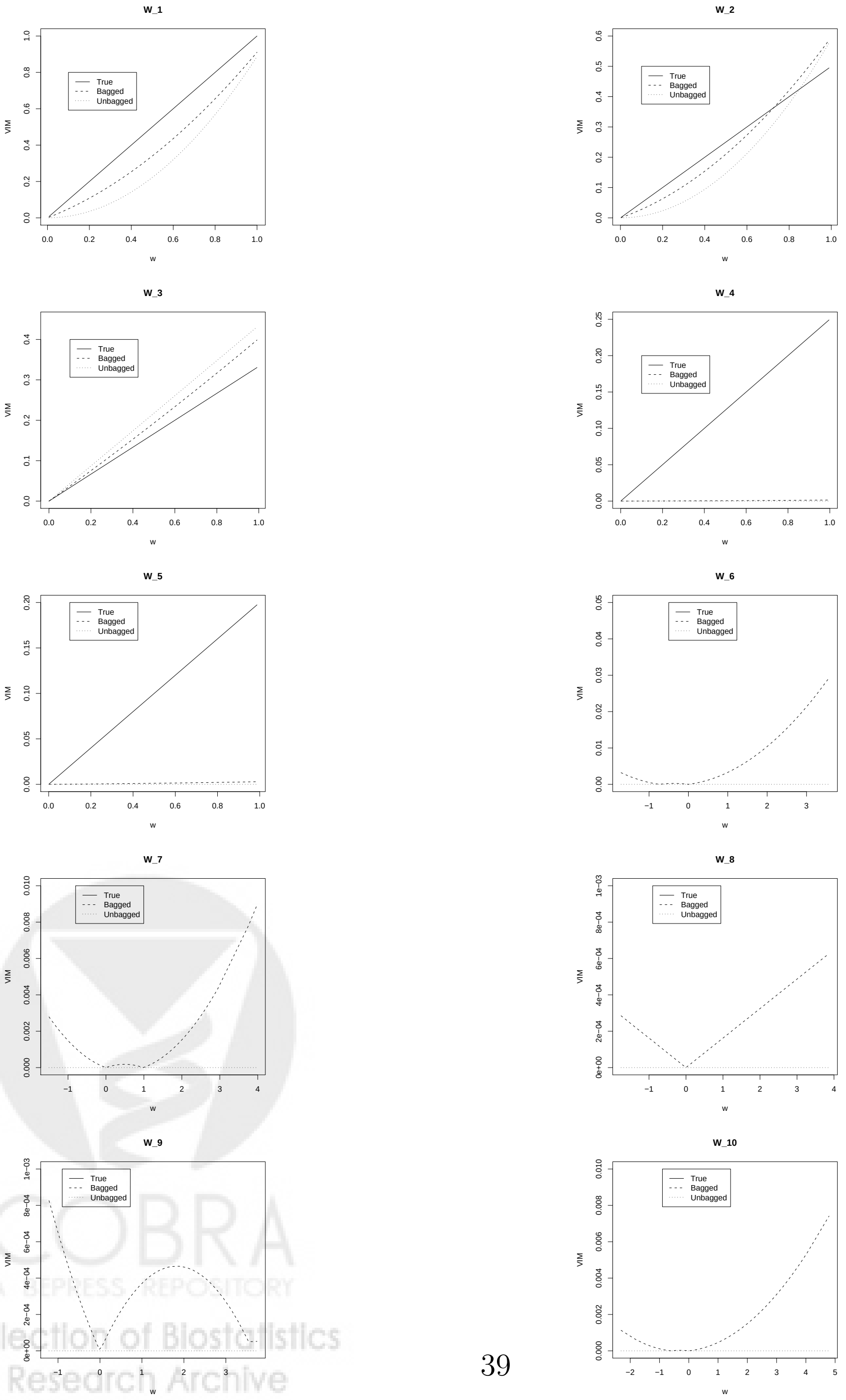
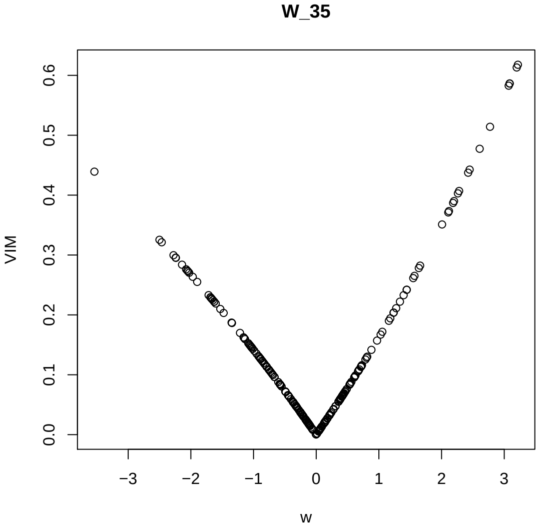


Figure 4: *DLBCL data analysis.* Variable Importance Measure against Range of w for W_{35} (Lymphochip unique id 28641)


W_35
VIM
w