University of California, Berkeley

U.C. Berkeley Division of Biostatistics Working Paper Series

Year 2003 *Paper* 137

Loss-Based Estimation with Cross-Validation: Applications to Microarray Data Analysis and Motif Finding

Sandrine Dudoit* Mark J. van der Laan† Sunduz Keles‡

Annette M. Molinaro** Sandra E. Sinisi†† Siew Leng Teng‡‡

*Division of Biostatistics, School of Public Health, University of California, Berkeley, sandrine@stat.berkeley.edu

†Division of Biostatistics, School of Public Health, University of California, Berkeley, laan@berkeley.edu

‡Division of Biostatistics, School of Public Health, University of California, Berkeley

**Division of Biostatistics, School of Public Health, University of California, Berkeley, annette.molinaro@yale.edu

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

‡‡Division of Biostatistics, School of Public Health, University of California, Berkeley

Loss-Based Estimation with Cross-Validation: Applications to Microarray Data Analysis and Motif Finding

Sandrine Dudoit, Mark J. van der Laan, Sunduz Keles, Annette M. Molinaro, Sandra E. Sinisi, and Siew Leng Teng

Abstract

Current statistical inference problems in genomic data analysis involve parameter estimation for high-dimensional multivariate distributions, with typically unknown and intricate correlation patterns among variables. Addressing these inference questions satisfactorily requires: (i) an intensive and thorough search of the parameter space to generate good candidate estimators, (ii) an approach for selecting an optimal estimator among these candidates, and (iii) a method for reliably assessing the performance of the resulting estimator. We propose a unified loss-based methodology for estimator construction, selection, and performance assessment with cross-validation. In this approach, the parameter of interest is defined as the risk minimizer for a suitable loss function and candidate estimators are generated using this (or possibly another) loss function. Cross-validation is applied to select an optimal estimator among the candidates and to assess the overall performance of the resulting estimator. This general estimation framework encompasses a number of problems which have traditionally been treated separately in the statistical literature, including multivariate outcome prediction and density estimation based on either uncensored or censored data. This article provides an overview of the methodology and describes its application to two problems in genomic data analysis: the prediction of biological and clinical outcomes (possibly censored) using microarray gene expression measures and the identification of regulatory motifs (i.e., transcription factor binding sites) in DNA sequences.

Contents

1 Introduction

1.1 Motivation

Our general estimation methodology was motivated by current statistical inference problems in the analysis of genomic data, such as: the prediction of biological and clinical outcomes (possibly censored) using microarray gene expression measures, the identification of regulatory motifs (i.e., transcription factor binding sites) in DNA sequences, and the genetic mapping of complex traits using single nucleotide polymorphisms (SNP).

Prediction of biological and clinical outcomes using microarray measures. *Microarrays* are high-throughput biological assays that can be used to measure the abundance of nucleic acids (DNA or RNA) on a genomic scale in different biological samples. In cancer research, for example, interest is in relating microarray measures of gene expression to biological and clinical outcomes in order to gain a more thorough understanding of the molecular basis of the disease and eventually develop better diagnosis and treatment strategies. *Outcomes* (phenotypes) of interest include tumor class, response to treatment, patient survival, and can be either polychotomous or continuous, censored or uncensored. *Explanatory variables* (genotypes), or *features*, include measures of transcript (i.e., mRNA) levels or DNA copy number for thousands of genes, treatment, epidemiological and histopathological variables. An important and immediate question is the choice of a good *predictor*, i.e., a function of the explanatory variables that has low error (i.e., risk) when used to predict the outcome. Should one use linear discriminant analysis, trees, support vector machines (SVMs), neural networks, or some other approach to construct this predictor? Predictor selection includes the related problem of *variable selection*, or *feature selection*, that is, the identification of a subset of marker genes to be used in the predictor function. Estimator selection problems in microarray data analysis follow the so-called "small n, large p" paradigm: thousands of explanatory variables are measured for each observational unit (e.g., patient), but the sample sizes available for estimation purposes are comparatively small.

Identification of regulatory motifs in DNA sequences. *Transcription factors* (TF) are proteins that selectively bind to DNA to regulate gene expression. Transcription factor *binding sites*, or *regulatory motifs*, are short

DNA sequences (5–25 base pairs) in the transcription control region of genes, i.e., in regions roughly 600–1,000 base pairs upstream of the gene start site in lower eukaryotes such as yeast. From *unaligned* DNA sequence data, the motif finding problem involves estimating motif start sites in individual sequences and the motif *position specific weight matrix* (PWM) (i.e., the distribution of bases at each position in the binding site). Common approaches to this problem are based on *maximum likelihood estimation* under a particular model for the distribution of bases in the motif and background sequences (Bailey and Elkan, 1994; Keleş et al., 2003b; Lawrence and Reilly, 1990). Another approach involves the *prediction* of gene expression levels based on sequence features, such as pentamers (and their interactions), and relies on cross-validation to identify motifs with good predictive power for gene expression levels (Keleş et al., 2002). Both approaches entail the selection of a *good* (i.e., low risk for a suitable loss function) model for transcription factor binding sites. For instance, in likelihood-based methods such as `COMODE` (Keleş et al., 2003b), model selection questions concern the distribution of bases in the motif (constraints on PWM), the distribution of bases in the background sequences, the motif width, the number of motifs per sequence.

A dominating feature in the above and other statistical inference problems in genomic data analysis is that they involve parameter estimation for high-dimensional multivariate distributions, with typically unknown and intricate correlation patterns among variables. Accordingly, statistical models for the data generating distribution correspond to large parameter spaces. For instance, for the prediction of clinical outcomes using microarray measures of gene expression, the parameter space may consist of the set of all possible linear combinations of tensor products of univariate polynomial basis functions of the explanatory variables (i.e., thousands of gene expression measures), in order to allow for higher order interactions among these variables. Even if it were possible to minimize a suitable error measure (empirical or cross-validated risk) over the entire parameter space, the resulting estimators would be too variable and ill-defined. Instead, we approximate the parameter space by a sequence of subspaces of increasing dimension and generate candidate estimators for each subspace. This approach therefore requires: (i) an intensive and thorough search of the parameter space to generate good candidate estimators; (ii) a procedure for selecting an optimal estimator among these candidates; and (iii) a method for reliably assessing the performance of the resulting estimator.

1.2 Estimation road map

Parameter estimation problems can be formulated generally and abstractly as follows. The data consist of realizations of random variables, $X_1, \ldots, X_n$, from an unknown *data generating distribution*, $F_{X,0}$. The goal is to use these data to estimate a *parameter* ψ_0 of the distribution $F_{X,0}$, where ψ_0 is defined as some function of $F_{X,0}$. That is, we wish to obtain an *estimator*, or function of the data, $\hat{\psi}$, that is *close* (in risk distance) to the parameter ψ_0. For example, in cancer microarray studies, X_i could consist of a pair $X_i = (W_i, Z_i)$, measured on a patient i, where W_i is a $d = 5000$-dimensional vector of microarray measures and Z_i is a possibly censored survival time. The parameter of interest ψ_0 could correspond to the $d \times d$ correlation matrix for the gene expression vector W or to the conditional expected value of the survival time Z given the gene expression vector W.

Our general strategy for data-adaptive estimation is driven by the choice of a *loss function* and relies on *cross-validation* for estimator selection and performance assessment. Our proposed estimation road map, which covers *censored data* situations, can be stated in terms of the following three main steps. Section 2 elaborates on each of these steps.

1. *Definition of the parameter of interest in terms of a loss function.* For the full data structure, define the parameter of interest as the minimizer of the expected loss, or *risk*, for a *loss function* chosen to represent the desired measure of performance (e.g., mean squared error in regression, entropy in density estimation). In censored data situations, apply the general *estimating function* methodology of van der Laan and Robins (2002) to map the *full, uncensored* data loss function into an *observed, censored* data loss function having the same expected value and leading to an efficient estimator of this risk (Section 2.2).

2. *Construction of candidate estimators based on a loss function.* Define a finite collection of candidate estimators for the parameter of interest based on a *sieve* of increasing dimension approximating the complete parameter space. For each element of the sieve, the candidate estimator is chosen as the minimizer of the empirical risk based on the observed data loss function (e.g., tree-based methods, D/S/A algorithm in Section 2.3 with the empirical risk as the objective function).

3. *Cross-validation for estimator selection and performance assessment.* Use cross-validation to estimate risk based on the observed data loss function and to select an optimal estimator among the candidates in Step 2 (Sections 2.4 and 2.5).

Note that we use the term *estimation* in a broad sense, to provide a unified treatment of multivariate prediction and density estimation based on either uncensored or censored data. Each of these problems can be dealt with according to the road map by the choice of a suitable loss function. A number of common estimation procedures follow the road map in the full data situation, but depart from it when faced with the obstacle of evaluating the loss function in the presence of censoring (e.g., classification and regression trees, where candidates in Step 2 are obtained by recursive binary partitioning of the covariate space). Here, we argue that one can, and should, also adhere to the above estimation road map in censored data situations. All that is required is to replace the full (uncensored) data loss function by an observed (censored) data loss function with the same expected value, i.e., with the same risk.

We note also that existing methods for Step 2 are not aggressive enough for the types of datasets encountered in genomics. In order to account for higher-order interactions among many variables (e.g., thousands of gene expression measures in microarray experiments), one needs to consider large parameter spaces. However, standard approaches either only accommodate variable main effects or are too rigid to generate a good set of candidate estimators. For example, while regression trees allow interactions among variables, the candidate tree estimators are generated according to a limited set of moves, amounting to forward selection (node splitting) followed by backward elimination (tree pruning). Instead, we recommend more aggressive and flexible algorithms, such as the D/S/A algorithm of (Molinaro and van der Laan, 2003; Sinisi and van der Laan, 2003), that at each step allow not only node splitting, but also node collapsing and substitutions (Section 2.3).

1.3 Outline

The present article provides an overview of our general data-adaptive loss-based estimation methodology with cross-validation and describes applica-

tions to the analysis of genomic data. Section 2 discusses the main features of our estimation road map, including the choice of a loss function (Step 1), the new D/S/A algorithm for generating candidate estimators (Step 2), cross-validation estimator selection and performance assessment (Step 3), and a novel loss-based approach for measuring variable importance. Sections 3 and 4 describe applications of the methodology to the prediction of biological and clinical outcomes (possibly censored) using microarray gene expression measures and the identification of regulatory motifs in DNA sequences.

Our general framework and its theoretical foundations are established in van der Laan and Dudoit (2003). This earlier manuscript proposes a unified cross-validation methodology for estimator construction, selection and performance assessment, and in particular provides finite sample results and asymptotic optimality results concerning cross-validation estimator selection for general data generating distributions, loss functions (possibly depending on a nuisance parameter), and estimators. These new theoretical results have the important practical implication that cross-validation selection can be used in intensive searches of large parameter spaces, even in finite sample situations. Special cases and applications are described in a collection of related articles: estimator selection and performance assessment based on uncensored data (Dudoit and van der Laan, 2003), estimator selection with censored data (Keleş et al., 2003a), likelihood-based cross-validation (van der Laan et al., 2003a), tree-based estimation with censored data (Molinaro et al., 2004), D/S/A algorithm for generating candidate estimators (Molinaro and van der Laan, 2003; Sinisi and van der Laan, 2003), supervised detection of regulatory motifs in DNA sequences (Keleş et al., 2003b).

2 Methods

2.1 Model

2.1.1 Full data structure

In many applications of interest, the *full data structure* will simply consist of a pair, $X = (W, Z)$, where $W = (W_1, \ldots, W_d)$ is a d-vector of *explanatory variables* (e.g., microarray expression measures for thousands of genes) and Z is a scalar *outcome* (e.g., survival time, tumor class, quantitative phenotype). However, to cover general estimation problems, we define the full data

structure as a stochastic process, $X \equiv \bar{X}(T) = \{X(t) = (R(t), L(t)) : 0 \leq t \leq T\}$, where T denotes either a fixed endpoint or a random survival time, $R(t) \equiv \mathrm{I}(T \leq t)$, and $L(t)$ is a covariate process. Denote the distribution of the full data structure X by $F_{X,0}$, where $F_{X,0}$ is assumed to belong to a certain model $\mathcal{M}^F$, possibly non-parametric and very large. The covariate process $L(t)$ may contain time-dependent and time-independent covariates. Denote the time-independent covariates by $W = L(0)$, a d-dimensional vector, measured at baseline. For random T, let $Z = \log T$ denote the log survival time. For fixed T, one may also be interested in monitoring $Z(t)$, $t \in \{t_0 = 0, \ldots, t_{m-1} = T\}$, an m-dimensional outcome process included in $X(t)$, such as T-cell counts at different timepoints.

2.1.2 Observed data structure

In the observed data world, one rarely sees all of the relevant variables in the process $X = \bar{X}(T) = \{X(t) : 0 \leq t \leq T\}$. Rather, one observes the full data process $X(t)$ up to the minimum, $\tilde{T} \equiv \min(T, C)$, of the survival time T and a univariate *censoring variable* C. In a clinical setting, this missing, or censored, survival data situation can be due to drop-out or the end of follow-up. The *observed data structure* can be written as $O \equiv (\tilde{T}, \Delta, \bar{X}(\tilde{T}))$, where Δ is the *censoring indicator*, $\Delta \equiv \mathrm{I}(T \leq C)$, equal to one for uncensored observations and to zero for censored observations.

The random variable O for the observed data has a distribution $P_0 = P_{F_{X,0},G_0}$, indexed by the full data distribution, $F_{X,0}$, and the conditional distribution, $G_0(\cdot|X)$, of the censoring time C given full data X. The survivor function for the censoring mechanism is denoted by $\bar{G}_0(c \mid X) \equiv Pr_0(C > c \mid X)$ and referred to as *censoring survivor function*. We make the standard *coarsening at random* (CAR) assumption for the censoring mechanism. If $X = (W, Z)$, that is, X does not include time-dependent covariates, then, under CAR, the censoring time C is conditionally independent of the survival time T given baseline covariates W. Thus, $\bar{G}_0(c \mid X) = \bar{G}_0(c \mid W)$ and the censoring survivor function only depends on the observed baseline covariates W. Gill et al. (1997), van der Laan and Robins (2002) (Section 1.2.3, in particular), and Robins and Rotnitzky (1992) provide further, thorough explanations of CAR.

2.2 Loss functions (Step 1)

2.2.1 Full data loss function

Parameters. For distributions $F_X \in \mathcal{M}^F$, define *parameters* $\psi \equiv \Psi(F_X)$ in terms of a mapping, $\Psi : \mathcal{M}^F \to \mathcal{D}(\mathcal{S})$, from the model $\mathcal{M}^F$ into a space $\mathcal{D}(\mathcal{S})$, where elements of $\mathcal{D}(\mathcal{S})$ are functions from a Euclidean space $\mathcal{S}$ into the real line $I\!R$. Thus, a parameter $\psi = \Psi(F_X)$ is itself a mapping $\psi : \mathcal{S} \to I\!R$. Let $\mathbf{\Psi} \equiv \{\psi = \Psi(F_X) : F_X \in \mathcal{M}^F\}$ denote the corresponding *parameter space*. Note that the use of upper case Ψ and lower case ψ allows us to distinguish between two types of mappings: the mapping $\Psi : \mathcal{M}^F \to \mathcal{D}(\mathcal{S})$, that defines a parameter $\psi = \Psi(F_X)$ for a particular distribution F_X, and the mapping (i.e., realization) $\psi : \mathcal{S} \to I\!R$, corresponding to this parameter. For example, for a full data structure $X = (W, Z)$, the space $\mathcal{S}$ is typically a subset of $I\!R^d$, corresponding to the explanatory variables W, and the parameter for a distribution F_X could be defined as the conditional expected value of the response Z given W, $\psi(W) = \Psi(F_X)(W) = E_{F_X}[Z \mid W]$.

Estimators. Assume that we have a sample, or *learning set*, of n independent and identically distributed (i.i.d.) observations, $X_1, \ldots, X_n$, from the distribution $F_{X,0} \in \mathcal{M}^F$. Let P_n denote the *empirical distribution* of $X_1, \ldots, X_n$, where P_n places probability $1/n$ on each realization X_i. Our goal is to use the sample to estimate the parameter $\psi_0 \equiv \Psi(F_{X,0})$, corresponding to the unknown *data generating distribution* $F_{X,0}$. An *estimator* $\hat{\psi} \equiv \hat{\Psi}(P_n)$ is simply a function of the empirical distribution P_n, that is, an *algorithm* one can apply to the data $X_1, \ldots, X_n$.

Loss functions and risk. We define a full data *loss function*, $L(X, \psi)$, such that its expected value, or *risk*, under $F_{X,0}$ is minimized at the parameter ψ_0. That is, ψ_0 is such that

$$\begin{aligned} \theta_0 \equiv E_{F_{X,0}}[L(X, \psi_0)] &= \int L(x, \psi_0) dF_{X,0}(x) \qquad (1) \\ &\equiv \min_{\psi \in \mathbf{\Psi}} \int L(x, \psi) dF_{X,0}(x) \\ &= \min_{\psi \in \mathbf{\Psi}} E_{F_{X,0}}[L(X, \psi)]. \end{aligned}$$

To simplify notation, we may use the subscript 0 to refer to parameters of the underlying data generating distributions $F_{X,0}$ (and G_0 in censored data

situations), that is, write $E_{F_{X,0}}[L(X,\psi)] = E_0[L(X,\psi)]$. Note that we do not require uniqueness of the risk minimizer, rather, we simply assume that there is a loss function whose risk is minimized by the parameter of interest ψ_0. In addition, depending on the parameter of interest, there could be numerous loss functions from which to choose and one should adopt the loss function that corresponds to the desired measure of performance for the estimation of ψ_0. Loss functions for common estimation problems are listed in Table 4. For instance, regression problems typically involve minimizing risk for the squared error loss function (a.k.a. mean squared error), while classification often involves minimizing risk for the indicator loss function (a.k.a. classification error), and density estimation is concerned with minimizing risk for the negative log-likelihood loss function (a.k.a. entropy).

Risk estimation. Since the data generating distribution $F_{X,0}$ is typically unknown, one cannot directly minimize risk as in equation (1). That is, the *conditional risk*,

$$\tilde{\theta}_n \equiv \int L(x, \hat{\Psi}(P_n)) dF_{X,0}(x), \tag{2}$$

for an estimator $\hat{\psi} = \hat{\Psi}(P_n)$, is typically unknown and needs to be estimated from the data, i.e., using the empirical distribution P_n. A naive risk estimator is the *empirical risk*, or *resubstitution estimator*, where the unknown $F_{X,0}$ is simply replaced by the empirical P_n

$$\hat{\theta}_n \equiv \int L(x, \hat{\Psi}(P_n)) dP_n(x) = \frac{1}{n} \sum_{i=1}^{n} L(X_i, \hat{\Psi}(P_n)). \tag{3}$$

However, it is well-known that estimator construction and selection methods aimed at optimizing the empirical risk do not produce estimators that minimize the true unknown risk and often suffer from over-fitting, i.e., are too data-adaptive. Instead, we turn to *cross-validation* to provide consistent risk estimators for use in estimator selection and performance assessment. As detailed in Section 2.4, the *cross-validated risk estimator* is obtained by constructing estimators $\hat{\Psi}(P^0_{n,S_n})$ on training sets of size $n(1-p_n)$ and estimating risk based on empirical distributions P^1_{n,S_n} for validation sets of size np_n (in place of the unknown $F_{X,0}$, as in equation (2))

$$\hat{\theta}_{n(1-p)} \equiv E_{S_n} \int L(x, \hat{\Psi}(P^0_{n,S_n})) dP^1_{n,S_n}(x). \tag{4}$$

In this representation, E_{S_n} indicates averaging of the validation set risks over the different splits of the learning set into training and validation sets. Thus, the risk definitions in equations (2), (3), and (4), differ in the choice of distributions for constructing the estimator (empirical distributions P_n or P^0_{n,S_n}) and for evaluating the loss function ($F_{X,0}$, P_n, or P^1_{n,S_n}).

Example 1. Prediction. In univariate (in the outcome) prediction problems, the full data structure is $X = (W, Z) \sim F_{X,0} \in \mathcal{M}^F$, where W is a d-dimensional vector of explanatory variables and Z a scalar outcome. For continuous outcomes Z, the parameter of interest is typically the *conditional expected value*, $\psi_0(W) = \Psi(F_{X,0})(W) = E_{F_{X,0}}[Z \mid W]$, of the outcome given the explanatory variables. The parameter space is $\mathbf{\Psi} = \{\Psi(F_X) : \Psi(F_X)(W) = E_{F_X}[Z \mid W], F_X \in \mathcal{M}^F\}$ and the loss function is the *squared error loss function*, $L(X, \psi) = (Z - \psi(W))^2$. The familiar *ordinary least squares* (OLS) regression approach corresponds to minimizing the empirical risk for the squared error loss function over linear combinations of individual explanatory variables.

Specifically, OLS considers a reduced model, $\mathcal{M}^{OLS} = \{F_X : E_{F_X}[Z \mid W] = W\beta, \beta \in \Re^d\}$, and parameter space, $\mathbf{\Psi}^{OLS} = \{\psi_\beta : \psi_\beta(W) = W\beta, \beta \in I\!R^d\}$. The parameter of interest is ψ_0, where $\psi_0(W) = \Psi(F_{X,0})(W) = E_{F_{X,0}}[Z \mid W] = \beta_0 W$ and the regression coefficient β_0 is such that

$$\beta_0 = \underset{\beta}{\operatorname{argmin}} \int L(x, \psi_\beta) dF_{X,0}(x) = \underset{\beta}{\operatorname{argmin}} \int (z - w\beta)^2 dF_{X,0}(x).$$

The OLS estimator $\hat{\beta}_{OLS}$ of the regression coefficient β_0 minimizes the empirical risk for the squared error loss function, i.e., the *mean squared error* (MSE),

$$\hat{\beta}_{OLS} = \underset{\beta}{\operatorname{argmin}} \int (z - w\beta)^2 dP_n(x) = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (Z_i - W_i\beta)^2,$$

where P_n denotes the empirical distribution for the data $X_1 = (W_1, Z_1), \ldots, X_n = (W_n, Z_n)$, that places probability $1/n$ on each realization X_i. In this article, we consider much broader (non-parametric) models $\mathcal{M}^F$, that correspond to general functions $\Psi(F_X)$ for the conditional expected values $E_{F_X}[Z \mid W]$ and allow, in particular, higher order interactions among explanatory variables

(Section 2.3).

Example 2. Density estimation. Likewise, the widely used *maximum likelihood estimation* (MLE) framework involves minimizing the empirical risk for the *negative log-likelihood loss function* over densities $\psi \in \mathbf{\Psi} = \{\psi_\lambda : \lambda \in \Lambda\}$, indexed by a parameter $\lambda \in \Lambda$. For example, ψ_λ could represent the density for the multivariate normal distribution $N(\lambda, I_d)$, with mean vector λ and $d \times d$ identity covariance matrix I_d. For full data structure $X \sim \psi_{\lambda_0}$ and loss function $L(X, \psi) = -\log \psi(X)$, the parameter of interest is

$$\lambda_0 = \underset{\lambda \in \Lambda}{\operatorname{argmin}} - \int \log \psi_\lambda(x) dP_0(x) = \underset{\lambda \in \Lambda}{\operatorname{argmin}} - \int \log \psi_\lambda(x) \psi_{\lambda_0}(x) dx.$$

(For simplicity, we define densities ψ with respect to the Lebesgue measure. However, one could define densities more generally, with respect to some dominating measure μ, to accommodate discrete distributions as well.) The MLE then minimizes the negative log-likelihood function

$$\begin{aligned} \hat{\lambda}_{MLE} &= \underset{\lambda \in \Lambda}{\operatorname{argmin}} - \int \log \psi_\lambda(x) dP_n(x) \\ &= \underset{\lambda \in \Lambda}{\operatorname{argmin}} - \log \prod_{i=1}^{n} \psi_\lambda(X_i). \end{aligned}$$

2.2.2 Observed data loss function

In the observed (censored) data world, we have a learning set of n 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 P_n denote the *empirical distribution* of $O_1, \ldots, O_n$, where P_n places probability $1/n$ on each realization O_i. The goal remains to find an estimator for a parameter ψ_0 defined in terms of the risk for a full data loss function $L(X, \psi)$, e.g., a predictor of the log survival time Z based on covariates W. An immediate problem is that loss functions such as the quadratic loss, $L(X, \psi) = (Z - \psi(W))^2$, cannot be evaluated for an observation O with censored survival time, i.e., for which $Z = \log T$ is not observed ($\Delta = 0$). Risk estimators based on only uncensored observations, such as $\frac{1}{n} \sum_i L(X_i, \psi) \Delta_i$, are *biased* for $E_0[L(X, \psi)]$ and, in particular, estimate instead the quantity $E_0[L(X, \psi) \bar{G}_0(T|X)]$ which is not minimized by the parameter of interest ψ_0.

The general *estimating function* methodology of van der Laan and Robins (2002) can be used to link the observed data world to the full data world. The general solution is to replace the *full (uncensored) data loss function*, $L(X, \psi)$, by an *observed (censored) data loss function*, $L(O, \psi \mid \upsilon_0)$, with the same expected value, i.e., with the *same risk*,

$$\int L(o, \psi \mid \upsilon_0) dP_0(o) = \int L(x, \psi) dF_{X,0}(x). \tag{5}$$

As detailed below, υ_0 denotes a nuisance parameter for the data generating distribution $P_0 = P_{F_{X,0},G_0}$. A simple observed data loss function is the *inverse probability of censoring weighted* (IPCW) loss function

$$L(O, \psi \mid \upsilon_0) \equiv L(X, \psi) \frac{\Delta}{\bar{G}_0(T|X)}, \tag{6}$$

where $\Delta = \mathrm{I}(T \leq C)$ is the censoring indicator and $\upsilon_0 = \Upsilon(P_0)$ denotes the nuisance parameter corresponding to the censoring survivor function, $\bar{G}_0$, for the censoring time C given full data X. Under the coarsening at random (CAR) assumption, $\bar{G}_0(T|X) = \bar{G}_0(T|W)$ only depends on the observed data and can be estimated, for example, using the Cox proportional hazards model. When an estimator $\bar{G}_n$ is used in place of the true unknown $\bar{G}_0$, the IPCW estimating function provides a consistent risk estimator under the following conditions: (i) $\bar{G}_0(T|X) > \delta > 0$, $F_{X,0}$-a.e., for some $\delta > 0$, and (ii) $\bar{G}_n$ is a consistent estimator for $\bar{G}_0$. For an uncensored observation ($\Delta = 1$), the IPCW observed data loss function is simply the full data loss function weighted by the inverse of the probability $\bar{G}_0(T|W)$ of no censoring before T; for a censored observation ($\Delta = 0$), the loss function is zero. We stress that in the absence of censoring, i.e., when $\bar{G}_0(t|w) \equiv 1 \; \forall t$, the IPCW observed data loss function reduces to the full data loss function, $L(O, \psi \mid \upsilon_0) = L(X, \psi)$. This ensures that the censored and full data estimators coincide when there is no censoring.

Example 1. Prediction. In the case of the squared error loss function used in regression, the empirical risk based on the IPCW loss function becomes a *weighted mean squared error* (MSE)

$$\int L(o, \psi \mid \hat{\upsilon}_n) dP_n(o) = \frac{1}{n} \sum_{i=1}^{n} (Z_i - \psi(W_i))^2 \frac{\Delta_i}{\bar{G}_n(T_i|W_i)},$$

where $\hat{v}_n = \hat{\Upsilon}(P_n)$ represents $\bar{G}_n$, an estimator of the nuisance parameter $\bar{G}_0$ derived under the CAR assumption for the censoring mechanism.

Example 2. Density estimation. Similarly, for the negative log-likelihood loss function used in density estimation, the empirical risk based on the IPCW loss function is

$$\int L(o, \psi \mid \hat{v}_n) dP_n(o) = -\frac{1}{n} \sum_{i=1}^{n} \log \psi(X_i) \frac{\Delta_i}{\bar{G}_n(T_i|W_i)}.$$

The ability to map from a full data loss function into an observed data loss function with the same risk offers several important practical advantages. Firstly, this allows us to directly extend full data estimation methodology to censored data situations. This in contrast to common censored data estimation approaches, such as survival trees, which bypass the risk estimation problem for censored outcomes by altering the node splitting, tree pruning, and performance assessment criteria in manners that are specific to censored survival data (Molinaro et al., 2004). In general, the splitting and pruning criteria seem to be chosen based on convenience for handling censored data and do not reduce to the preferred choice for uncensored data. Most tree-based regression and density estimation procedures rely on the negative log-likelihood loss function, with the explicit or implicit goal of estimating the conditional survivor function given explanatory variables, and differ mainly in their choice of model for the observed data likelihood within nodes. This general difficulty in evaluating risk for censored observations results in a discontinuity between the full and observed data worlds. Secondly, as shown in Molinaro et al. (2004), gains in accuracy can be achieved by employing a loss function that is specific to the parameter of interest (e.g., by using the squared error loss function for regression rather than the negative log-likelihood loss function typically used in survival trees). Finally, the IPCW estimating function approach allows us to assess performance on censored data for arbitrary loss functions. Current methods typically rely on the negative log-likelihood loss function or lead to biased risk estimators by ignoring censored observations altogether.

2.3 Generating candidate estimators based on loss function (Step 2)

Having defined the parameter of interest in Step 1, as the risk minimizer for a particular loss function, Step 2 of the road map is concerned with generating a sequence of candidate estimators by minimizing the empirical risk (for the same or possibly another loss function as in Step 1) over subspaces of increasing dimension approximating the complete parameter space. In general, it is not feasible to consider all possible elements of the subspaces and one needs an efficient search algorithm for optimizing risk over these subspaces. Tree-structured estimators, such as CART (Breiman et al., 1984), correspond to one such procedure, whereby candidate estimators are obtained by recursive binary partitioning of the covariate space. However, as discussed in Molinaro et al. (2004) and Molinaro and van der Laan (2003), trees do not provide an exhaustive enough search of the subspaces: the candidate estimators are generated according to a limited set of moves, amounting to forward selection (node splitting) followed by backward elimination (tree pruning). Instead, we favor more aggressive and flexible algorithms, such as the D/S/A algorithm of Molinaro and van der Laan (2003) and Sinisi and van der Laan (2003), that at each step allow not only node splitting, but also node collapsing and substitutions.

2.3.1 Parameterization of the parameter space using linear combinations of basis functions

Consider a full data structure of the form $X = (W, Z)$, where W is a d-vector of explanatory variables and Z is a possibly multivariate outcome. Define a countable set of *basis functions*, $\{\phi_j : j \in I\!N\}$, indexed by the non-negative integers $I\!N$, such that every parameter $\psi \in \mathbf{\Psi}$ can be arbitrarily well approximated by finite linear combinations of these basis functions (or some known function of linear combinations of basis functions, such as the logit function in binary classification or the exponential function in the Cox proportional hazards model). That is, define *regression functions*, $\psi_{I,\beta}$, as

$$\psi_{I,\beta}(\cdot) \equiv \sum_{j \in I} \beta_j \phi_j(\cdot), \tag{7}$$

where $I \in \mathcal{I}$ denotes a countable *index set* and $\mathcal{I}$ is a collection of subsets of $I\!N$. For a given index set $I \in \mathcal{I}$, the *regression coefficients* $\beta = (\beta_1, \ldots, \beta_{|I|})$

are assumed to belong to $B_I \equiv \{\beta : \psi_{I,\beta} \in \mathbf{\Psi}\} \subseteq I\!R^{|I|}$.

The choice of basis functions $\{\phi_j : j \in I\!N\}$ depends on the estimation approach. In polynomial regression, the ϕ_j are polynomial functions of the explanatory variables (Sinisi and van der Laan, 2003). In histogram regression (e.g., regression trees), for a given index set $I \in \mathcal{I}$, the $\{\phi_j : j \in I\}$ are indicators for sets $\{S_j : j \in I\}$ that form a partition of the covariate space (Molinaro et al., 2004; Molinaro and van der Laan, 2003). The basis functions ϕ_j may themselves be defined as *tensor products of univariate basis functions*, $e_0, e_1, e_2, \ldots$, such as polynomial powers (e.g., $e_0(x) = 1$, $e_1(x) = x$, $e_2(x) = x^2$,...), spline basis functions, or wavelets basis functions. Given a d-vector $\vec{p} = (p_1, \ldots, p_d) \in I\!N^d$, let $\phi_{\vec{p}}(W) = e_{p_1}(W_1) \times \ldots \times e_{p_d}(W_d)$ denote the tensor product of univariate basis functions identified by $\vec{p}$. For instance, for polynomial basis functions, the multivariate basis functions are $\phi_{\vec{p}}(W) = W_1^{p_1} \ldots W_d^{p_d}$.

The collection of basis functions $\{\phi_j : j \in I\!N\}$ (or $\{\phi_{\vec{p}} : \vec{p} \in I\!N^d\}$, in the tensor product representation above), provides a basis for the complete parameter space $\mathbf{\Psi}$, which can be represented by

$$\mathbf{\Psi} = \{\psi_{I,\beta} = \sum_{j \in I} \beta_j \phi_j : \beta \in B_I, \;\; I \in \mathcal{I}\}. \tag{8}$$

One can then define a *sieve*, $\{\mathbf{\Psi}_k\}$, of subspaces $\mathbf{\Psi}_k \subseteq \mathbf{\Psi}$, of increasing dimension approximating the complete parameter space $\mathbf{\Psi}$. For example,

$$\mathbf{\Psi}_k \equiv \left\{ \psi_{I,\beta} = \sum_{j \in I} \beta_j \phi_j : \beta \in B_I, \, I \in \mathcal{I}, \, |I| \leq k \right\}. \tag{9}$$

Our approach is to seek, for each index set size k, the estimator that minimizes the empirical risk over the subspace $\mathbf{\Psi}_k$. We tackle this risk optimization problem in two steps: optimization over regression coefficients $\beta \in B_I$ for a given index set I (e.g., least squares estimation as in Section 2.3.2, for the squared error loss) and optimization over index sets I (e.g., D/S/A algorithm in Section 2.3.3, below). One can further reduce the number of candidates $\psi_{I,\beta}$ in $\mathbf{\Psi}_k$ by imposing constraints on the basis functions ϕ_j or regression coefficients β_j. For instance, in polynomial regression, one can enforce constraints on the degree of the polynomial bases, such as: $\sum_{j=1}^{d} \mathrm{I}(p_j \neq 0) \leq k'$ or $\sum_{j=1}^{d} p_j \leq k'$. The particular restrictions can be chosen by cross-validation.

2.3.2 Estimation of regression coefficients for a given subset of basis functions

Given index sets $I \in \mathcal{I}$, define I-specific subspaces

$$\mathbf{\Psi}_I \equiv \{\psi_{I,\beta} : \beta \in B_I\}. \tag{10}$$

For each subspace $\mathbf{\Psi}_I$, the regression coefficients β are estimated by minimizing the empirical risk, that is,

$$\begin{aligned} \hat{\beta}_I = \beta_I(P_n) &\equiv \underset{\beta \in B_I}{\operatorname{argmin}} \int L(o, \psi_{I,\beta} \mid \hat{\upsilon}_n) dP_n(o) \\ &= \underset{\beta \in B_I}{\operatorname{argmin}} \sum_{i=1}^{n} L(O_i, \psi_{I,\beta} \mid \hat{\upsilon}_n), \end{aligned} \tag{11}$$

where $\hat{\upsilon}_n = \hat{\Upsilon}(P_n)$ is an estimator of the nuisance parameter $\upsilon_0 = \Upsilon(P_0)$ for the observed data loss function (Section 2.2.2). Denote the resulting I-specific estimators by $\hat{\psi}_I = \hat{\Psi}_I(P_n) \equiv \psi_{I,\beta_I(P_n)}$, $I \in \mathcal{I}$.

In the special case of the squared error loss function, with full data, $\hat{\psi}_I$ is simply the least squares linear regression estimator corresponding with the variables identified by the index set I. That is,

$$\hat{\beta}_I = \underset{\beta \in B_I}{\operatorname{argmin}} \sum_{i=1}^{n} (Z_i - \psi_{I,\beta}(W_i))^2 = \underset{\beta \in B_I}{\operatorname{argmin}} \sum_{i=1}^{n} (Z_i - \sum_{j \in I} \beta_j \phi_j(W_i))^2.$$

2.3.3 D/S/A algorithm for minimizing risk over subsets of basis functions

We propose a new algorithm for minimizing risk over subsets of basis functions, i.e., over index sets I, according to three types of moves for the elements of I: deletions, substitutions, and additions. We refer to this algorithm as the *Deletion/Substitution/Addition algorithm*, or *D/S/A algorithm*. The main features of this novel approach are summarized below for tensor product basis functions ψ_j (e.g., tensor products of univariate polynomial basis functions in polynomial regression). The reader is referred to Sinisi and van der Laan (2003) for a more complete discussion and simulation studies assessing the performance of this general search procedure. Adaptations to

histogram regression, with indicator basis functions, are discussed in Molinaro and van der Laan (2003).

The D/S/A algorithm for minimizing risk over index sets I is defined in terms of three functions, $DEL(I)$, $SUB(I)$, and $ADD(I)$, which map an index set $I \in \mathcal{I}$ of size k into *sets of index sets* of size $k-1$, k, and $k+1$, respectively (Box 1). That is, for deletion moves,

$$DEL : I \in \mathcal{I} \rightarrow DEL(I) \subseteq \mathcal{I},$$
$$\text{with } |I^-| = |I| - 1 \text{ for } I^- \in DEL(I).$$

Box 1. Deletion/Substitution/Addition moves.

Consider index sets $I \subseteq \mathbb{N}^d$ and let $\mathcal{I}$ denote a collection of subsets of $\mathbb{N}^d$.

Deletion moves. Given an index set $I \in \mathcal{I}$ of size $k = |I|$, define a set $DEL(I) \subseteq \mathcal{I}$ of index sets of size $k-1$, by deleting individual elements of I. This results in k possible deletion moves, i.e., $|DEL(I)| = k$.

Substitution moves. Given an index set $I \in \mathcal{I}$ of size $k = |I|$, define a set $SUB(I) \subseteq \mathcal{I}$ of index sets of size k, 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 $\vec{p}$. That is, for each $\vec{p} \in I$, consider moves $\vec{p} \pm \vec{u}_j$, where $\vec{u}_j$ denotes the unit d-vector with one in position j and zero elsewhere, $j = 1, \ldots, d$. This results in up to $k \times (2d)$ possible substitution moves, i.e., $|SUB(I)| = k \times (2d)$.

Addition moves. Given an index set $I \in \mathcal{I}$ of size $k = |I|$, define a set $ADD(I) \subseteq \mathcal{I}$ of index sets of size $k+1$, by adding to I an element of $SUB(I)$ or one of the d unit vectors $\vec{u}_j$, $j = 1, \ldots, d$. This results in up to $k \times (2d) + d$ possible addition moves, i.e., $|ADD(I)| = k \times (2d) + d$.

Note that, for subspaces $\mathbf{\Psi}_k$ defined in terms of restrictions other than the size k of the index sets I (e.g., constraints on the degree of polynomial basis functions), substitution moves $\vec{p} \pm \vec{u}_j$ may result in inadmissible candidate estimators. In this case, substitution moves should be replaced by *swap moves*, where, for example, a number of components of $\vec{p} \pm \vec{u}_j$ are set to zero

in order to satisfy the constraints specifying $\mathbf{\Psi}_k$ (Sinisi and van der Laan, 2003).

Next, we describe how the three basic moves of the D/S/A algorithm can be used to generate index sets $I_k(P_n)$, that seek to minimize the empirical risk function, $f_E(I)$, over all index sets I of size less than or equal to k, $k = 1, \ldots, m$ (Box 2). For an index set $I \in \mathcal{I}$, the *empirical risk* of the I-specific estimator $\hat{\psi}_I = \hat{\Psi}_I(P_n)$ (as defined in Section 2.3.2) is

$$\begin{aligned} I \rightarrow f_E(I) &\equiv \int L(o, \hat{\psi}_I \mid \hat{\upsilon}_n) dP_n(o) \\ &= \frac{1}{n} \sum_{i=1}^{n} L(O_i, \hat{\psi}_I \mid \hat{\upsilon}_n), \end{aligned} \tag{12}$$

where $\hat{\psi}_I = \hat{\Psi}_I(P_n)$ and $\hat{\upsilon}_n = \hat{\Upsilon}(P_n)$ are estimators based on the empirical distribution P_n for the *entire learning set*. In the special case of the squared error loss function, with full data, the empirical risk function is simply the mean squared error (cf. residual sum of squares) for $\hat{\psi}_I$

$$f_E(I) = \frac{1}{n} \sum_{i=1}^{n} (Z_i - \hat{\psi}_I(W_i))^2.$$

Denote the best (in terms of empirical risk) index set I of size less than or equal to k, $k = 1, \ldots, m$, by

$$I_k^\star(P_n) \equiv \underset{\{I : |I| \leq k,\ I \in \mathcal{I}\}}{\operatorname{argmin}} f_E(I).$$

The D/S/A algorithm in Box 2 returns for each k, an index set $I_k(P_n)$ that approximates $I_k^\star(P_n)$. Denote the resulting estimator by $\hat{\psi}_k = \hat{\Psi}_k(P_n) \equiv \hat{\psi}_{I_k(P_n)}$. Cross-validation can then be used to select the optimal index set size k, as detailed in Section 2.4,

$$k(P_n) \equiv \underset{k}{\operatorname{argmin}}\, E_{S_n} \int L(o, \hat{\Psi}_k(P^0_{n,S_n}) \mid \hat{\upsilon}_{n,S_n^0}) dP^1_{n,S_n}(o), \tag{13}$$

where $\hat{\Psi}_k(P^0_{n,S_n})$ and $\hat{\upsilon}_{n,S_n^0} = \hat{\Upsilon}(P^0_{n,S_n})$ are estimators based on the empirical distributions P^0_{n,S_n} for the *training sets only*. Denote the final estimator corresponding to the index set size $\hat{k} = k(P_n)$ by $\hat{\psi} = \hat{\Psi}(P_n) \equiv \hat{\psi}_{k(P_n)}$.

Box 2. Deletion/Substitution/Addition algorithm for optimizing the empirical risk function.

1. **Initialization.** Set $I_0 = \emptyset$ and $BEST(k) = \infty$, $k = 1, 2, \ldots$, where $BEST(k)$ represents the current lowest value of the objective function $f = f_E$ for index sets I of size k.

2. **Algorithm (*).** Let $k = |I_0|$. Find an optimal updated index set I^- of size $k-1$, among all allowed **deletion** moves: $I^- \equiv \text{argmin}_{I \in DEL(I_0)} f(I)$. If $f(I^-) < BEST(k-1)$, then set $I_0 = I^-$, $BEST(k-1) = f(I^-)$, and go back to (*).

   Otherwise, find an optimal updated index set $I^=$ of the same size k as I_0, among all allowed **substitution** moves: $I^= \equiv \text{argmin}_{I \in SUB(I_0)} f(I)$. If this update improves on I_0, that is, $f(I^=) < f(I_0)$, then set $I_0 = I^=$, $BEST(k) = f(I^=)$, and go back to (*).

   Otherwise, find an optimal updated index set I^+ of size $k+1$, among all allowed **addition** moves: $I^+ \equiv \text{argmin}_{I \in ADD(I_0)} f(I)$. If this update improves on I_0, that is, $f(I^+) < f(I_0)$, then set $I_0 = I^+$ and $BEST(k+1) = f(I^+)$.

3. **Stopping rule.** Run the algorithm until the current index set size $k = |I_0|$ is larger than a user-supplied m or until $f(I^+) - f(I_0) < \epsilon$ for a user-specified $\epsilon > 0$. Denote the last set I by $I_{\text{final}}(P_n)$.

Note the following. Firstly, the D/S/A algorithm is such that $BEST(k)$ is decreasing in k, since addition moves only occur when they result in a decrease in risk over the current index set size. Thus, the best subset of size k is also the best subset of size less than or equal to k. Secondly, each step of the D/S/A algorithm is linear in the dimension d of the covariate space and in the current size k of the index set. An interesting open question is the number of iterations for substitution moves. Finally, the D/S/A algorithm for generating candidate estimators is completely defined by the following choices: the loss function, the basis functions ϕ_j defining the parameterization $\psi_{\beta,I}$ of the parameter space, and the sets of deletion, substitution, and addition moves. Consequently, the D/S/A algorithm can be adapted straightforwardly to address a broad range of estimation problems, with dif-

ferent objective functions.

The above approach is very similar to commonly adopted methods in the model selection literature, whereby estimators identified by index sets of the same size are compared in terms of their empirical risk and cross-validation is only applied to select the size of the index sets (i.e., the size of the model). However, unlike previously proposed model selection approaches (e.g., forward/backward type algorithms), the new D/S/A algorithm performs an extensive search of the parameter space, truly aimed at minimizing the empirical risk function over all index sets of a given size.

Example 1. Prediction: D/S/A moves with polynomial basis functions. Suppose we wish to predict a continuous outcome Z based on a 4-dimensional vector of explanatory variables, $W = (W_1, W_2, W_3, W_4)$, $d = 4$. We illustrate the three main moves in the D/S/A algorithm, with basis functions defined as tensor products of univariate polynomial basis functions, $\phi_{\vec{p}}(W) = e_{p_1}(W_1) \times e_{p_2}(W_2) \times e_{p_3}(W_3) \times e_{p_4}(W_4) = W_1^{p_1} W_2^{p_2} W_3^{p_3} W_4^{p_4}$, $\vec{p} = (p_1, p_2, p_3, p_4) \in I\!N^4$, where $e_j(x) = x^j$, $j = 0, 1, 2, \ldots$. In this case, index sets I are subsets of $I\!N^4$.

Suppose the current index set in the D/S/A algorithm is $I_0 = \{\vec{p}_1, \vec{p}_2\}$, where $\vec{p}_1 = (1, 1, 1, 0)$, $\vec{p}_2 = (0, 1, 0, 5)$, and $k = |I_0| = 2$. This index set corresponds to basis functions $\phi_{\vec{p}_1}(W) = W_1 W_2 W_3$ and $\phi_{\vec{p}_2}(W) = W_2 W_4^5$. The deletions set, $DEL(I_0)$, contains two index sets of size $k = 1$

$$DEL(I_0) = \Big\{\{\vec{p}_1\}, \{\vec{p}_2\}\Big\} = \Big\{\{(1, 1, 1, 0)\}, \{(0, 1, 0, 5)\}\Big\}.$$

The substitutions set, $SUB(I_0)$, contains thirteen index sets, $I^{=} = \{\vec{p}_1^{\,*}, \vec{p}_2^{\,*}\}$, of size $k = 2$, where either $\vec{p}_1^{\,*} = \vec{p}_1$ or $\vec{p}_2^{\,*} = \vec{p}_2$ (here, the cardinality of $SUB(I_0)$ is less than $k \times (2d) = 16$, because moves that result in negative

powers are not allowed, e.g., $\vec{p}_2 - \vec{u}_1$).

$$
\begin{aligned}
SUB(I_0) = \Big\{ & \{\vec{p}_1 + \vec{u}_1 = (2,1,1,0),\ \vec{p}_2 = (0,1,0,5)\}, \\
& \{\vec{p}_1 + \vec{u}_2 = (1,2,1,0),\ \vec{p}_2 = (0,1,0,5)\}, \\
& \{\vec{p}_1 + \vec{u}_3 = (1,1,2,0),\ \vec{p}_2 = (0,1,0,5)\}, \\
& \{\vec{p}_1 + \vec{u}_4 = (1,1,1,1),\ \vec{p}_2 = (0,1,0,5)\}, \\
& \{\vec{p}_1 - \vec{u}_1 = (0,1,1,0),\ \vec{p}_2 = (0,1,0,5)\}, \\
& \{\vec{p}_1 - \vec{u}_2 = (1,0,1,0),\ \vec{p}_2 = (0,1,0,5)\}, \\
& \{\vec{p}_1 - \vec{u}_3 = (1,1,0,0),\ \vec{p}_2 = (0,1,0,5)\}, \\
& \{\vec{p}_1 = (1,1,1,0),\ \vec{p}_2 + \vec{u}_1 = (1,1,0,5)\}, \\
& \{\vec{p}_1 = (1,1,1,0),\ \vec{p}_2 + \vec{u}_2 = (0,2,0,5)\}, \\
& \{\vec{p}_1 = (1,1,1,0),\ \vec{p}_2 + \vec{u}_3 = (0,1,1,5)\}, \\
& \{\vec{p}_1 = (1,1,1,0),\ \vec{p}_2 + \vec{u}_4 = (0,1,0,6)\}, \\
& \{\vec{p}_1 = (1,1,1,0),\ \vec{p}_2 - \vec{u}_2 = (0,0,0,5)\}, \\
& \{\vec{p}_1 = (1,1,1,0),\ \vec{p}_2 - \vec{u}_4 = (0,1,0,4)\} \Big\}.
\end{aligned}
$$

The additions set, $ADD(I_0)$, contains 13+4 = 17 index sets, $I^+ = \{\vec{p}_1, \vec{p}_2, \vec{p}_3\}$, of size $k = 3$, where $\vec{p}_3$ is either one of the thirteen new 4-vectors introduced in the above substitutions set $SUB(I_0)$ or one of the four unit vectors $\vec{u}_j$, $j = 1, \ldots, 4$. For instance, one of the seventeen index sets $I^+ \in ADD(I_0)$ is $I^+ = \{\vec{p}_1 = (1,1,1,0), \vec{p}_2 = (0,1,0,5), \vec{p}_3 = \vec{p}_2 + \vec{u}_3 = (0,1,1,5)\}$. This set corresponds to the three basis functions $\phi_{\vec{p}_1}(W) = W_1W_2W_3$, $\phi_{\vec{p}_2}(W) = W_2W_4^5$, and $\phi_{\vec{p}_3}(W) = W_2W_3W_4^5$, and hence, to the following candidate predictor for the outcome Z: $\psi_{I^+,\beta}(W) = \beta_1 W_1W_2W_3 + \beta_2 W_2W_4^5 + \beta_3 W_2W_3W_4^5$.

2.4 Cross-validation for estimator selection (Step 3)

2.4.1 The estimator selection problem

Search procedures, such as the D/S/A algorithm for optimizing the empirical risk function $f_E(I)$ over same size index sets I, can be used to construct a sequence of candidate estimators, $\hat{\psi}_k = \hat{\Psi}_k(P_n) \in \mathbf{\Psi}$, $k \in \{1, \ldots, K_n\}$, for the parameter ψ_0. The next step is to select an optimal estimator among these candidates. Specifically, the *estimator selection problem* involves choosing a data-adaptive selector $\hat{k} = k(P_n)$, so that the *risk distance*, or *risk difference*,

$d_n(\hat{\psi}_{\hat{k}}, \psi_0)$, between the estimator $\hat{\psi}_{\hat{k}}$ and the parameter ψ_0, converges to zero at an asymptotically optimal rate. Here,

$$\begin{aligned} d_n(\hat{\psi}_k, \psi_0) &\equiv \int \Big\{ L(x, \hat{\psi}_k) - L(x, \psi_0) \Big\} \, dF_{X,0}(x) \qquad (14) \\ &\qquad \text{(full data loss function)} \\ &= \int \Big\{ L(o, \hat{\psi}_k \mid \upsilon_0) - L(o, \psi_0 \mid \upsilon_0) \Big\} \, dP_0(o) \\ &\qquad \text{(observed data loss function).} \end{aligned}$$

Ideally, one would like to choose $\hat{k}$ as the *optimal benchmark selector*, $\tilde{k}_n$, which minimizes the risk distance $d_n(\hat{\psi}_k, \psi_0)$, that is,

$$\tilde{k}_n \equiv \underset{k \in \{1,\ldots,K_n\}}{\operatorname{argmin}} \; d_n(\hat{\psi}_k, \psi_0) = \underset{k \in \{1,\ldots,K_n\}}{\operatorname{argmin}} \; \tilde{\theta}_n(k), \qquad (15)$$

where $\tilde{\theta}_n(k)$ is the *conditional risk* for the candidate estimator $\hat{\psi}_k = \hat{\Psi}_k(P_n)$

$$\tilde{\theta}_n(k) \equiv \int L(x, \hat{\psi}_k) dF_{X,0}(x) = \int L(o, \hat{\psi}_k \mid \upsilon_0) dP_0(o). \qquad (16)$$

However, the risk distance $d_n(\hat{\psi}_k, \psi_0)$, and hence the optimal benchmark selector $\tilde{k}_n$, depend on the *unknown* data generating distribution P_0. Thus, in practice, the selection problem involves estimating the conditional risk $\tilde{\theta}_n(k)$ for each candidate estimator $\hat{\psi}_k \in \mathbf{\Psi}$, $k \in \{1, \ldots, K_n\}$, and seeking k that minimizes this risk estimator. Cross-validation is a general approach for risk estimation and estimator selection.

Example 1. Prediction. For the squared error loss function, $L(X, \psi) = (Z - \psi(W))^2$, the risk difference simplifies to

$$d_n(\hat{\psi}_k, \psi_0) = \int \Big(\hat{\psi}_k(w) - \psi_0(w) \Big)^2 \, dF_{W,0}(w),$$

that is, a squared *bias* term for $\hat{\psi}_k$ as an estimator of ψ_0.

Example 2. Density estimation. For the negative log-likelihood loss function, the risk difference is the *Kullback-Leibler divergence*, or *relative entropy*, between densities $\hat{\psi}_k$ and ψ_0

$$d_n(\hat{\psi}_k, \psi_0) = - \int \log \left(\frac{\hat{\psi}_k(x)}{\psi_0(x)} \right) \psi_0(x) dx.$$

2.4.2 Cross-validation

The main idea in *cross-validation* (CV) is to divide the available learning set into two sets: a *training set* and a *validation set*. Observations in the training set are used to compute (or *train*) the estimator(s) and the validation set is used to assess the performance of (or *validate*) this estimator(s) in terms of a loss function.

To derive a general representation for cross-validation, we introduce a binary random n-vector, or *split vector*, $S_n \in \{0,1\}^n$, independent of the empirical distribution P_n. A realization of $S_n = (S_{n,1}, \ldots, S_{n,n})$ defines a particular split of the learning set of n observations into a training set and a validation set,

$$S_{n,i} \equiv \begin{cases} 0, & \text{if } i\text{th observation is in training set,} \\ 1, & \text{if } i\text{th observation is in validation set.} \end{cases}$$

The particular distribution of the split vector S_n defines the type of cross-validation procedure. This representation covers many types of CV procedures, including leave-one-out cross-validation (LOOCV), V-fold cross-validation, Monte-Carlo cross-validation, and bootstrap-based cross-validation (van der Laan and Dudoit, 2003). Let P^0_{n,S_n} and P^1_{n,S_n} denote the empirical distributions of the training and validation sets, respectively, and let $p = p_n = n_1/n$ be the proportion of observations in the validation set, where $n_1 = \sum_{i=1}^n \mathrm{I}(S_{n,i} = 1)$.

Given a candidate estimator $\hat{\psi}_k = \hat{\Psi}_k(P_n)$, the *cross-validation risk estimator* of the conditional risk $\tilde{\theta}_n(k)$, using the full data loss function, is

$$\begin{aligned} \hat{\theta}_{n(1-p)}(k) &\equiv E_{S_n} \int L(x, \underbrace{\hat{\Psi}_k(P^0_{n,S_n})}_{\text{Training}}) \underbrace{dP^1_{n,S_n}(x)}_{\text{Validation}} \\ &= E_{S_n} \frac{1}{n_1} \sum_{\{i:S_{n,i}=1\}} L(X_i, \hat{\Psi}_k(P^0_{n,S_n})). \end{aligned} \tag{17}$$

In censored data problems, the cross-validation risk estimator of $\tilde{\theta}_n(k)$, based

on the IPCW loss function, is given by

$$\begin{aligned}\hat{\theta}_{n(1-p)}(k) &\equiv E_{S_n}\int L(o,\underbrace{\hat{\Psi}_k(P^0_{n,S_n})\mid \hat{\upsilon}^0_{n,S_n}}_{\text{Training}})\underbrace{dP^1_{n,S_n}(o)}_{\text{Validation}} \\ &= E_{S_n}\frac{1}{n_1}\sum_{\{i:S_{n,i}=1\}} L(X_i,\hat{\Psi}_k(P^0_{n,S_n}))\frac{\Delta_i}{\bar{G}^0_{n,S_n}(T_i\mid W_i)}.\end{aligned} \tag{18}$$

Here, $\hat{\Psi}_k(P^0_{n,S_n})$ and $\hat{\upsilon}_{n,S^0_n}=\hat{\Upsilon}(P^0_{n,S_n})$ denote, respectively, estimators for the parameter of interest ψ_0 and the loss function nuisance parameter υ_0 ($\bar{G}_0$ for the IPCW loss function), *using only the training set*. The *cross-validation selector* $\hat{k}=k(P_n)$ is chosen so that, among all K_n candidate estimators, $\hat{\Psi}_{\hat{k}}$ has the best performance on the validation sets

$$\hat{k} \equiv \underset{k\in\{1,\ldots,K_n\}}{\operatorname{argmin}}\ \hat{\theta}_{n(1-p)}(k). \tag{19}$$

Example 1. Prediction. For the squared error loss function, $L(X,\psi)=(Z-\psi(W))^2$, the cross-validation selector is

$$\hat{k}=\underset{k\in\{1,\ldots,K_n\}}{\operatorname{argmin}}\ E_{S_n}\sum_{\{i:S_{n,i}=1\}}(Z_i-\hat{\Psi}_k(P^0_{n,S_n})(W_i))^2.$$

For regression problems with censored outcomes, the cross-validation selector based on the IPCW squared error loss function is given by

$$\hat{k}=\underset{k\in\{1,\ldots,K_n\}}{\operatorname{argmin}}\ E_{S_n}\sum_{\{i:S_{n,i}=1\}}(Z_i-\hat{\Psi}_k(P^0_{n,S_n})(W_i))^2\frac{\Delta_i}{\bar{G}^0_{n,S_n}(T_i\mid W_i)},$$

where $\bar{G}^0_{n,S_n}$ is an estimator of the censoring survivor function based only on the training set.

Example 2. Density estimation. For the negative log-likelihood loss function, $L(X,\psi)=-\log\psi(X)$, the cross-validation selector is given by

$$\hat{k}=\underset{k\in\{1,\ldots,K_n\}}{\operatorname{argmin}}\ -E_{S_n}\sum_{\{i:S_{n,i}=1\}}\log\hat{\Psi}_k(P^0_{n,S_n})(X_i).$$

For density estimation problems with censored outcomes, the cross-validation selector based on the IPCW loss function is given by

$$\hat{k}=\underset{k\in\{1,\ldots,K_n\}}{\operatorname{argmin}}\ -E_{S_n}\sum_{\{i:S_{n,i}=1\}}\log\hat{\Psi}_k(P^0_{n,S_n})(X_i)\frac{\Delta_i}{\bar{G}^0_{n,S_n}(T_i\mid W_i)},$$

where $\bar{G}^0_{n,S_n}$ is an estimator of the censoring survivor function based only on the training set.

2.4.3 Asymptotic optimality of cross-validation selection

A selector $\hat{k} = k(P_n)$ is said to be *asymptotically equivalent* with the optimal benchmark $\tilde{k}_n$ if the ratio of risk distances

$$\frac{d_n(\hat{\psi}_{\hat{k}}, \psi_0)}{d_n(\hat{\psi}_{\tilde{k}_n}, \psi_0)} \longrightarrow 1 \text{ in probability as } n \to \infty.$$

In particular, then $\hat{k}$ is *asymptotically optimal*.

van der Laan and Dudoit (2003) derive finite sample and asymptotic optimality results concerning the cross-validation selector for general data generating distributions, loss functions (possibly depending on a nuisance parameter, υ_0, as in the IPCW loss function), and estimators. The asymptotic optimality result states that the cross-validation selector, $\hat{k}$, performs asymptotically as well as the optimal benchmark selector, $\tilde{k}_n$, based on the unknown data generating distribution P_0, provided that, as $n \to \infty$: (i) the proportion of observations in the validation set $p_n \to 0$ and (ii) $\log(K_n)/np_n$ and $\int(\bar{G}_n - \bar{G}_0)^2(t \mid x)dF_{X,0}(x)$ both converge to zero faster than the rate at which the estimator $\hat{\psi}_{\tilde{k}_n}$ converges to the parameter ψ_0 in risk distance, i.e., faster than $d_n(\hat{\psi}_{\tilde{k}_n}, \psi_0) \to 0$ (see van der Laan and Dudoit (2003) for full statements and proofs of the results). These new theoretical results have the important practical implication that, even finite sample situations, one can use cross-validation to engage in an intensive search of a large parameter space.

2.5 Cross-validation for performance assessment (Step 3)

2.5.1 Honest cross-validation

It is important to note that risk estimators from cross-validation relate only to the aspects of estimation that were cross-validated. Hence, it is essential to perform cross-validation on the entire estimation, or training, process, including feature selection and other choices, such as the number of neighbors k in nearest neighbor classification (k-NN) and the kernel in support

vector machines (SVM). Otherwise, risk estimators can be severely biased downward, i.e., overly optimistic. Cross-validation has been widely-used in genomic data analysis, to compare estimators and for overall estimator performance assessment. For instance, in cancer microarray studies, estimates of classification error (i.e., risk for the indicator loss function) are often reported to support statements such as *"Clinical outcome X for cancer Y can be predicted accurately based on microarray gene expression measures."* However, it is common practice in these studies to screen genes and fine-tune predictor parameters (e.g., number of neighbors in k-NN, kernel in SVMs) using the entire learning set and then perform cross-validation only on the final portion of the predictor building process. The resulting error rates are therefore biased downward and give an overly optimistic view of the predictive power of microarray expression measures.

2.5.2 Nested cross-validation

Suppose that an estimator $\hat{\psi} = \hat{\Psi}(P_n)$, of the parameter ψ_0, has been selected as described above by cross-validation on a learning set of n observations. The overall performance of this 'final' estimator now needs to be assessed based on an independent *test set*. A *double*, or *nested*, *cross-validation* study can be performed, in which the learning set is obtained from a partition of a complete dataset of $n^\star$ observations into a learning set and a test set. Let $P_{n^\star}$ denote the empirical distribution of the complete dataset of $n^\star$ observations. The CV risk estimator of the overall performance of the selected estimator is given by

$$E_{S_n^\star} \int L(o, \hat{\Psi}(P^0_{n^\star,S_n^\star}) \mid \hat{\upsilon}^0_{n^\star,S_n^\star}) dP^1_{n^\star,S_n^\star}(o),$$

where $S_n^\star$ refers to binary split vectors for the entire dataset of $n^\star$ observations and $P^0_{n^\star,S_n^\star}$ corresponds to the empirical distribution P_n of a learning set of n observations. Note that the entire estimation procedure (i.e., all three steps in the road map) is now applied to each learning set, i.e., each $P^0_{n^\star,S_n^\star}$.

2.5.3 Risk confidence intervals

The risk estimators from cross-validation are *statistics*, i.e., they are functions of the empirical distribution P_n, and thus vary from sample to sample. It is therefore natural to study the sampling distribution of these statistics and derive confidence intervals for the risk they are estimating. Dudoit and

van der Laan (2003) prove that cross-validated risk estimators are consistent and asymptotically linear for the risk based on the true underlying distribution, and use these results to derive confidence intervals for the unknown risk. An approximate asymptotic $(1-\alpha)100\%$ *confidence interval* for the conditional risk $\tilde{\theta}_n$, defined in equation (2), is given by

$$\hat{\theta}_{n(1-p)} \pm z_{1-\alpha/2}\frac{\hat{\sigma}_n}{\sqrt{n}}, \tag{20}$$

where

$$\hat{\sigma}_n^2 \equiv \int (IC(x \mid P_n))^2 dP_n(x),$$

$$IC(x \mid P_n) \equiv L(x, \hat{\Psi}(P_n)) - \int L(x, \hat{\Psi}(P_n)) dP_n(x),$$

and $\Phi(z_{\alpha/2}) = 1-\alpha/2$ for the standard normal cumulative distribution function $\Phi(\cdot)$.

2.6 Loss-based variable importance

A common and practical question in prediction problems is to assess the *importance* of a variable (or set of variables) in terms of its predictive ability for an outcome of interest. For instance, in microarray experiments, one is interested in determining how important each gene (or set of genes) is for the prediction of a particular biological or clinical outcome. Measures of variable importance can then assist in the identification of a subset of marker genes for the outcome.

We propose to define variable importance in terms of a loss function (Teng et al., 2003). Consider a full data structure $X = (W, Z) \sim F_X$, where W is a d-dimensional vector of explanatory variables and Z is an outcome of interest. Let $J \subseteq \{1, \ldots, d\}$ and $\bar{J}$ refer to a subset of explanatory variables and its complement, respectively. Denote the reduced data structure, based on only the explanatory variables indexed by J, using $X(J) = (W(J), Z)$, where $W(J) = (W_j : j \in J)$, and the corresponding distribution by $F_{X(J)}$. Consider J-specific parameter spaces, $\mathbf{\Psi}(J) \equiv \{\Psi_J(F_X) = \Psi(F_{X(J)}) : F_X \in \mathcal{M}^F\}$, where given a subset $J \subseteq \{1, \ldots, d\}$ and distribution $F_X \in \mathcal{M}^F$, the parameters $\Psi_J(F_X) = \Psi(F_{X(J)})$ are well-defined analogs of the full parameter $\Psi(F_X)$. For instance, one can extend the $|J|$-dimensional distributions $F_{X(J)}$ to d-dimensional distributions, that are degenerate for $W(\bar{J})$,

i.e., assign mass one to some constant value for $W(\bar{J})$. When such extended distributions belong to the model $\mathcal{M}^F$, then $\mathbf{\Psi}(J) \subseteq \mathbf{\Psi}$. For example, in regression problems, the parameters of interest are conditional expected values of the outcome given sets of explanatory variables, i.e., $\Psi(F_{X(J)})(W) = E_{F_X}[Z \mid W(J)]$. Given a loss function $L(X, \psi)$, such as the squared error loss, $L(X, \psi) = (Z - \psi(W))^2$, one can then define the *J-specific parameters*, $\psi_{J0} = \Psi(F_{X(J),0})$, as the risk minimizers over $\mathbf{\Psi}(J)$, that is,

$$\int L(x, \psi_{J0}) dF_{X,0}(x) \equiv \min_{\psi \in \mathbf{\Psi}(J)} \int L(x, \psi) dF_{X,0}(x). \tag{21}$$

In particular, the full parameter $\psi_0 = \Psi(F_{X,0})$ corresponds to $J = \{1, \ldots, d\}$.

The *variable importance parameter*, $\gamma_{J0} = \Gamma_J(F_{X,0})$, for the set of variables indexed by J, can now be defined as the difference between the risk for the $\bar{J}$-specific parameter $\psi_{\bar{J}0}$, defined *without* the variables indexed by J, and the risk for the parameter ψ_0, based on *all* d explanatory variables. That is,

$$\gamma_{J0} \equiv \int \{L(x, \psi_{\bar{J}0}) - L(x, \psi_0)\} dF_{X,0}(x). \tag{22}$$

Note that in most applications, $\mathbf{\Psi}(\bar{J}) \subseteq \mathbf{\Psi}$, so that the importance parameters γ_{J0} are non-negative. Thus γ_{J0} measures the increase in risk (error) resulting from omitting explanatory variables $W(J) = (W_j : j \in J)$ from the estimation process. We stress that this general definition of variable importance applies to *sets of variables* and therefore allows examination of not only variable main effects (i.e., individual $j \in \{1, \ldots, d\}$), but also higher order interactions among variables. In particular, in the context of microarray experiments, this general definition can be used to assess the importance of gene clusters in terms of their predictive power for an outcome of interest. In addition, in high-dimensional problems, one could consider variable importance measures for orthogonal transformations of the explanatory variables (e.g., from singular value decomposition).

The variable importance parameters can be estimated using *variable importance statistics*, $\hat{\gamma}_J = \hat{\Gamma}_J(P_n)$, that are functions of the empirical distribution P_n, using either the empirical risk or the cross-validated risk. For the empirical risk

$$\hat{\gamma}_J \equiv \int \{L(x, \hat{\psi}_{\bar{J}}) - L(x, \hat{\psi})\} dP_n(x), \tag{23}$$

where the $\hat{\psi}_J = \hat{\Psi}_J(P_n)$ are estimators of the J-specific parameters based on the empirical distribution P_n, and $\hat{\psi}$ corresponds to $J = \{1, \ldots, d\}$.

Example 1. Prediction. The full parameter is $\psi_0(W) = E_0[Z \mid W]$ and the J-specific parameters are conditional expected values of the outcome Z given explanatory variables $W(J)$,

$$\begin{aligned} \psi_{J0}(W) &= E_0[Z \mid W(J)] \\ &= E_0[E_0[Z \mid W] \mid W(J)] \\ &= E_0[\psi_0(W) \mid W(J)]. \end{aligned}$$

The variable importance parameters, defined in terms of the squared error loss function, are

$$\begin{aligned} \gamma_{J0} &= \int \{(z - \psi_{\bar{J}0}(w))^2 - (z - \psi_0(w))^2\} dF_{X,0}(x) \\ &= E_0[(\psi_0(W) - \psi_{\bar{J}0}(W))^2] \\ &= E_0[Var_0[Z \mid W(\bar{J})]] - E_0[Var_0[Z \mid W]]. \end{aligned}$$

In the special case of linear conditional expectations, i.e., $\psi_0(W) = \sum_{j=1}^d \beta_{j0} W_j$, then

$$\psi_{\bar{J}0}(W) = \sum_{j \in \bar{J}} \beta_{j0} W_j + \sum_{j \in J} \beta_{j0} E_0[W_j \mid W(\bar{J})],$$

thus

$$\begin{aligned} \gamma_{J0} &= E_0\left[\left(\sum_{j \in J} \beta_{j0}(W_j - E_0[W_j \mid W(\bar{J})])\right)^2\right] \\ &= \sum_{j \in J} \beta_{j0}^2 E_0[Var_0[W_j \mid W(\bar{J})]] \\ &\quad + 2 \sum_{j,j' \in J,\ j<j'} \beta_{j0}\beta_{j'0} E_0[Cov_0[W_j, W_{j'} \mid W(\bar{J})]] \\ &= \sum_{j \in J} \beta_{j0}^2 Var_0[W_j] \qquad \text{(for independent } W_j\text{'s)} \\ &\quad + 2 \sum_{j,j' \in J,\ j<j'} \beta_{j0}\beta_{j'0} Cov_0[W_j, W_{j'}]. \end{aligned}$$

For single variables, i.e., $J = \{j\}$, the importance parameter is simply

$$\gamma_{\{j\}0} = \beta_{j0}^2 E_0[Var_0[W_j \mid W(\bar{J})]]$$

and reduces to $\gamma_{\{j\}0} = \beta_{j0}^2 Var_0[W_j]$ for independent explanatory variables.

3 Prediction of survival in microarray experiments

To evaluate our proposed loss-based estimation methodology and demonstrate its application to tree-structured estimation with censored data, we present the following results from a simulation study and analysis of breast cancer survival and CGH copy number data. Preliminary simulation results for a new D/S/A algorithm for histogram regression are also provided in this section. More detailed results and discussion can be found in Molinaro et al. (2004) and Molinaro and van der Laan (2003).

3.1 Simulation study: survival trees

The proposed survival tree approach based on the IPCW loss function was compared to that of LeBlanc and Crowley (1992), which is implemented as a default for censored data in the R `rpart` function (Therneau and Atkinson, 1997). The loss function for the survival trees of LeBlanc and Crowley (1992) is based on the observed data negative log-likelihood for a Cox proportional hazards model with the same baseline hazard for each node. Trees based on the IPCW loss function can be grown using the `rpart` function, by setting the `method` argument to "`anova`" and by providing the IPCW weights for individual observations through the `weights` argument. The censoring survivor function, $\bar{G}_0$, used in the IPCW loss function, is estimated separately for each training set. In what follows, Method 1 and Method 2 refer, respectively, to the survival trees of LeBlanc and Crowley (1992) and to trees grown using the proposed IPCW loss function. The two approaches differ in the choice of loss function for splitting and pruning and thus lead to two different partitions of the covariate space, i.e., to different assignments of observations to terminal nodes. Given such a final partition, we then consider two survival estimation methods for the terminal nodes: the IPCW mean survival time and the Kaplan-Meier (KM) median survival time. These two

types of estimators correspond to full data parameters defined in terms of the squared and absolute error loss functions, respectively. The two different loss functions and the two different within-node estimation methods thus produce *four* different predictors of survival (namely, Method 1 with IPCW mean, Method 1 with KM median, Method 2 with IPCW mean, Method 2 with KM median), which were compared by simulation as described below.

The following model was considered for the full data structure: $Z \equiv \log T = W^2 + \epsilon$, where W and ϵ are independent random variables with $W \sim U(0,1)$, $\epsilon \sim N(0, \sigma^2)$, and $\sigma = 0.25$. Thus, $E_0[Z|W] = \text{Median}_0[Z|W] = W^2$ and the conditional survivor function is given by $S_0(z \mid W) = Pr_0(Z \geq z \mid W) = 1 - \Phi((z - W^2)/\sigma)$, where $\Phi(\cdot)$ denotes the standard normal cumulative distribution function. Censoring times C were simulated using mixtures of three uniform distributions. The censoring survivor function, $\bar{G}_0$, used in the IPCW loss function, was estimated separately for each training set, by fitting a Cox proportional hazards model to the survival time T and covariate W.

One hundred simulated learning sets were generated from an observed data distribution with 20% censoring, for sample sizes $n = 250$, 600, 1250, and 6000. Risk estimates, based on test sets of size $N = 5000$ generated from the full data distribution, were computed for each of the four predictors, using the L_2 loss function for the IPCW within-node mean estimation method and the L_1 loss function for the KM median estimation method. Within each sample size, the four test set risk estimates were averaged over the $B = 100$ repetitions. Method 1 and Method 2 were compared by forming the ratio of Method 2's average risk to that of Method 1, separately for each of the two within-node estimation methods.

Ratios of average test set risk are displayed in Table 1 for both the KM median and IPCW mean estimation methods; ratios less than one correspond to improved accuracy for Method 2, i.e., for trees based on the new IPCW loss function. The results illustrate the impact on accuracy of the choice of loss function used for node splitting and tree pruning. As expected, when the parameter of interest is the conditional mean survival, the risk is smaller for partitions generated by Method 2 ("IPCW Mean" column). The IPCW loss function also corresponds to lower risk when interest is in estimating the median survival. The difference in risk decreases with increasing sample size.

Table 1: *Simulation study: survival trees.* Comparison of survival trees grown with Method 1 (`rpart`'s default) and Method 2 (proposed IPCW loss function). Ratios of average risk for Method 2 to Method 1 are displayed for the KM median and IPCW mean within-node estimation methods for four sample sizes, n. Individual entries of the table are ratios of average test set risk $(1/B)\sum_{b=1}^{B}\int L(x,\hat{\Psi}(P_n^b))dP_N^b(x)$, where $\hat{\Psi}$ refers to one of the four survival predictors, P_n^b and P_N^b denote, respectively, the learning set and test set empirical distributions in the bth simulation, $N = 5000$, $B = 100$. For the KM median within-node estimation method (column 2), L is the absolute error loss, and for the IPCW mean within-node estimation method (column 3), L is the squared error loss.

Sample size, n	Ratios of average risk	
	KM Median	IPCW Mean
250	0.9422	0.8838
600	0.9524	0.9062
1250	0.9629	0.9244
6000	0.9767	0.9533

3.2 Breast cancer survival and CGH copy number data analysis

Our censored regression tree method was also applied to a dataset from a *comparative genomic hybridization* (CGH) study of breast cancer patients. Data were collected on 152 patients, all with initial occurrences of breast cancer; 52 subsequently recurred. Time to event (in years) was defined as time to recurrence. Patients with no recurrence at the time of death or of final follow-up are censored. Explanatory variables include epidemiological variables (e.g., age at diagnosis, race), histopathological variables (e.g., tumor stage, grade), and DNA copy number measures from a CGH microarray with 2254 bacterial artificial chromosomes (BACs).

The 152 observations were split at random into a learning set and a test set of 128 and 24 (i.e., five sixths and one sixth) observations, respectively, while retaining the appropriate level of censoring. Trees were grown using the learning set and their overall performance assessed on the test set. Five-fold cross-validation of the learning set was used to select the 'best' tree (again, retaining the appropriate level of censoring). The censoring survivor function, $\bar{G}_0$, used in the IPCW loss function, was estimated separately for each of the five training sets in the cross-validation, by fitting a Cox proportional hazards model to the epidemiological and histopathological variables. The full learning set tree is shown in Figure 1, with filled circles for the two-split subtree. Each terminal node is described by the IPCW mean log survival time (in years) and the number of observations. The legend in the bottom left corner indicates the chromosomal location of each BAC. The first two splits are based on BACs that fall in chromosomal regions known to contain genes related to breast cancer (personal communication with Joe Gray and Fred Waldman).

This preliminary analysis illustrates limitations of single trees based on microarray measures: they typically involve a very small number of splits and therefore only provided limited biological insight. Improved prediction accuracy and more information on chromosomal regions related to breast cancer survival may be obtained from aggregation methods such as bagging and boosting and the use of loss-based variable importance measures as proposed in Section 2.6. In addition, we are exploring more aggressive procedures, based on the D/S/A algorithm, that include "OR" statements in addition

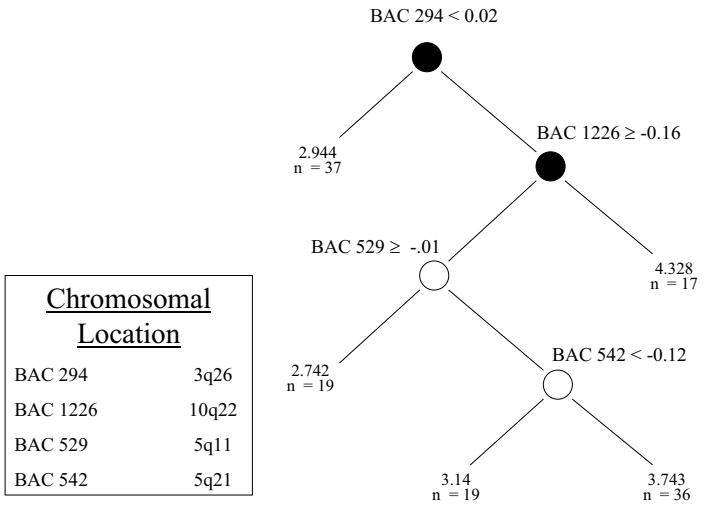

Figure 1: *Breast cancer survival and CGH copy number data analysis.* Survival tree built from the learning set of 128 patients, using the IPCW squared error loss function. Each terminal node is described by the IPCW mean log survival time (in years) and the number of observations.

to the "AND" statements of tree estimators (Molinaro and van der Laan, 2003).

3.3 Simulation study: D/S/A algorithm for histogram regression

We report the following preliminary results from a simulation study comparing a new D/S/A algorithm for histogram regression (i.e., as in Section 2.3, using indicator basis functions) to standard regression trees. We consider the following (full data) model: $Z \equiv W^2 + \epsilon$, where W and ϵ are independent random variables with $W \sim N(0,1)$, $\epsilon \sim N(0,\sigma^2)$, and $\sigma = 0.25$. The parameter of interest is $\psi_0(W) = E_0[Z|W] = W^2$ and the loss function is the

squared error loss. One hundred simulated learning sets were generated from the above distribution, for sample sizes $n = 250$, 500, and 1000. Regression trees were grown using the R `rpart` function, selecting tree size using five-fold cross-validation, with the 1-SE rule and without the 1-SE rule (Breiman et al., 1984; Therneau and Atkinson, 1997). The D/S/A algorithm of Molinaro and van der Laan (2003), with indicator basis functions, was applied to generate a sequence of candidate estimators (i.e., partitions of the real line) and five-fold cross-validation was used to select the optimal number of sets in the partition.

Risk estimates, based on test sets of size $N = 1000$ were computed for each of the three predictors, using the squared error loss function. Summary statistics over the $B = 100$ simulations are displayed in Table 2 for the test set risks and the partition sizes (i.e., number of indicator basis functions for final predictor).

These preliminary results demonstrate that significant gains in accuracy can be achieved, even in univariate situations, by selecting partitions using the D/S/A algorithm rather than standard tree methods. We anticipate even greater gains in accuracy from the D/S/A algorithm in multivariate situations. In addition, the results raise questions regarding the benefits of the widely-used 1-SE rule. In our simple example, where the parameter $\psi_0(W) = E_0[Z \mid W] = W^2$ is a quadratic function of the explanatory variable W, the D/S/A algorithm is able to exploit the symmetry of the parameter ψ_0, while the more rigid tree algorithms cannot recognize this symmetry. Trees built without the 1-SE rule (i.e., `tree0SE` for which tree size is obtained by minimizing the cross-validated risk) result in roughly twice as many sets in the final partition compared to the D/S/A algorithm.

4 Supervised detection of regulatory motifs in DNA sequences

4.1 COMODE

Keleş et al. (2003b) recently developed a likelihood-based method, called COMODE (**Co**nstrained **Mo**tif **De**tection), for the *supervised* detection of transcription factor *binding sites*, i.e., *regulatory motifs*. This new approach was

Table 2: *Simulation study: D/S/A algorithm for histogram regression.* Test set risk (mean squared error) for histogram regression with D/S/A algorithm (`DSA`), regression trees with 1-SE rule (`tree1SE`), and regression trees without 1-SE rule (`tree0SE`). The following summary statistics are reported for each prediction method and each sample size n, over $B = 100$ simulations: risk average (Risk Avg.), risk standard deviation (Risk SD), average partition size (Avg. Size), and average risk over average `DSA` risk (Ratio).

n	Method	Risk Avg.	Risk SD	Avg. Size	Ratio
250	`DSA`	0.26125	0.09384	7.44	1
	`tree1SE`	0.45305	0.14195	5.69	.577
	`tree0SE`	0.35172	0.09927	14.45	.743
500	`DSA`	0.18935	0.07318	9.95	1
	`tree1SE`	0.27216	0.08574	9.55	.696
	`tree0SE`	0.22187	0.07544	21.26	.853
1000	`DSA`	0.14080	0.04016	12.06	1
	`tree1SE`	0.18489	0.05206	13.02	.762
	`tree0SE`	0.15403	0.04916	28.44	.914

motivated by recent articles in the biological literature that suggest a direct relationship between the structural footprint of a protein on DNA and the information content profile of its position specific weight matrix and further indicate that transcription factors with similar structures bind to sites with similar information content profiles (Mirny and Gelfand, 2002). `COMODE` supervises the search for transcription factor binding sites using information derived from structural characteristics of protein-DNA interactions. Specific structural constraints on the motifs are enforced as constraints on the information content profile and/or individual entries of their position specific weight matrix.

Recall that the distribution of the four DNA bases in a motif can be represented by a *position specific weight matrix* (PWM), with rows corresponding to nucleotides $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ and columns to positions in the motif. Let $\overrightarrow{P}_w = (p_{w1}, p_{w2}, p_{w3}, p_{w4})$ denote the distribution of bases at position w of the motif, i.e., column w of the PWM, where the nucleotides $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ are recoded as $\{1, 2, 3, 4\}$. The *information content* (IC) of the PWM at position w is then given by

$$IC(w) = 2 + \sum_{j=1}^{4} p_{wj} \log_2 p_{wj} = 2 - \mathrm{Entropy}(w) \in [0, 2]. \tag{24}$$

The information content profile of a PWM is a measure of a binding site's tolerance for substitution: high IC, low tolerance. The IC achieves its maximum when $p_{wj} = 1$ for some base j and its minimum when all four bases are equally likely, i.e., $p_{wj} = 1/4$ for all j, $j = 1, 2, 3, 4$.

In `COMODE`, the widely used two component multinomial mixture model (Bailey and Elkan, 1994; Lawrence and Reilly, 1990) is extended to a *constrained multinomial mixture model*, by imposing constraints on the information content profile or on specific parameters of the motif PWM. The flexible framework of `COMODE` allows a wide variety of constraints. Examples of simple constraints include palindromicity, gap inclusion, control of the number of nucleotide repeats. Complex constraints include parametric modeling of the whole information content profile. Estimation of motif start site, entries of the PWM , and other model parameters is performed by *constrained maximum likelihood estimation*.

As in unconstrained motif detection methods, many model selection issues arise also with COMODE, regarding, for example, the motif width, the number of motifs per sequence, and the distribution of bases in the background sequence. Additional model selection issues, specific to the supervised framework, concern the type of constraints to be applied to the information content profile or to specific parameters of the motif PWM. When there is limited or no information on the structural properties of the protein-DNA interaction, our estimation road map for motif detection includes applying various types of constraints to the motif PWM and selecting the model that provides the best fit by cross-validation. Since COMODE relies on maximum likelihood estimation, it is natural to perform model selection using *likelihood-based cross-validation*, i.e., using the negative log-likelihood loss function (van der Laan et al., 2003a). We present below some results from the simulation studies and the data analysis performed by Keleş et al. (2003b).

4.2 Simulation study

Simulation studies were implemented to assess the performance of likelihood-based cross-validation for selecting among various binding site models, that correspond to different information content profiles for the PWM (Keleş et al., 2003b) or to different motif widths (van der Laan et al., 2003b).

4.2.1 Likelihood-based cross-validation for motif information content profile selection

$B = 100$ datasets, each comprising $n = 30$ sequences of length $L = 100$, were generated using an i.i.d. background model with multinomial base probabilities $Pr(\mathrm{A}) = 0.3$, $Pr(\mathrm{C}) = 0.2$, $Pr(\mathrm{G}) = 0.2$, $Pr(\mathrm{T}) = 0.3$. An instance of a weak motif of width $W = 13$ was inserted in a varying percentage ($F = 100\%, 75\%, 50\%, 25\%$) of the sequences. The information content profile of the motif PWM satisfies the following piecewise linear constraint

$$IC(w; \theta_1^*, \theta_2^*) = \theta_1^* + |w - w^*| \tan \theta_2^*, \qquad w = 1, \ldots, W,$$

where the motif width is $W = 13$, the parameter w^* is set to the motif center, $w^* = 7$, and (θ_1^*, θ_2^*) are unknown parameters to be estimated. The sequence logo of the motif PWM is given in Figure 2. As apparent from this logo (y-axis represents the total information content, with range $[0, 2]$), the values chosen for θ_1^* and θ_2^* give low overall information content, that is, correspond

to a binding site with weak signal.

Four different types of constraints for the motif information content profile were supplied to COMODE.

- **Profile model 0:** Unconstrained IC profile, as in the unconstrained multinomial mixture model of Bailey and Elkan (1994) and Lawrence and Reilly (1990).
- **Profile model I:** Piecewise linear ($\vee$-shaped) IC profile satisfying

$$IC(w; \theta_1, \theta_2) = \theta_1 + |w - w^*| \tan \theta_2,$$

  where θ_1 and θ_2 are additional parameters that need to be estimated.
- **Profile model II:** Ordered IC profile, such that the middle positions of the motif have lowest information content

$$IC(1) \geq IC(2) \geq \cdots \geq IC(w^*)$$
$$IC(w^*) \leq IC(w^* + 1) \leq \cdots \leq IC(W).$$

  Note that this type of profile does not require estimation of additional parameters.
- **Profile model III:** Piecewise linear ($\wedge$-shaped) IC profile satisfying

$$IC(w; \theta_1, \theta_2) = \theta_1 - |w - w^*| \tan \theta_2,$$

  where θ_1 and θ_2 are additional parameters that need to be estimated.

Note that Profile II roughly matches the true IC profile (Profile I), while Profile III is the mirror image of Profile I and is thus misspecified.

COMODE was used to perform constrained maximum likelihood estimation of the motif PWM and start sites (as well as additional parameters θ_1 and θ_2, for profiles I and III), under the above four models for the information content profile of the PWM. In this simulation study, the correct motif width W and center w^* were provided to COMODE, however, in practice these can also be selected with likelihood-based cross-validation. A *sensitivity* measure

was computed as follows for each profile model in each of the B simulated datasets

$$\widehat{sens}_b^k = \frac{|K_b \cap \hat{K}_b^k|}{|K_b|},$$

where $K_b = \{$set of *true* motif sites in simulated dataset $b\}$ and $\hat{K}_b^k = \{$set of predicted motif sites $k = 0$, I, II, III, $b = 1, \ldots, B$. Figure 3 displays boxplots of these sensitivity measures, for four different values of F, the percentage of motif occurrences per sequence. At $F = 100\%$, `COMODE` with either Profile I or II is performing dramatically better than an unconstrained motif search (Profile 0), which indicates that even though the motif signal is weak, incorporating knowledge about the information content profile (i.e., supervising the search) helps to discriminate it from the background. `COMODE` with Profiles I and II remains superior to `COMODE` with Profile 0 as the percentage F of motif occurrences per sequence decreases. Moreover, `COMODE` has similar performance with either Profile I or II, suggesting robustness of this motif detection approach to different profiles in the same `high-low-high` profile class. All four models, except that corresponding to Profile III, resulted in high *specificity* (between 0.92 and 0.95), i.e., did not predict sites on sequences that did not have a motif occurrence. As expected, `COMODE` with Profile III performs the worst since it searches for a motif with IC profile that is a mirror image of the true IC profile.

Two-fold likelihood-based cross-validation was also applied to select among the four binding site models corresponding to the four types of IC profiles. The numbers of times each profile model was selected out of $B = 100$ simulations, with $F = 100\%$, are as follows: Profile 0: 0, Profile I: 61, Profile II: 39, Profile III: 0. We first note that cross-validation successfully discards the misspecified Profile model III. Profile 0 is unconstrained and has the flexibility to match the true profile. However, since the signal for the binding site is weak (i.e., low overall information content), the unconstrained model is inferior when compared to models with Profiles I and II that match the true information content profile.

4.2.2 Likelihood-based cross-validation for motif width selection

$B = 200$ datasets, each comprising $n = 20,100$ sequences of length $L = 600$, were generated using an i.i.d. background model. A motif of width $W = 10$ was inserted in each of the sequences. Motif start sites and PWM were

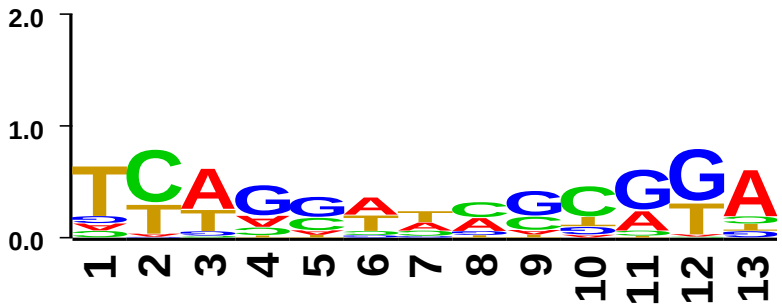

Figure 2: *Weak motif.* Sequence logo for a PWM with low overall information content.

estimated using `COMODE` with no constraints on the PWM, for motif widths W ranging from 6 to 15 base-pairs. Two-fold ($p = 0.5$) and five-fold ($p = 0.2$) likelihood-based cross-validation procedures were used to select motif width. The results are summarized in Table 3. We note that likelihood-based cross-validation was generally successful at identifying the correct motif width of 10, with, as expected, a better performance for the larger sample size. Over-fitting was not an issue, in the sense that cross-validation did not tend to select wider motifs than the truth. In addition, two-fold cross-validation performed well compared the more computer intensive five-fold cross-validation.

4.3 *S. cerevisiae* sequence data analysis

As described in detail in Keleş et al. (2003b), `COMODE` was applied to genome-wide binding sequence data from the yeast *Saccharomyces cerevisiae*. Likelihood-based cross-validation was used for model selection purposes. Here, we only report results for the transcription factor `BAS1`. A total of 19 upstream regions were identified as bound by `BAS1` by Lee et al. (2002). The binding site of `BAS1` is expected to be 5–6 base-pairs-wide, with a main conserved region. This structural information was translated into a constraint on the information content profile of the motif PWM, by forcing the IC at each position to be greater than a threshold. The threshold and motif width were determined by two-fold likelihood-based cross-validation. Specifically, the following constraints were enforced on the `BAS1` motif PWM: $IC(w) \geq \theta$, for $w \in \{1, \cdots, W\}$, with $W \in \{5, \ldots, 18\}$ and $\theta \in \{0.6, 1.2, 1.8\}$. This corre-

Table 3: *COMODE. Likelihood-based cross-validation for motif width selection.* Number of simulations (out of $B = 200$) each motif width W was selected, for sample sizes $n = 20, 100$ and using two- and five-fold CV. The true motif width is 10.

		$n = 20$		$n = 100$	
		2-fold	5-fold	2-fold	5-fold
	6	0	20	0	22
	7	24	42	0	10
	8	40	10	15	14
	9	11	17	3	3
W	**10**	**121**	**98**	**147**	**142**
	11	0	10	35	9
	12	0	3	0	0
	13	0	0	0	0
	14	0	0	0	0
	15	1	0	0	0

sponds to 14×3 models to be compared by likelihood-based cross-validation. The sequence logo of the PWM obtained by `COMODE` is displayed in Figure 4. This sequence logo matches that for the true consensus site reported for `BAS1` (Daignan-Fornier and Fink, 1992) and is only one base wider than the truth. Results for transcription factors `ARO80` and `SWI5` are given in Keleş et al. (2003b)

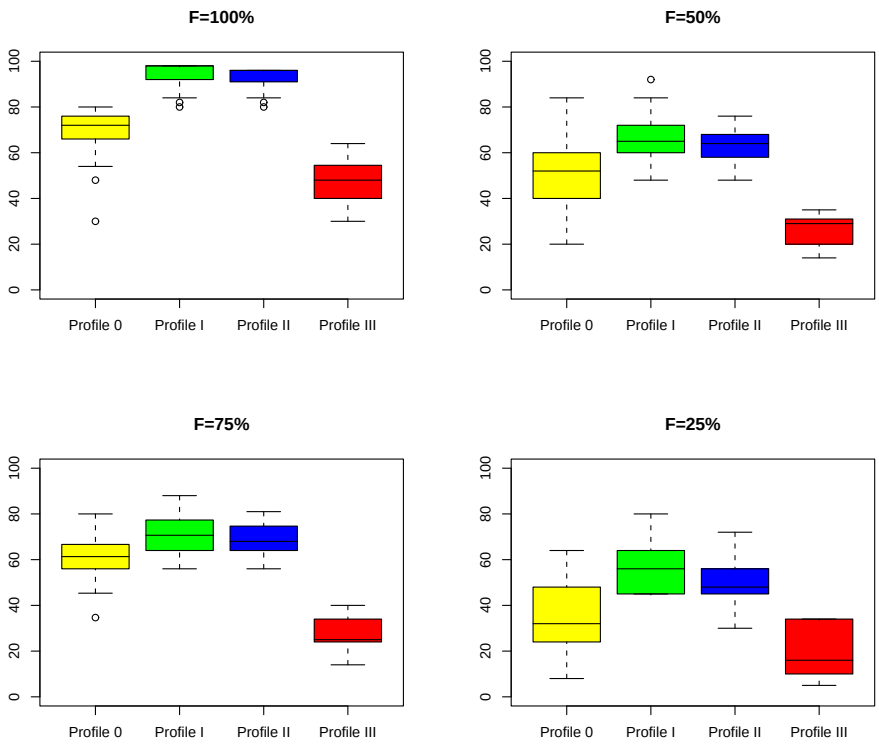

Figure 3: ***COMODE****. Likelihood-based cross-validation for motif information content profile selection.* Boxplots of sensitivity measures for four different PWM information content profile models.

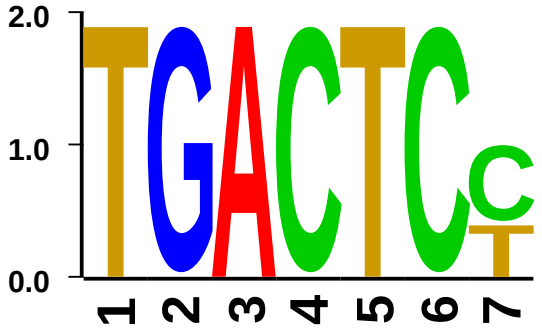

Figure 4: *COMODE. S. cerevisiae sequence data analysis.* Sequence logo for the `BAS1` PWM. The consensus sequence for `BAS1` reported in Daignan-Fornier and Fink (1992) is `TGACTC`. The sequence logo was obtained using software available at `http://genio.informatik.uni-stuttgart.de/GENIO/logo/logo.cgi`.

Table 4: *Loss functions.* Examples of full data loss functions, $L(X, \psi)$, for different estimation problems. Censored data loss functions, $L(O, \psi \mid \upsilon_0)$, can be obtained by applying the IPCW mapping to the full data loss functions $L(X, \psi)$: $L(O, \psi \mid \upsilon_0) \equiv L(X, \psi)\frac{\Delta}{\bar{G}_0(T|W)}$. See van der Laan and Robins (2002) and Molinaro et al. (2004) for other available mappings.

Full data	**Parameter**	**Full data loss function**
X	ψ_0	$L(X, \psi)$
Univariate prediction, continuous outcome, $Z \in I\!R$		
$X = (W, Z)$	Conditional mean:	Squared error (L2):
	$\psi_0(W) = E_0[Z\|W]$	$L(X, \psi) = (Z - \psi(W))^2$
	Conditional median:	Absolute error (L1):
	$\psi_0(W) = \text{Median}_0[Z\|W]$	$L(X, \psi) = \|Z - \psi(W)\|$
Univariate prediction, polychotomous outcome, $Z \in \{1, \ldots, K\}$		
$X = (W, Z)$	Class with max posterior probability:	Indicator (risk = classification error rate):
	$\psi_0(W) = \text{argmax}_z\ Pr_0(z \mid W)$	$L(X, \psi) = \text{I}(Z \neq \psi(W))$
	Posterior class probabilities:	Negative log-likelihood (risk = entropy):
	$\psi_0(X) = Pr_0(Z \mid W)$	$L(X, \psi) = -\log \psi(X)$
	Class with max posterior probability:	Gini:
	$\psi_0(X) = \text{I}\big(Z = \text{argmax}_z Pr_0(z \mid W)\big)$	$L(X, \psi) = 1 - \psi(X)$
Multivariate prediction, continuous outcome, $Z = (Z_l : l = 0, \ldots, m-1) \in I\!R^m$		
$X = (W, Z)$	Conditional mean vector:	Quadratic (L2):
	$\psi_0(W) = (E_0[Z_l \mid W] : l = 0, \ldots, m-1)$	$L(X, \psi) = (Z - \psi(W))^\top \Omega(W)(Z - \psi(W))$
Density estimation		
X	Density:	Negative log-likelihood:
	$\psi_0(X) = \frac{d}{dX} F_{X,0}(X)$	$L(X, \psi) = -\log \psi(X)$
Hazard function estimation		
$X = (W, T)$	Hazard function for T given W:	Negative log-likelihood loss for density $g(T\|W)$:
	$\psi_0(T, W) = \lambda_0(T\|W) = -\frac{d}{dT} \log(\bar{G}_0(T\|W))$	$L(X, \psi) = -\log \psi(T, W) + \int_0^T \psi(u, W) du$

Survivor function: $\bar{G}(t|w) = 1 - G(t|w) = Pr(T > t|w)$; Density: $g(t|w) = \frac{d}{dt} G(t|w) = \lambda(t|w) \exp(-\int_0^t \lambda(u|w) du)$.

Acknowledgements

We would like to thank Joe Gray, Dan Moore, and Fred Waldman (Comprehensive Cancer Center, University of California, San Francisco) for graciously providing the CGH dataset, biological insight, and fruitful discussions. We are also grateful to Terry Therneau and Elizabeth Atkinson (Mayo Clinic) for their thorough explanation of the R `rpart` package. Finally, we would like to acknowledge Mike Eisen, Derek Chiang, and Alan Moses (University of California, Berkeley) for a stimulating collaboration on the identification of transcription factor binding sites.

References

T. L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pages 28–36, 1994.

L. Breiman, J. H. Friedman, R. Olshen, and C. J. Stone. *Classification and regression trees*. The Wadsworth statistics/probability series. Wadsworth International Group, 1984.

B. Daignan-Fornier and G. R. Fink. Coregulation of purine and histidine biosynthesis by the trans criptional activators BAS1 and BAS2. *Proceedings of the National Academy of Sciences*, 89:6746–6750, 1992.

S. Dudoit and M. J. van der Laan. Asymptotics of cross-validated risk estimation in model selection and performance assessmen. Technical Report 126, Division of Biostatistics, University of California, Berkeley, 2003. URL `www.bepress.com/ucbbiostat/paper126/`.

R. D. Gill, M. J. van der Laan, and J. R. Robins. Coarsening at random: Characterizations, conjectures and counter-examples. In D. Y. Lin and T. R. Fleming, editors, *Proceedings of the First Seattle Symposium in Biostatistics, 1995*, Springer Lecture Notes in Statistics, pages 255–294, 1997.

S. Keleş, M. J. van der Laan, and S. Dudoit. Asymptotically optimal model selection method for regression on censored outcomes. Technical Report

124, Division of Biostatistics, University of California, Berkeley, 2003a. URL www.bepress.com/ucbbiostat/paper124/.

S. Keleş, M. J. van der Laan, S. Dudoit, M. B. Eisen, and B. Xing. Supervised detection of regulatory motifs in DNA sequences. *Statistical Applications in Genetics and Molecular Biology*, 2(1), 2003b. Article 5.

S. Keleş, M. J. van der Laan, and M. B. Eisen. Identification of regulatory elements using a feature selection method. *Bioinformatics*, 18:1167–1175, 2002.

C. E. Lawrence and A. A. Reilly. An expectation maximization (em) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins: Structure, Function and Genetics*, 7:41–51, 1990.

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

T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. R. Harbison, C. M. Thompso n, Simon I., Zeitlinger J., E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J. Tagne, Volkert T. L., E. Fr aenkel, Gifford D. K., and R. A. Young. Transcriptional regulatory networks in *Saccharomyces cerevisiae* . *Science*, 298:799–804, 2002.

L. A. Mirny and M. S. Gelfand. Structural analysis of conserved base pairs in protein-DNA comple xes. *Nucleic Acids Research*, 30(7):1704–1711, 2002.

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*, 2004. (Accepted).

A. M. Molinaro and M. J. van der Laan. A Deletion/Substitution/Addition algorithm for partitioning the covariate space in prediction. Technical report, Division of Biostatistics, UC Berkeley, 2003. (In preparation).

J. Robins and A. Rotnitzky. *Recovery of information and adjustment for dependent censoring using surrogate markers*, chapter AIDS Epidemiology, Methodological issues. Bikhauser, 1992.

S. Sinisi and M. J. van der Laan. A general Deletion/Substitution/Addition algorithm in prediction. Technical report, Division of Biostatistics, UC Berkeley, 2003. (In preparation).

S. L. Teng, S. Dudoit, and M. J. van der Laan. Loss-based measures of variable importance. Technical report, Division of Biostatistics, University of California, Berkeley, 2003. (In preparation).

T. Therneau and E. Atkinson. An introduction to recursive partitioning using the rpart routine. Technical Report 61, Section of Biostatistics, Mayo Clinic, Rochester, 1997.

M. J. van der Laan and S. Dudoit. Unified cross-validation methods for selection among estimators: Finite sample results, asymptotic optimality, and applications. Technical Report 130, Division of Biostatistics, University of California, Berkeley, 2003. URL `www.bepress.com/ucbbiostat/paper130/`.

M. J. van der Laan, S. Dudoit, and S. Keleş. Asymptotic optimality of likelihood based cross-validation. Technical Report 125, Division of Biostatistics, University of California, Berkeley, 2003a. URL `www.bepress.com/ucbbiostat/paper125/`.

M. J. van der Laan, S. Dudoit, and S. Keleş. Asymptotic optimality of likelihood based cross-validation. *Biometrika*, 2003b. (Submitted).

M. J. van der Laan and J. Robins. *Unified Methods for Censored Longitudinal Data and Causality*. Springer, 2002.