BIOINFORMATICS ORIGINAL PAPER Vol. 26 no. 10 2010, pages 1357–1363
doi:10.1093/bioinformatics/btq142

Data and text mining

Advance Access publication April 7, 2010

partDSA: deletion/substitution/addition algorithm for partitioning the covariate space in prediction

Annette M. Molinaro[1,*], Karen Lostritto[1] and Mark van der Laan[2]

[1]Division of Biostatistics, Yale University Schools of Public Health and Medicine, 60 College St., New Haven, CT 06519 and [2]Division of Biostatistics, University of California, Berkeley, Earl Warren Hall #7360, Berkeley, CA 94720-7360, USA

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: Until now, much of the focus in cancer has been on biomarker discovery and generating lists of univariately significant genes, as well as epidemiological and clinical measures. These approaches, although significant on their own, are not effective for elucidating the synergistic qualities of the numerous components in complex diseases. These components do not act one at a time, but rather in concert with numerous others. A compelling need exists to develop analytically sound and computationally advanced methods that elucidate a more biologically meaningful understanding of the mechanisms of cancer initiation and progression by taking these interactions into account.

Results: We propose a novel algorithm, *partDSA*, for prediction when several variables jointly affect the outcome. In such settings, piecewise constant estimation provides an intuitive approach by elucidating interactions and correlation patterns in addition to main effects. As well as generating *'and'* statements similar to previously described methods, *partDSA* explores and chooses the best among all possible *'or'* statements. The immediate benefit of *partDSA* is the ability to build a parsimonious model with *'and'* and *'or'* conjunctions that account for the observed biological phenomena. Importantly, *partDSA* is capable of handling categorical and continuous explanatory variables and outcomes. We evaluate the effectiveness of *partDSA* in comparison to several adaptive algorithms in simulations; additionally, we perform several data analyses with publicly available data and introduce the implementation of *partDSA* as an `R` package.

Availability: http://cran.r-project.org/web/packages/partDSA/index.html

Contact: annette.molinaro@yale.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on January 3, 2010; revised on March 29, 2010; accepted on March 30, 2010

1 INTRODUCTION

By pinpointing and targeting specific early events in disease development, clinicians aim toward a more preventative model of attacking cancer. These early events can be measured as genomic, epidemiologic and/or clinical variables. Genomic variables can be measured on expression or comparative genomic hybridization (CGH) microarrays, epidemiologic variables with questionnaires, and clinical variables with pathology and histology reports. These measurements are then used to predict clinical outcomes such as number of nodes involved or response to treatment as measured by decrease in tumor size.

In theory, given the interactions of biological components inherent in carcinogenesis, model building should require the examination of all possible combinations of variables. However, in practice, it is frequently impossible to search over this entire set. Tree-based methods, which rely on the construction of piecewise constant estimators avoid this problem by attempting to approximate the search using recursive binary partitioning. Perhaps the most popular methodology in use today is classification and regression trees (*CART*; Breiman *et al.*, 1984). There are three main aspects to this tree-structured estimation scheme: (i) the splitting rule for generating the candidate predictors; (ii) the selection of a 'right-sized' tree, referred to as pruning; (iii) estimation of the parameter of interest within each terminal node. Solutions to each of these problems typically involve optimization of a loss-based criterion. Subsequently, the final *CART* model is represented primarily by a list of *'and'* statements. Although the *CART* algorithm does not intentionally build *'or'* statements, it is sometimes possible post-analysis to form such conjunctions with two or more terminal nodes.

Unlike *CART*, which fits a piecewise constant estimate within every terminal node, multivariate adaptive regression splines (MARS; Friedman, 1991) fit piecewise linear functions. The result is a function that is continuous with respect to a continuous covariate. MARS was originally developed for a continuous outcome although adaptations are possible for those that are categorical.

After *CART* and *MARS*, the most analogous method in the statistical literature is *Logic Regression*, a novel algorithm that constructs predictors as Boolean combinations of binary covariates (Kooperberg *et al.*, 2001; Ruczinski *et al.*, 2003). One restriction of *Logic Regression* is that it does not allow for continuous covariates, only binary.

Therefore, to accomplish the task of aggressively searching a highly complex variable space with a variety of variable types, we propose *partDSA*: a deletion/substitution/addition algorithm for partitioning the covariate space in prediction. In addition to generating *'and'* statements, *partDSA* explores and chooses the best among all possible *'or'* statements to build the most parsimonious model. There are numerous motivating examples of this in carcinogenesis. For instance, during the cell cycle there may be several regions of DNA that have been altered leading to a gain on one chromosome and a loss on another. Although these

*To whom correspondence should be addressed.

two mutations may be mutually exclusive, the end result could be similar. As such, we would want to account for *'or'* orderings, e.g. 'loss at locus 1 *'or'* loss at locus 3 *'and'* gain at locus 2 predicts outcome of '*y*', in addition to a list of *'and'* statements. In addition, *partDSA* allows for variables measured as continuous, categorical or ordinal that include covariates from various types of arrays as well as histology, epidemiology and pathology measures.

We suggest that *partDSA* is a flexible and aggressive data-adaptive tool that, depending on the unknown underlying data-generating distribution, will perform as well as, if not better than, previously suggested approaches. For example, *CART* utilizes a limited set of moves amounting to forward selection (node splitting) followed by backward elimination (tree pruning). In contrast, our proposed algorithm will not only split partitions (nodes in tree estimation) it will also combine and substitute partitions. These additional moves will allow us to unearth intricate correlation patterns and further elucidate interactions in addition to main effects. In the following, we introduce the key elements of *partDSA* and relevant notation. We report results from both simulations and publicly available data from a cancer study. Lastly, we summarize the method and discuss the advantages and potential limitations of our method as well as the direction of future research.

2 METHODS

partDSA is based on a unified loss-based methodology for estimator construction, selection and performance assessment employing 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. Our general statistical framework of the unified loss-based methodology and its theoretical foundations are described in Van der Laan and Dudoit (2003) and, more specific to recursive partitioning, in Molinaro *et al.* (2004). To motivate the *partDSA* algorithm, we begin with the data structure and loss functions specific to univariate outcome prediction. Subsequently, *partDSA* is formally introduced, including the main components of moves, ordering, and risk functions.

Observed data structure: in the univariate outcome prediction problem, we are interested in building and evaluating the performance of a rule or procedure fitted to n independent observations, corresponding to the n independent subjects in a study. Accordingly, we observe a random sample of n *i.i.d.* observations $X_1,\ldots,X_n$, where $X=(Y,W)$ contains a univariate outcome Y and a collection of p measured explanatory variables, or features, $W=(W_1,\ldots,W_p)'$. For example, in microarray experiments W includes RNA or protein expression, chromosomal amplification and deletions or epigenetic changes; while in proteomic data, it includes the intensities at the mass over charge (m/z) values. The collection of features may also contain explanatory variables measured in the clinic and/or by histopathology such as a patient's age or tumor stage. We denote the distribution of the data structure X by F_X. The variables that constitute W can be measured on a continuous, ordinal or categorical scale. Although this covariate process may contain both time-dependent and time-independent covariates, we will focus on the time-independent W. The univariate outcome Y may be a continuous measure such as tumor size, a categorical or ordinal measure such as stage of disease, or a binary measure such as disease status. The goal of univariate outcome prediction is to predict Y from the collection of features, W, via the use of statistical learning algorithms.

Loss functions for univariate outcome prediction: *partDSA* employs loss functions for two key stages of model building: separating the observed sample into different groups (or partitions) and selecting the final prediction model. More generally, in the context of piecewise constant estimation (described in Section 2.1.1), the candidate estimators are generated by partitioning a suitably defined covariate space into disjoint and exhaustive partitions. As illustrated in Molinaro *et al.* (2004), univariate outcome prediction, multivariate outcome prediction and density estimation can each be handled by specifying a suitable loss function. Here, we will focus solely on univariate outcome prediction and refer the reader to the aforementioned for the loss-functions and implementation of the two other scenarios.

Where we observe n observations of $X_1,\ldots,X_n$, the parameter of interest, ψ_0, is a mapping $\psi:\mathcal{S}\rightarrow\mathbb{R}$, from a covariate space $\mathcal{S}$ into the real line $\mathbb{R}$. Denote the parameter space by Ψ. The parameter ψ_0 can be defined in terms of a *loss function*, $L(X,\psi)$, as the minimizer of the expected loss, or *risk*. That is, ψ_0 is such that

$$E_{F_X}[L(X,\psi_0)] = \int L(x,\psi_0)dF_X(x)$$
$$\equiv \min_{\psi\in\Psi}\int L(x,\psi)dF_X(x)=\min_{\psi\in\Psi}E_{F_X}[L(X,\psi)].$$

The purpose of the loss function L is to quantify performance. Thus, depending on the parameter of interest, there could be numerous loss functions from which to choose. If the outcome Y is continuous, frequently the parameter of interest is the conditional mean $\psi_0(W)=E[Y\,|\,W]$ which has the corresponding squared error loss function, $L(X,\psi)=(Y-\psi(W))^2$. Another common parameter of interest is the conditional median $\psi_0(W)=Med[Y\,|\,W]$, which has the corresponding absolute error loss function $L(X,\psi)=|\,Y-\psi(W)\,|$.

If the outcome Y is categorical, the parameter of interest involves the class conditional probabilities, $Pr_0(y|W)$. For the indicator loss function, $L(X,\psi)=I(Y\neq\psi(W))$, the optimal parameter is $\psi_0(W)=\text{argmax}_y Pr_0(y\,|\,W)$, the class with maximum probability given covariates W. One could also use a loss function that incorporates differential misclassification costs. Note that in the standard *CART* methodology, Breiman *et al.* (1984) favor replacing the indicator loss function in the splitting rule by measures of node impurity, such as the entropy, Gini, or twoing indices. The indicator loss function is still used for pruning and performance assessment. It turns out that the entropy criterion corresponds to the negative log-likelihood loss function, $L(X,\psi)=-\log\psi(X)$, and parameter of interest $\psi_0(X)=Pr_0(Y|W)$. Likewise, the Gini criterion corresponds to the loss function $L(X,\psi)=1-\psi(X)$, with parameter of interest $\psi_0(X)=1$ if $Y=\text{argmax}_y Pr_0(y\,|\,W)$ and 0 otherwise. These modifications amount to using different loss functions for the same parameter at different stages of the model-building process.

2.1 *partDSA*

As an alternative to previous methods for generating piecewise constant estimates, we describe a completely data adaptive, aggressive and flexible algorithm to search the entire covariate space. Here, we detail how to approximate the parameter space by a sequence of subspaces of increasing dimension and generate candidate estimators for each subspace as well as define the moves and specific ordering of *partDSA*.

2.1.1 Generating candidate piecewise constant estimators As defined, $X=(Y,W)$, where Y is the random outcome and W is a p-vector of baseline covariates. Define a countable set of *basis functions*, $\{\phi_j:j\in\mathbb{N}\}$, indexed by the non-negative integers $\mathbb{N}$. These basis functions are simply set indicators $\{\mathcal{R}_j:j\in I\}$, which form a partition of the covariate space $\mathcal{S}$, where I is an index set, $I\in\mathcal{I}$, and $\mathcal{I}$ is a collection of subsets of $\mathbb{N}$. Here, $\mathcal{R}_j$ denotes partitions of $\mathcal{S}$ that are disjoint ($\mathcal{R}_j\cap\mathcal{R}_{j'}=\emptyset$, $j\neq j'$) and exhaustive ($\mathcal{S}=\cup_{j\in I}\mathcal{R}_j$). Now, every parameter $\psi\in\Psi$ can be written (and approximated) as a finite linear combination of the basis functions

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

where for a given index set $I\in\mathcal{I}$, the coefficients $\beta=(\beta_1,\ldots,\beta_{|I|})$ belong to $B_I\equiv\{\beta:\psi_{I,\beta}\in\Psi\}\subseteq\mathbb{R}^{|I|}$. These are of the form referred to as *piecewise constant regression models* (Härdle, 1989).

The complete parameter space Ψ can be written as the collection of basis functions $\{\phi_j : j \in \mathbb{N}\}$ and represented by

$$\Psi \equiv \{\psi_{I,\beta}(\cdot) = \sum_{j \in I} \beta_j \phi_j(\cdot) : \beta \in B_I, I \in \mathcal{I}\}.$$

In general, it is not possible to consider all candidate estimators $\psi \in \Psi$. Define a *sieve*, $\{\Psi_k\}$, of subspaces $\Psi_k \subset \Psi$, of increasing dimension approximating the complete parameter space Ψ as

$$\Psi_k \equiv \left\{ \psi_{I,\beta}(\cdot) = \sum_{j \in I} \beta_j \phi_j(\cdot) : \beta \in B_I, I, |I| \leq k \right\},$$

where k denotes the index set size (i.e. number of basis functions). Now, for every k we want to find the estimator that minimizes the empirical risk over the subspace Ψ_k. That can be done by initially optimizing over the regression coefficients $\beta \in B_I$ for a given index set I and then optimizing over the index sets I.

Estimation of regression coefficients β for a given subset of basis functions: given index sets $I \in \mathcal{I}$, define I-specific subspaces

$$\Psi_I \equiv \{\psi_I, \beta : \beta \in B_I\}.$$

For each subspace Ψ_I, the regression coefficients β are estimated by minimizing the empirical risk, i.e.

$$\hat{\beta}_I = \beta_I(P_n) \equiv \text{argmin}_{\beta \in B_I} \int L(x, \psi_{I,\beta}) dP_n(x)$$

$$= \text{argmin}_{\beta \in B_I} \sum_{i=1}^{n} L(X_i, \psi_{I,\beta}),$$

where P_n denotes the empirical measure. It is possible to write the I-specific estimators as $\hat{\psi}_I = \Psi_I(\cdot|P_n) \equiv \psi_{I,\beta_{I(P_n)}}, I \in \mathcal{I}$. An example of this is with the squared error loss function; $\hat{\psi}_I$ is then the least squares linear regression estimator corresponding with the variables identified by the index set I.

Optimization over the index sets I: *partDSA* utilizes three specific moves, or step functions (described below), to generate index sets (i.e. different partitionings of the covariate space) with the goal of minimizing a risk function over all the generated index sets. For a particular partitioning, or index set, $I \in \mathcal{I}$, the *empirical risk* of the I-specific estimator is

$$I \to f_1(I) \equiv \int L(x, \hat{\psi}_I) dP_n(x)$$

$$= \frac{1}{n} \sum_{i=1}^{n} L(X_i, \hat{\psi}_I),$$

where $\hat{\psi}_I = \Psi_I(\cdot|P_n)$ is an estimator based on the empirical distribution P_n. With the empirical risk function, the algorithm searches to minimize it over all index sets I of size less than or equal to k, where $k = 1, \ldots, K$. For each k there is a best partitioning that can be denoted

$$I_k^*(P_n) \equiv \text{argmin}_{\{I:|I|=k, I \in \mathcal{I}\}} f_1(I).$$

The algorithm searches for an approximation of $I_k^*(P_n)$, which is designated as $I_k(P_n)$ and the resulting estimator as $\hat{\psi}_k = \Psi_k(P_n)$. This results in a sieve of increasingly complex estimators indexed by k.

Cross-validation is employed for two purposes: to select an optimal estimator among the candidates generated in the sieve and to assess the performance of the resulting estimator (Molinaro and Lostritto, 2010; Molinaro *et al.*, 2005). Both are based on the chosen loss function. For selecting the optimal estimator, an alternative to cross-validation would be to minimize the empirical risk as a measure of error across the entire parameter space. However, this estimate would be highly variable and too data-dependent. Van der Laan and Dudoit (2003) derive finite sample and asymptotic optimality results for the cross-validation selector for general data-generating distributions, loss functions and estimators. The implication of these results is that selection via cross-validation is adequate in thorough searches of large parameter spaces.

2.1.2 Algorithm moves In addition to generating *'and'* statements, *partDSA* explores and chooses the best among all possible *'or'* statements by employing three different step functions: DELETION, SUBSTITUTION, and ADDITION. These functions generate index sets I, i.e. partitionings, and map the index set $I \in \mathcal{I}$ of size k into sets of index sets of size $k-1$, k, and $k+1$. The goal is to use these moves to minimize a risk function based on the chosen loss over all the generated index sets. They are defined as:

- DELETION: A DELETION move forms a union of two partitions of the covariate space regardless of their spatial location, i.e. the two partitions need not be contiguous. Formally, given a particular partitioning, i.e. an index set $I \in \mathcal{I}$, which consists of k basis functions representing indicator functions of partitions, such that $|I| = k$, we define the set $DEL(I) \subset \mathcal{I}$ as that which contains all possible unions of two disparate partitions. This new set, $DEL(I)$, is of size C_2^k.
- SUBSTITUTION: A SUBSTITUTION move divides two disparate partitions into two (disjoint and mutually exhaustive) subsets each and then forms combinations of the four subsets resulting in two new partitions. Thus, this step forms unions of partitions (or subsets within the partitions) as well as divides partitions. Formally, given an index set $I \in \mathcal{I}$, where $|I| = k$, we define the set $SUB(I) \subset \mathcal{I}$ by splitting all partitions into two subsets each and subsequently forming all unique combinations of the $2k$ subsets. This new set, $SUB(I)$, is of size $6 * C_2^k$, due to the six unique combinations for every two partitions (i.e. four subsets).
- ADDITION: An ADDITION move splits one partition into two distinct partitions. Formally, given an index set $I \in \mathcal{I}$, where $|I| = k$, we define the set $ADD(I) \subset \mathcal{I}$ as that which contains two basis functions for every initial basis function. The new basis functions represent the 'best split' of the original partition into two distinct partitions. As such $ADD(I)$ is of size $2k$.

Similar to *CART*, 'best split' is the split that most decreases the residual sum of squares for the entire space. However, as all splits are individually examined it is simply the split that minimizes the within-node (i.e. partition) sum-of-squares.

By implementing these moves, a sieve of increasingly complex predictors can be generated. Each of these predictors represents the 'best' predictor of size k, where $k = 1, \ldots, K$. As previously mentioned, to select the best partitioning, i.e. index set, cross-validation is used.

A unique and highly important contribution of *partDSA* is through the DELETION and SUBSTITUTION steps as this is when unions of partitions are formed. These unions of potentially disparate partitions result in *'or'* statements. Thus, we can define subsets of the partitions $\mathcal{R}_j$ of the covariate space $\mathcal{S}$ as S_l, where each partition can be a union of several subsets or smaller partitions of the covariate space. The subsets S_l themselves are disjoint ($S_l \cap S_{l'} = \emptyset, l \neq l'$) and exhaustive ($\mathcal{S} = \cup_{l \in I} S_l$). The predicted value for each of the partitions is constant resulting in a histogram estimate of the regression surface. This allows basis functions, which may be comprised of numerous *'and'* and *'or'* statements.

2.1.3 Algorithm ordering Having outlined the three moves, DELETION, SUBSTITUTION and ADDITION and the risk functions, the ordering is the final step of the algorithm. Minimizing the empirical risk function results in a sieve of estimators indexed by k. The vector $BEST(k)$ will be used to store the estimated empirical risk corresponding with the best partitioning of size k. Given the goal of minimizing $I \to f_1(I)$, there are three steps to this process:

(1) Initiate Algorithm. The algorithm is initiated by setting the running partitioning, I_0, to the null set. For piecewise constant regression, the null set is that set which includes the entire covariate space as one partition. Then $f_1(I_0)$ is evaluated and $BEST(1)$ is given its value. A stopping value indicating the maximum number of basis functions to consider is assigned as cut-off-growth (COG).

(2) Move through Step Functions.

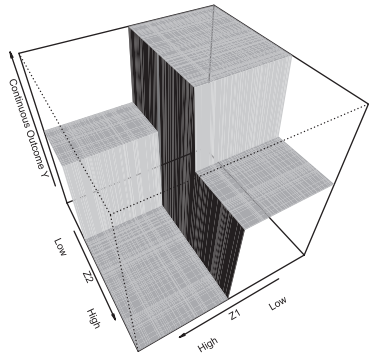

Fig. 1. Two dimensional display of *partDSA* results. Two disparate partitions form a union.

- * Set $k=|I_0|$. If $k>1$ find an optimal updated I^- of size $k-1$ among all allowed DELETION moves, where $I^- \equiv \text{argmin}_{I\in Del(I_0)} f_1(I)$. If $f(I^-) < BEST(k-1)$ then $BEST(k-1)=f(I^-)$, $I_0=I^-$ and return to *.
- Else, find an optimal updated $I^=$ of size k among all allowed SUBSTITUTION moves, where $I^= \equiv \text{argmin}_{I\in Sub(I_0)} f_1(I)$. If $f(I^=) < BEST(k)$ then $BEST(k)=f(I^=)$, $I_0=I^=$ and return to *.
- Else, find an optimal updated I^+ of size $k+1$ among all allowed ADDITION moves, where $I^+ \equiv \text{argmin}_{I\in Add(I_0)} f_1(I)$. If $f(I^+) < BEST(k+1)$ then $BEST(k+1)=f(I^+)$, $I_0=I^+$ and return to *.

(3) Stop Algorithm. If $|I|=COG$ stop the algorithm.

When the algorithm is stopped there is a list of best estimators $I_k(P_n)$, where $k=1,\ldots,COG$. As detailed in Section 2.1.1, cross-validation is used to select the best k. An example in two dimensions (i.e. with Z_1 and Z_2 as covariates) is shown in Figure 1. *partDSA* is initiated with all observations in one partition. After performing ADDITION and DELETION steps the partitioning reads:

- Given a low value for Z_1 AND a low value for Z_2, the patient has the highest value of the outcome Y.
- Given a high value of Z_1 AND a high value of Z_2, the patient has the lowest value of the outcome Y.
- Given a low value of Z_1 AND a high value of Z_2 OR a low value of Z_2 AND a high value of Z_1, the patient has an intermediate value of the outcome Y.

There are several ways to limit the number of basis functions for the risk function. The first is to restrict the minimum number of observations in a partition $\mathcal{R}_j$ or subset S_l such that no more splits can occur if that minimum is reached. In *CART*, this is referred to as 'minbucket.' The second is to require a prespecified improvement in the empirical risk before the $BEST(\cdot)$ can be replaced (we refer to this as minimum percent difference, or *MPD*). For example, for a new estimator of size k to replace the current best estimator of size k, with $MPD=0.3$, the new empirical risk must be 30% less than that of the best estimator's.

3 RESULTS

3.1 Synthetic data

To understand the flexibility and aggressive nature of *partDSA,* we compared it to the best adaptive algorithms in the statistical literature: *CART* (Breiman *et al.*, 1984), *MARS* (Friedman, 1991) and *Logic Regression* (Ruczinski *et al.*, 2003). All four algorithms can be used for modeling categorical or continuous outcomes with binary predictors; while all but *Logic Regression* can also be used with continuous predictors. In the simulations reported below, for

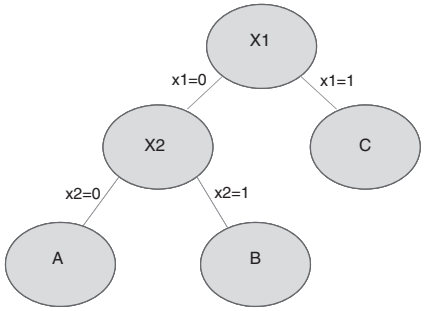

Fig. 2. Graphical display of Simulation Models 1 and 2.

Table 1. *Simulation 1 results*

Method	*partDSA*	*CART*	MARS	LR
True model size	2	3	4	2
Fitted model size	1.980	2.86	5.8	2.76
No. of vars used	1.920	1.86	3.04	2.62
No. of vars $X1, X2$ used	1.920	1.86	2	2
No. of vars $X3,\ldots,X9$ used	0	0	1.04	0.62
Error rate relative to truth	0.198	0.326	0.403	0.311
Ratio of error/*partDSA*	1	1.65	2.04	1.57

partDSA and *CART*, the minimum number of observations per node was set to 20, the maximum number of partitions to 10 and the minimum percent difference to 0.01. For both, the best model was chosen by the $1+SE$ rule suggested in Breiman *et al.* (1984). We used the `R` packages `partDSA` and `rpart`, to run *partDSA* and *CART*, respectively. For *MARS*, we used the `R` package `earth`, the penalty was set to two, the number of interactions to four, and the maximum number of model terms to 15. The best model was chosen to minimize the GCV score. For *Logic Regression*, we used the `R` package `LogicReg`, the penalty was set to two and the best model was chosen by randomization. In the results, model size refers to the number of final partitions for *partDSA*, terminal nodes for *CART*, terms for *MARS* and leaves for *Logic Regression*. In addition to model selection measures, the error rate of the fitted model relative to the true model is reported for each simulation.

The first simulation, similar in nature to that of Ruczinski *et al.* (2003), was generated with 50 training sets each containing 250 observations and nine binary covariates. Fifty independent test sets with 1000 observations each were also generated to evaluate the performance of the predictors built on the training sets. For both sets, variables $X1$ and $X3-X9$ had an independent Bernoulli(0.5) distribution. Variable $X2$ had a Bernoulli(0.2) distribution if $X1=0$ and Bernoulli(0.8) otherwise. A continuous outcome was generated from the model $Y=5*(X1\vee X2)+N(0,\epsilon)$, where $\epsilon=3$ if $(\widehat{X1}\wedge X2)$ and equals 1 otherwise. A *CART* tree representing this model is shown in Figure 2. This simulation generates most observations from either terminal node A or C with few in B. Additionally, nodes B and C have the same outcome. The smaller number of observations and higher variance (i.e. $\epsilon=3$) associated with terminal node B is a realistic scenario intended to illustrate the difference between *CART* and *partDSA*. Specifically, *partDSA* will combine terminal nodes B and C providing a more stable estimate of the outcome for those observations. The results are shown in Table 1.

The second simulation was identical in setting to the first except the outcome was binary as opposed to continuous. The outcome was generated such that Y had a Bernoulli(0.1) distribution if $(\widehat{X1}\wedge\widehat{X2})$; a Bernoulli(0.9) distribution if $X1$; or a Bernoulli(0.6) distribution if $(\widehat{X1}\wedge X2)$. The results are shown in Table 2.

Table 2. *Simulation 2 results*

Method	*partDSA*	*CART*	MARS	LR
True model size	2	3	4	2
Fitted model size	2	2.26	5.34	2
No. of vars used	1.68	1.26	3.04	2
No. of vars $X1, X2$ used	1.68	1.26	2	1.72
No. of vars $X3, \ldots, X9$ used	0	0	1.04	0.28
Error rate relative to truth	3.2%	7.2%	3.5%	4.1%
Ratio of error/*partDSA*	1	2.29	1.09	1.29

Table 3. *Simulation 3 results*

Method	*partDSA*	*CART*	MARS
True model size	2	3	4
Fitted model size	1.98	2.86	6.10
No. of vars used	1.96	1.86	2.98
No. of vars $X1, X2$ used	1.94	1.86	2
No. of vars $X3, \ldots, X9$ used	.020	0	.980
Error rate relative to truth	0.576	0.596	1.11
Ratio of error/*partDSA*	1	1.03	1.93

The third simulation includes continuous covariates. As a result, *partDSA* could only be compared to *CART* and *MARS*. Similar to the previous simulations, 50 training sets of 250 observations and 50 independent test sets of 1000 observations were generated. However, now $X1-X9$ have an independent Uniform(0,1) distribution. The outcome Y was distributed as follows: $Y = 5*(X1 \leq 0.5 \vee X2 \leq 0.15) + N(0, \epsilon)$, where $\epsilon = 4$ if $(X1 > 0.5 \wedge X2 \leq 0.15)$ and equals 1 otherwise. For *partDSA* and *CART* the minimum number of observations in each node was decreased to 15; while the penalty for *MARS* was increased to 5 and the degree was decreased to 2. The results are shown in Table 3.

In all three simulations, *partDSA* chooses the correct predictors and builds a model of the appropriate size. Additionally, in comparison to the other algorithms, *partDSA*'s selected model has the lowest error. *CART* also chooses the correct predictors but tends to build too small of a model (especially with a continuous outcome in the second simulation), which results in associated errors ranging from slightly larger to double that of *partDSA*'s. *MARS* selects the two correct predictors but also an incorrect predictor leading to a larger model than necessary in all three settings. With a binary outcome, *MARS*'s error is double that of *partDSA*'s, while it is only marginally larger with a continuous outcome. *Logic Regression* selects the two correct predictors but has a tendency to also choose an incorrect predictor. This happens more frequently with a binary outcome resulting in too large of a final model. Results from additional scenarios are shown in the Supplementary Material including simulations with continuous covariates and a binary outcome (Simulation 4), a combination of continuous and binary covariates with a continuous outcome (Simulation 5), Simulations 1–5 with 500 observations in the training set, as well as 500 variables in Simulation 4 (see, Supplementary Tables 1–8).

3.2 Data analysis

In addition to synthetic data, we evaluated the performance of *partDSA* with publicly available data. One such analysis is presented here, while the others are included in the Supplementary Material. The chosen data analysis for presentation focuses on a diffuse large B-cell lymphoma (DLBCL) study as described in Rosenwald *et al.* (2002). The purpose of this study was to determine whether the prognosis of DLBCL patients is associated with molecular features of the tumors. As only 35–40% of DLBCL patients respond to chemotherapy treatment, it has been suggested that there are meaningful subtypes of DLBCL which may predict response to treatment. Therefore, tumors from 240 patients were sampled and hemopathologists assessed the subtype of each tumor as activated B-cell like (ABC), germinal center B-cell like (GCB) and Type 3 (Type III).

Table 4. Data analysis results

Method	*partDSA*	*CART*	*MARS*
Fitted model size	4.16	2.54	6.96
No. of vars used	2.30	1.54	3.4
Error rate relative to truth	25.4%	29.9%	26.0%

For each of the patients, messenger RNA expression of 7399 genes was measured using DNA microarrays. Rosenwald *et al.* (2002) assigned subsets of the genes significantly associated with survival into four different gene signatures: germinal center B-cell (GCBcell; 151 genes), MHC class II (MHCII; 37 genes), lymph-node (LN; 357 genes) and Proliferation (Prolif; 1333 genes). For the purposes of their study, Rosenwald *et al.* (2002) used a linear combination of the four gene signatures along with the *BMP6* gene score in a Cox's proportional hazard model to predict a survival outcome. Here, we use the same gene signatures and scores as covariates in order to predict lymphoma subtype.

For the purposes of our analysis, the total 240 patients were repeatedly divided into training and test sets each of size 120. The patients were randomly assigned to the training or test sets such that each subtype was proportionally represented. This stratification allows for equal representation of all three subtypes such that classification relying on majority consensus is not biased toward any of the three (Quackenbush, 2004).

In this analysis, the prediction errors among *CART*, *partDSA*, and *MARS* were compared. For *CART* and *partDSA*, the analysis conducted employed 10-fold cross-validation in the training set to select the best number of partitions, corresponding to the first minimum over the 10-folds of estimated error. The entire training set was then used to build a model of this selected size and the risk, i.e. prediction error, was subsequently assessed on the test set. The minimum number of observations in each node was set to 20, the maximum number of possible partitions to 10, and for *partDSA*, the minimum percent difference to 0.2. For *MARS*, the penalty was set equal to two, maximum number of interactions to four and maximum model size to 15. The analysis was repeated 50 times each time with a different split of the data into a training and test set. Table 4 reports the average selected model size, the average number of unique variables and the average prediction error for the models chosen by the three algorithms.

The models from *partDSA* and *MARS* result in similar error rates; both are slightly less than that of *CART*'s. Importantly, the model size for *partDSA* is almost half that of *MARS* meaning that *partDSA* is describing the same covariate space with fewer coefficients, i.e. more efficiently. Note, a naive estimator with no coefficients, i.e. the predicted value for the test set is set equal to the most frequent outcome value, would result in a test set error of 52%.

For illustration purposes, the description of a *partDSA* model built with one split of the training/test sets follows. In this example, the best number of partitions as chosen by the first minimum of the 10-fold cross-validated error was four. The two gene signatures selected to build the partitions are GCBcell and proliferation. The final partitions and predicted subtypes can be written as:

- *Partition 1:* A GCBcell score ≤ 0.31 **AND** Prolif score ≤ -0.17 predicts the Type III subtype.
- *Partition 2:* A GCBcell score > 0.31 **AND** LymphNode score ≤ 0.17 **OR** a GCBcell score between -0.48 and 0.31 **AND** Prolif score > -0.17 predicts the GCB subtype.

Table 5. Classification of patients based on *partDSA* analysis

	Training set ($n=120$)				Test set ($n=120$)			
Prediction	ABC	GCB	III	Error	ABC	GCB	III	Error
ABC	21	1	5	0.22	21	2	6	0.28
GCB	13	48	0	0.21	14	49	3	0.26
Type III	2	9	21	0.34	2	6	17	0.32

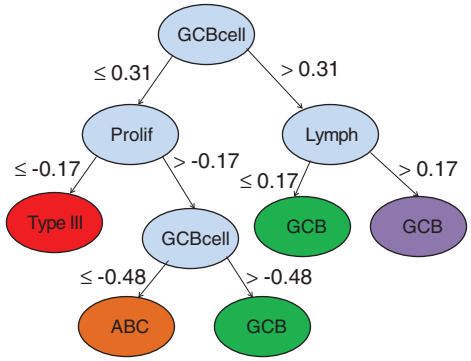

Fig. 3. Graphical display of *partDSA* data analysis. This graph shows the *partDSA* partitioning based on the three signatures: 'GCBcell', 'Prolif' and 'LymphNode'. There are four final partitions denoted by different colors: the first, in red, denotes subtype Type III; the second, in green, and the fourth, in purple, denote subtype GCB; and the third partition, in orange, denotes subtype ABC.

- *Partition 3:* A GCBcell score ≤ -0.48 **AND** a Prolif score > -0.17 predicts the ABC subtype.
- *Partition 4:* A GCBcell score > 0.31 **AND** a LymphNode score > 0.17 predicts the GCB subtype.

In Table 5, the numbers of patients within each of the lymphoma subtypes are listed in the columns separately for the training and test sets. The true subtypes are separated in rows by the *partDSA* predicted subtype. Also, included is the prediction error. From this table, partitions two and four (both predict subtype GCB) have the lowest error for the training and test sets, misclassifying 13 patients in the training set and 17 in the test set. Partition three (predicts subtype ABC) has the next lowest prediction error for both sets, misclassifying 6 out of 27 resulting in a training set error of 22% and 8 out of 29 resulting in a test set error of 28%. The numbers in the first partition reflect the increased difficulty with classifying the Type III subtype. The overall prediction error for the training set is 25% and for the test set is 27.5%.

The final model can also be shown graphically as seen in Figure 3, while a tree depicting the *CART* analysis for the same data is shown in Figure 1 in the Supplementary Material. Here, for *partDSA*, the results are shown as a tree where the four terminal partitions are represented by the colors red, green, orange and purple. The data adaptivity of *partDSA* is *visually* represented by the same green color oval representing one final partition (similar to *CART*'s terminal nodes) with predicted GCB subtype. In comparison, the *CART* tree for the same data only splits on one variable (GCBcell signature) resulting in two terminal nodes with predicted subtypes of GCB and ABC (Supplementary Figure 1). Thus, the *CART* analysis does not allot any patients for the third subtype, Type III. Importantly, note that the *partDSA* partitioning shown in Figure 3 is not equivalent to a *CART* tree. The reason is that a *CART* tree with the identical structure would estimate outcomes for two terminal nodes instead of one (as *partDSA* does) resulting in less precise and more unstable predictions of the corresponding subtypes that are also at risk of being pruned via cross-validation. Thus, by choosing the best of the *'or'* statements in the form of a single partition, *partDSA* builds a more parsimonious model that maintains a greater number of observations in the final partitions and, consequently, is better able to estimate the parameter of interest.

Table 6. Running times of *partDSA* for various datasets

Data	Set size		No. of	Time (s)	
	Training	Test	vars	Serial	Parallel
Lymphoma	120	120	5	31.21	6.80
German Br. Ca.	343	343	7	36.94	7.74
Boston housing	253	253	13	60.4	13.35
Compress. Strgth	515	515	8	70.7	16.55
Tecator	120	120	100	285.59	66.90
Tecator (PCA)	120	120	22	42.38	8.66

3.3 Implementation

partDSA is implemented as an `R` package (Molinaro *et al.*, 2009). Currently, the package accommodates continuous and categorical outcomes and covariates. The user has the choice of the v for v-fold cross-validation; size of the minimum partition, *minbucket*; largest number of partitions to explore, *COG*; minimum percent difference for splitting, *MPD*; and, loss function. The choices of loss functions are Gini and negative log-likelihood for categorical outcomes and the squared error for continuous outcomes. Additionally, *partDSA* is available in serial and parallel versions.

As *partDSA* performs an exhaustive search of the covariate space, the running time of this algorithm is an important consideration. By implementing a parallel version the running time is quite reasonable. To demonstrate, we tested six datasets (available in `R`), five of them ran in under 30 s and the sixth ran in just over a minute. All simulations were run on the Yale University Life Sciences High Performance Computing Center's Bulldogi, a cluster of 170 Dell PowerEdge 1955 nodes, each containing 2 dual core 3.0 Ghz Xeon 64 bit EM64T Intel cpus, for a total of 680 cores. Each node has 16 GB RAM. The parameter values were set to: *minbucket* $=6$, *COG* $=10$, and *MPD* $=0.1$, the defaults for the `R` package. Table 6 shows the improvement in running time between a sequential and parallel version of *partDSA*.

4 CONCLUSION

partDSA is a novel tool for generating a piecewise constant estimation sieve of candidate estimators based on an intensive and comprehensive search over the entire covariate space. The strength of this new algorithm is that it builds precise estimates of the parameter of interest based on *'and'* and *'or'* statements. These conjunctions allow combinations and substitutions of partitions for the purpose of discovering intricate correlation patterns and interactions in addition to main effects. As such, *partDSA* provides users an additional tool for their statistical toolbox.

Depending on the application, *partDSA* will supersede other methods by being not only more aggressive but also more flexible. As seen in the synthetic and real data sections, as well as the Supplementary Material, *partDSA* consistently had a lower error than *CART*, *MARS* and, when applicable, *Logic Regression*. In addition, *partDSA* selected the correct variables and built more

parsimonious models than the other algorithms. Importantly, due to *partDSA*'s formation of an *'or'* statement as a single partition, a greater number of observations are assigned to the final partitions in comparison to *CART* resulting in more precise and stable estimates of the parameter of interest (as described at the end of Section 3.2).

Due to a focus on minimizing the empirical risk of an estimator, *partDSA* can accommodate numerous settings with either continuous or categorical outcomes. As a result, the user can decide which loss function is most appropriate for their application. In the *partDSA* `R` package, the squared error loss function is implemented for continuous outcomes as well as the Gini and negative log-likelihood loss functions for categorical outcomes.

There are several directions for future work. First, with a continuous outcome, *partDSA* employs a squared error loss function for two key stages of model building: separating patients into different partitions and selecting the final prediction model. However, an immediate difficulty arises with censored survival data, for the set of observed event times cannot be treated as an uncensored sample; that is, we must modify the loss function so that it can accommodate censored outcome data in a meaningful way. Therefore, the first direction for future work is to expand *partDSA* for censored outcomes with several appropriate loss functions. Next, to assess the value of an individual variable in the current implementation, *partDSA* returns a simple frequency of the number of times a variable is selected for each partitioning. The second direction will include exploring more informative variable importance measures and including the relevant ones in the `R` package.

ACKNOWLEDGEMENTS

We would like to thank Steve Weston, REvolution Computing, Robert Bjornson and Nicholas Carriero for assistance with programming and computing advice.

Funding: National Institutes of Health National Cancer Institute (K-22 CA123146 to A.M.M.); 'Yale University Life Sciences Computing Center' and National Institutes of Health High End Shared Instrumentation Grant (RR19895 for instrumentation); National Institutes of Health National Library of Medicine (T15 LM07056 to K.L.).

Conflict of Interest: none declared.

REFERENCES

Breiman,L. *et al*. (1984) *Classification and Regression Trees*. Wadsworth and Brooks/Cole, Monterey, CA.

Friedman,J.H. (1991) Multivariate adaptive regression splines. *Ann. Stat.*, **19**, 1—141.

Härdle,W. (1989) *Applied nonparametric regression*. *Number 17 in Econometric Society Monographs*. Cambridge University Press, Cambridge, New York.

Kooperberg,C. *et al*. (2001) Sequence analysis using logic regression. *Genet. Epidemiol.*, **S1**, 626–631.

Molinaro,A.M. and Lostritto,K. (2010) Statistical bioinformatics: a guide for life and biomedical science researchers. In Jae K.Lee (ed.) *Statistical resampling for large screening data analysis such as classical resampling, Bootstrapping, Markov chain Monte Carlo, and statistical simulation and validation strategies*. John Wiley & Sons, Inc., Hoboken, New Jersey, pp.219–248.

Molinaro,A.M. *et al*. (2004) Tree-based multivariate regression and density estimation based on right-censored data. *J. Multivar. Anal.*, **90**, 154–177.

Molinaro,A.M. *et al*. (2005) Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, **21**, 3301–3307.

Molinaro,A.M. *et al*. (2009) partDSA: partitioning using deletion, substitution, and addition moves. Available at `http://cran.r-project.org/web/packages/partDSA/index.html`. (last accessed date April 9, 2010).

Quackenbush,J. (2004) Meeting the challenges of functional genomics: from the laboratory to the clinic. *Preclinica2*, **5**, 313–316.

Rosenwald,A. *et al*. (2002) The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *N. Engl. J. Med.*, **346**, 1937–1946.

Ruczinski,I. *et al*. (2003) Logic regression. *J. Comput. Graph. Stat.*, **12**, 474–511.

van der Laan,M.J. and Dudoit,S. (2003) 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.