Additive Regression Trees and Smoothing Splines - Predictive Modeling and Interpretation in Data Mining

Bin Li and Prem K. Goel

Abstract. We suggest a two-phase boosting method, called "additive regression tree and smoothing splines" (ARTSS), which is highly competitive in prediction performance. However, unlike many automated learning procedures, which lack interpretability and operate as a "black box", ARTSS allows us to (1) estimate the marginal effect smoothly; (2) test the significance of non-additive effects; (3) separate the variable importance on main and interaction effects; (4) select variables and provide a more flexible modeling strategy. We apply ARTSS to two public domain data sets and discuss the understanding developed from the model.

1. INTRODUCTION

Boosting is one of the most powerful and successful learning ideas introduced from the machine learning community. From statistics perspective, boosting can be seemed as a stagewise additive modeling strategy, see e.g. Friedman, Hastie and Tibshirani (2000). Boosting builds an additive model

$$\hat{f}(x) = \sum_{m=1}^{M} \beta_m b(x; \gamma_m), \tag{1.1}$$

where $b(x, \gamma_m)$ is a base learner parameterized with γ_m. Moreover, in the numerical optimization, stagewise additive expansion is closely related to steepest-descent minimization in function space. A general gradient-descent boosting paradigm was developed by Friedman (2001), together with a gradient tree boosting algorithm, called "multiple additive regression trees" (MART). Empirical results have shown that MART achieves highly accurate prediction performance comparing to its competitors. However, since MART uses regression trees as its base learner, the representation of the fitted model is extremely complicated. Thus we cannot discern whether the approximated function is close to a simple one, such as linear or additive, or whether it involves complex interactions among the variables.

2000 *Mathematics Subject Classification.* Primary 62-06; Secondary 62G08.

Key words and phrases. Boosting, Interpretation, MART.

This work was partially supported by the National Science Foundation under Grant CTS-0321911.

On the other hand, *simple additive models* estimate an additive approximation to the multivariate regression function without interaction. The benefits of an additive approximation are at least twofold. First, since each of the individual additive terms is estimated using a univariate smoother, the "*curse of dimensionality*" is avoided, at the cost of not being able to approximate the interaction effects. Second, estimates of the individual terms explain how the response variable changes with the corresponding independent variable. Additive smoothing spline is one of the most popular techniques for nonparametric function approximation. It provides a smooth estimated function by imposing a roughness penalty in the reproducing kernel Hilbert space, see e.g., Wahba (1990) and Gu (2002). Although the direct extension of smoothing spline to the high order interactions is straightforward in principle, but difficult in practice due to the prohibitive computational cost.

In this paper, we examine a two-phase boosting algorithm, named "Additive Regression Trees and Smoothing Splines" (ARTSS), which fits an additive model by using smoothing splines and/or *stumps* (single-split trees with only two terminal nodes) in the first phase, followed by MART in the second phase. When the underlying function is additive, ARTSS approximates the function (almost) entirely in the first phase. Numerical results indicate that when the underlying function is smooth, ARTSS is superior to MART in predictive performance. *Moreover*, ARTSS provides additional interpretation and advantages against MART, i.e., (1) better estimates of marginal effects for continuous variables; (2) indicates whether or not the underlying function is (approximately) additive; (3) separate the variable importance on main and interaction effects; (4) provide a more flexible modeling strategy.

The rest of the paper is organized as follows. In Section 2, variants of boosting algorithms involved in ARTSS are briefly described. In Section 3, the ARTSS algorithm for regression is presented, followed by the numerical results from the simulation study and a real application. In Section 4, the interpretation based on ARTSS is presented. Extension of ARTSS to classification problem is described in Section 5, followed by the applications on two real data sets in Section 6.

2. Boosting: Stagewise Additive Modeling

Given n observations of the form $\{y_i, \mathbf{x}_i\}_1^n = \{y_i, x_{i1}, \ldots, x_{ip}\}_1^n$, we consider the fundamental problem of finding a function $F(\mathbf{x})$ mapping p dimensional input vector $\mathbf{x}$ to response variable y, such that over the joint distribution of all $(y, \mathbf{x})$ values, the expected value of some prespecified loss function $L(y, F(\mathbf{x}))$ is minimized

$$F(\mathbf{x}) = arg\min_{f(\mathbf{x})} E_{y,\mathbf{x}} L(y, F(\mathbf{x})). \tag{2.1}$$

Boosting approximates $F(\mathbf{x})$ by an additive expansion in (1.1), where the expansion coefficients $\{\beta_m\}_1^M$ and the parameters $\{\gamma_m\}_1^M$ are jointly fit to the training data in a forward "stagewise" fashion. One starts with an initial guess $\hat{f}_0(\mathbf{x})$, and then for $m = 1, \ldots, M$

$$(\beta_m, \gamma_m) = arg\min_{\beta,\gamma} \sum_{i=1}^{n} L(y_i, \hat{f}_{m-1}(\mathbf{x}_i) + \beta b(\mathbf{x}_i; \gamma)), \tag{2.2}$$

and

$$\hat{f}_m(\mathbf{x}) = \hat{f}_{m-1}(\mathbf{x}) + \beta_m b(\mathbf{x}; \gamma_m). \tag{2.3}$$

Note that the *stagewise* strategy is different from *stepwise* approaches that readjust previously entered terms when new ones are added. The nature of stagewise fitting contributes to a large extent for boosting's resistent-to-overfitting property, see e.g. Friedman et al. (2000).

2.1. Gradient Descent Boosting and MART. In gradient descent boosting (Friedman, 2001), it solves (2.2) approximately for any differentiable loss function $L(y, f(\mathbf{x}))$ in two phases, based on the *pseudo-responses*

$$\tilde{y}_{im} = -\left[\frac{\partial L(y_i, f(\mathbf{x}_i))}{\partial f(\mathbf{x}_i)}\right]_{f(\mathbf{x})=\hat{f}_m(\mathbf{x})}. \tag{2.4}$$

First, the basis function $b(\mathbf{x};\gamma)$ is fit by the least square

$$\gamma_m = arg\min_{\gamma,\delta} \sum_{i=1}^{n} [\tilde{y}_{im} - \delta b(\mathbf{x}_i;\gamma)]^2. \tag{2.5}$$

Second, given $b(\mathbf{x};\gamma_m)$, the optimal value of the expansion coefficient β_m is determined

$$\beta_m = arg\min_{\beta} \sum_{i=1}^{n} L(y_i, \hat{f}_{m-1}(\mathbf{x}_i) + \beta b(\mathbf{x}_i;\gamma_m)). \tag{2.6}$$

In MART, the base learner $b(\mathbf{x};\gamma)$ is a H terminal node regression tree. At each iteration m, a regression tree partitions the $\mathbf{x}$ space into H-disjoint regions $\{R_{hm}\}_{h=1}^{H}$ and predicts a constant γ_{hm} in each region. The detailed algorithm for MART is the following.

MART Algorithm (Friedman 2001)

(1) $\hat{f}_0(\mathbf{x}) = arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$.
(2) Repeat for $m = 1, 2, \ldots, M$:
  (a) $\tilde{y}_i = -\left[\frac{\partial L(y_i, f(\mathbf{x}_i))}{\partial f(\mathbf{x}_i)}\right]_{f(\mathbf{x})=\hat{f}_{m-1}(\mathbf{x})}$, $i = 1, 2, \ldots, n$.
  (b) $\{R_{hm}\}_1^H = H$-terminal node tree based on $\{\tilde{y}_{im}, \mathbf{x}_i\}_1^n$.
  (c) $\gamma_{hm} = arg\min_{\gamma} \sum_{\mathbf{x}_i \in R_{hm}} L(y_i, \hat{f}_{m-1}(\mathbf{x}_i) + \gamma)$.
  (d) $\hat{f}_m = \hat{f}_{m-1} + \nu \cdot \gamma_{hm} I(\mathbf{x} \in R_{hm})$.
(3) End algorithm.

The ν in the above is the "shrinkage" parameter between 0 and 1 and controls the *learning rate* of the procedure. Empirical results have shown (see e.g., Friedman, 2001) that small values of ν *always* lead to better generalization error. The choice of M, i.e., when to stop the algorithm, is based on monitoring the estimation performance on a separate validation set. In this paper, MART is run by using the **gbm** package in **R**, produced by Greg Ridgeway.

2.2. Boosting with Componentwise Smoothing Spline. Bühlmann and Yu (2003) proposed L_2Boost, boosting with squared error loss, with componentwise cubic smoothing splines as base learners. The functional class for the cubic smoothing splines, known as *Sobolev space*, is defined on the interval $[a, b]$ as

$$\mathcal{W}_2 = \{f : f' \text{ absolutely continuous and } \int_a^b [f''(x)]^2 dx < \infty\}. \tag{2.7}$$

The smoothing spline solution is the $\hat{f}_\lambda(x)$ minimizing

$$\frac{1}{n}\sum_{i=1}^{n}[y_i - f(x_i)]^2 + \lambda \int_a^b [f''(x)]^2 dx. \tag{2.8}$$

If all the independent variables are continuous, componentwise boosting with smoothing splines builds an additive model with univariate smoothing splines as the base learners. Bühlmann and Yu (2003) showed that L_2Boost (boosting with L_2 loss) with componentwise smoothing splines achieves optimal convergence rate in one-dimensional case, and also adapts to higher-order smoothness.

2.3. Boosting with Subsampling. It has been shown that the both prediction accuracy and execution speed of boosting can be substantially improved by incorporating randomization into the procedure, see Breiman (1999) and Friedman (2002). Breiman (1999) proposed a hybrid bagging-boosting procedure, called "adaptive bagging", to fit the additive expansions in (1.1), i.e., fit the base leaner $\beta_m b(x; \gamma_m)$ in each iteration based on the bootstrapped training samples. Friedman (2002) proposed a slightly different approach, called *stochastic gradient boosting*, to incorporate randomness into the procedure. Specifically, at each iteration a subsample of the training data is drawn at random *without replacement* from the training set to fit the base learner. An **R** implementation of MART in **gbm** package contains the stochastic gradient boosting algorithm on half of the subsample as the default procedure.

3. Additive Regression Trees and Smoothing Splines in Regression

In ARTSS, two phases are employed to approximate $F(\mathbf{x})$. The first phase approximate the additive function $F^A(\mathbf{x}) = c + \sum_{j=1}^{p} F^A(x_j)$ that minimizes the loss function over the joint distribution of all $(y, \mathbf{x})$ by using boosting with componentwise smoothing splines and/or stumps. The second phase tryies to recover the difference between $F(\mathbf{x})$ and $F^A(\mathbf{x})$ by using MART. Thus, the base learners in the first phase of ARTSS are one dimensional, i.e., componentwise smoothing spline for continuous variable and stump otherwise, whereas the base learners in the second phase are regression trees. Throughout the paper, squared error loss is used for the regression problem. The ARTSS algorithm is outlined as follows.

ARTSS Algorithm in Regression

Phase 1.

(1) $\hat{f}_0(\mathbf{x}) = mean\{y_i\}_{i=1}^n$.
(2) Repeat for $m = 1, 2, \ldots, M$:
    (a) Set the current residual $r_i = y_i - \hat{f}_{m-1}(\mathbf{x}_i)$, $i = 1, \ldots, n$.

(b) $\{\pi(i)\}_1^n$ = bootstrap sample of $\{i\}_1^n$.

(c) $(\gamma_m, j_m) = \underset{\gamma, j \in \{1,\ldots,p\}}{\arg\min} \sum_{i=1}^{n} [r_{\pi(i)} - b(x_{\pi(i)j}; \gamma)]^2$.

(d) $\hat{f}_m = \hat{f}_{m-1} + \nu \cdot b(x_{j_m}; \gamma_m)$.

(3) $r_i = y_i - \hat{f}_M(\mathbf{x}_i)$, $i = 1, \ldots, n$.

Phase 2.
Apply MART on $\{\mathbf{x}_i, r_i\}_1^n$.

3.1. Regularization in ARTSS. It is well known that regularization is an important issue in boosting. In general, there are at least three ways for regularization. (1) Control the complexity of base learners. For the regression trees, we can control the complexity by restricting either the number of terminal nodes or the depth of trees. For the smoothing splines, we can constrain the degree of freedom, i.e., trace of the smoother matrix. Bühlmann and Yu (2003) showed the effect of controlling the degree of freedom for the cubic smoothing splines on generalization squared error. Their results suggest that using the small values for the degree of freedom achieves better generalization error than using large values. In this paper, we fix the degree of freedom to be three (same as the one they used). In MART, the optimal (in terms of prediction) complexity of trees depends on the underlying function $F(x)$, and can be chosen based on an independent validation set. (2) Control the learning rate ν of the boosting procedure. It is well known that using small ν never hurts but usually improves prediction accuracy substantially. In this paper, ν is fixed at 0.05. (3) Control the number of iterations M in boosting procedure. The stopping criterion for both the first and second phases in ARTSS is based on monitoring the estimation performance on a separate validation set.

3.2. Data Splitting Strategy in ARTSS. In ARTSS fitting, a separate validation set is needed to set the tuning parameters, such as the number of iterations in each phase. There are two approaches to split the data. First, one can allocate a subset of data as the validation set, where the rest as the training set. Second, one can implement k-fold cross validation strategy in ARTSS fitting. Namely, partition the training set into k subsets of (approximately) equal size, and fit the ARTSS model k times, each time leaving out one of the subsets from training, but using only the omitted subset to tune the parameters. The final model is the average of these k fitted models. This approach is especially useful for small data sets. In this paper, without specifying a validation set, the five-fold ($k = 5$) cross validation strategy is used in ARTSS fitting.

3.3. Numerical Results. Although the main objective of this paper is in interpretation, prediction is always considered to be one of the most important criteria to evaluate a data mining method. It is well known that characteristics of problems will affect the prediction performance of a method. Two of the most important characteristics of any problem affecting performance are the underlying true function $F(x)$ and the joint distributions of input variables. In order to gauge the value of any estimation method, it is necessary to accurately evaluate its performance over different situations. This is most conveniently accomplished through Monte Carlo simulation, where data can be generated according to a wide variety

of prescriptions, and resulting performance accurately calculated. The procedure used here to generate test functions is based on the random function generator described in Friedman (2001). Two settings are considered here so that the underlying true function is (i) non-additive (with interaction) and (ii) additive (without interaction).

The following true model is used to generate simulated data:

$$y = \sum_{l=1}^{20} a_l g_l(\mathbf{z}_l) + \sigma\epsilon, \quad \epsilon \sim N(0,1), \tag{3.1}$$

where σ was chosen to have signal-to-noise ratio (SNR) equal to two, i.e., $\sigma^2 = var(y)/5$. The coefficients $\{a_l\}_1^{20}$ are randomly generated from a uniform distribution between -1 and 1. Each $g_l(\mathbf{z}_l)$ is a function of a randomly selected subset, of size p_l, of the p-input variables $\mathbf{x}$. Specifically,

$$\mathbf{z}_l = \{x_{\pi_l(j)}\}_{j=1}^{p_l}, \tag{3.2}$$

where each π_l is a random permutation of the integers $\{1, \ldots, p\}$. In the non-additive setting, the size of each subset p_l is itself random, $p_l = \lfloor 1.5 + u \rfloor$, where $\lfloor t \rfloor$ denotes the integer part of t, with u being drawn from an exponential distribution with mean equal to two. In the additive setting, $p_l \equiv 1$. Each $g_l(\mathbf{z}_l)$ is an p_l-dimensional Gaussian function

$$g_l(\mathbf{z}_l) = exp(-\frac{1}{2}((\mathbf{z}_l - \mu_l)^T \mathbf{V}_l (\mathbf{z}_l - \mu_l)), \tag{3.3}$$

where each of the mean vectors $\{\mu_l\}_1^{20}$ is randomly generated from the same distribution as that of the input variables $\mathbf{x}$. The $p_l \times p_l$ covariance matrix $\mathbf{V}_l$ is also randomly generated. Specifically, $\mathbf{V}_l = \mathbf{U}_l \mathbf{D}_l \mathbf{U}_l^T$, where $\mathbf{U}_l$ is a random orthogonal matrix (uniform on Harr measure) and $\mathbf{D}_l = diag\{d_{1l}, \ldots, d_{p_l l}\}$. The square root of the eigen values are randomly generated from a uniform distribution between 0.1 and 2.0. The joint distribution of $\mathbf{x}$ is multivariate normal distribution with mean zero and covariance matrix $\mathbf{\Sigma}$, where $\Sigma(a,b) = \rho^{|a-b|}$. ρ is randomly generated from a uniform distribution between -0.5 and 0.5. In this study, p is fixed at ten.

In the experiment, we compared the prediction performance of ARTSS with MART and MARS (multivariate adaptive regression splines) proposed by Friedman (1991). MARS is run by using the **mda** package in **R**. To compare the prediction performance, we use the *comparative test error*, defined by

$$c_{i,j} = \frac{MSE_{ij}}{min\{MSE_{i,l}\}_{l=1,2,3}}, \; i = 1, \ldots, 100, \; j = 1, 2, 3 \tag{3.4}$$

over 100 replications for each method. This quantity facilitates individual comparisons by using the error of the best method for each data set to calibrate the difficulty of the problem. The training set for each replication contains either 300 or 1000 observations, while testing set consists of 2000 observations.

Figure 1 shows the comparative test errors for ARTSS, MART and MARS over 100 replications each in non-additive and additive setting. Simulation results shown in Figure 1 imply that: (1) ARTSS achieves better prediction performance than MART and MARS under both additive and non-additive settings; (2) the benefit of using ARTSS as compared to MART is larger under the additive (small sample size) case than the non-additive (large sample size) case.

In addition, we consider the often-analyzed dataset of ozone concentration in the Los Angeles basin, which has been also been considered by Breiman (1998) in

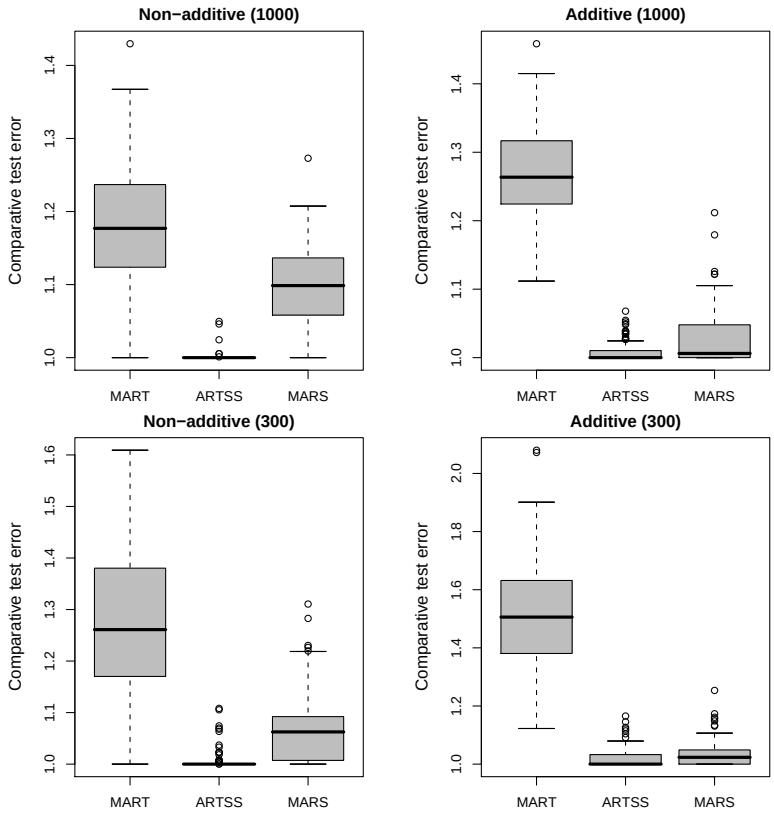

FIGURE 1. Boxplots of comparative test errors for ARTSS, MART and MARS. Training sets in the upper (lower) panel contain 1000 (300) observations.

connection with boosting. There are nine input variables and the sample size is 330. Besides ARTSS, MART and MARS, here we also consider smoothing splines (without interaction), which is run by **gss** package in **R**, L_2Boost with stumps, and *additive ARTSS* (ARTSS with only the first phase). We estimate the testing error by randomly splitting the data into 300 training and 30 testing observations and simulating 50 such random partition. Table 1 displays the mean and standard deviation (in parenthesis) of the test MSE over 50 runs. We conclude that ARTSS with/without the second phase is better than MART and L_2Boost with stumps, and have comparable performance to cubic smoothing spline and MARS. This is because ARTSS, MARS and smoothing splines fit smooth surface rather than a non-smooth surface from regression trees. Furthermore, since the sample size is not large, it may not be possible for MART to reliably estimate models with large number of trees. A referee pointed out the different results between ARTSS and cubic smoothing, although both conduct additive smoothing splines. This may because the ARTSS tends to fit a sparse model, e.g. most of the ARTSS models (without second phase) didn't include variable 'humidity' and 'Inversion base temperature' among 50 runs on Ozone data. Like MART, due to the small sample size, ARTSS may not reliably estimate model with large number of trees in the second phase.

Method	MSE ($\times 10^{-2}$)
Additive ARTSS	11.79 (3.05)
ARTSS	11.73 (3.05)
MART	14.06 (4.49)
L_2Boost with stumps	15.06 (4.35)
MARS	12.56 (3.05)
Cubic smoothing splines	11.63 (2.86)

TABLE 1. Comparison of Test Set MSEs for Ozone Data.

3.4. Computational Consideration. Consider the n observations with p predictors. The computation for the univariate smoothing splines can be of the order $O(n)$, see e.g. Hutchison and de Hoog (1985). On the other hand, the computation for trees is of the order $O(pnlogn)$. In general, ARTSS has the same order of computation as MART.

3.5. Asymptotic Property of ARTSS. Stone (1985) showed that under a common smoothness assumption the additive spline estimates achieve the same optimal convergence rate as they do in one-dimensional case. This indicates that the ARTSS estimate $\hat{f}(\mathbf{x})$ without the second phase also enjoys the optimal convergence rate. Note that this estimate converges to $F^A(x)$, the closest additive function in terms of squared error loss, but not the target function $F(x)$ itself. On the other hand, Bühlmann (2002) showed the consistency results in both regression and classification for L_2Boost with tree-type basis functions under some regularity conditions. Note that this result doesn't require the target function to be smooth, and even the predictor can be finite categorical variable. Combining the consistency result from Stone (1985) with that of Bühlmann (2002) implies the consistency of ARTSS under some smoothness and regularity conditions.

4. Interpretation in ARTSS

In many applications, it is highly desirable to be able to interpret the derived approximation $\hat{f}(\mathbf{x})$. This includes gaining an understanding of those particular input variables that are most influential in contributing to its variation, and the nature of the dependence of $\hat{f}(\mathbf{x})$ on those influential inputs. Moreover, sparseness and hierarchical structure are two popular assumptions in statistical model fitting. By the two-phase nature of ARTSS fitting, these two assumptions can be naturally implemented in ARTSS.

4.1. Estimate Marginal Effect. As defined at the beginning of Section 3, $F^A(\mathbf{x})$, the "best" additive function, is sum of a constant c plus $\sum_{j=1}^{p} F^A(x_j)$, subject to the constraints that $EF^A(x_j) = 0$ for $1 \le j \le p$. Moreover, we defined the marginal effect of x_j as $F^A(x_j)$. Thus, if the underlying function $F(\mathbf{x})$ is additive, then the marginal effect is the same as the corresponding component in $F(\mathbf{x})$. In ARTSS, the estimate of marginal effect for x_j, $\hat{f}^A(x_j)$, is simply the sum

of all the additive terms due to x_j in the first phase

$$\sum_{m=1}^{M} \nu \cdot b(x_{j_m}; \gamma_m) \cdot \mathbf{1}(j_m = j). \tag{4.1}$$

Friedman (2001) introduced *partial dependence* to describe the dependence of response variable on a subset of variables. Given any subset $\mathbf{x}_s$ of the input variables indexed by $s \subset \{1, \ldots, p\}$. The partial dependence is defined as

$$F_s(\mathbf{x}_s) = E_{\mathbf{x}_{\setminus s}}[f(\mathbf{x})], \tag{4.2}$$

where $E_{\mathbf{x}_{\setminus s}}[\cdot]$ means expectation over the joint distribution of all the input variables with index not in s. Although $F^A(x_j)$ is defined different from $F_j(x_j)$, they coincide with when the underlying function $f(\mathbf{x})$ is additive or all the input variables are independent. In practice, partial dependence can be estimated from the data by

$$\hat{F}_s(\mathbf{x}_s) = \frac{1}{n} \sum_{i=1}^{n} \hat{f}(\mathbf{x}_s, \mathbf{x}_{i\setminus s}), \tag{4.3}$$

where $\{\mathbf{x}_{i\setminus s}\}_1^n$ are the data values of $\mathbf{x}_{\setminus s}$. By the nature of tree-based methods, the data-sparse region does not have a chance to split and is fitted by a constant no matter what the underlying function is. We call this the "*low probability effect*" on tree-based methods.

To illustrate, we simulate data from an additive model with two input variables

$$y = -2x_1 + x_2 + \sigma\epsilon, \quad \epsilon \sim N(0, 1),$$

where σ is chosen to have SNR=2. The input variables $(x_1, \ldots, x_5)$ are generated from the multivariate normal distribution with mean zero and covariance matrix $\mathbf{\Sigma}$, where $\Sigma(a, b) = 0.3^{|a-b|}$. Two scenarios with different sample sizes are considered. **Case 1:** the model is fitted on a training set with 50 observations and an independent validation set with 50 observations is used to determine when to stop the algorithm. **Case 2:** both training and validation sets have 500 observations.

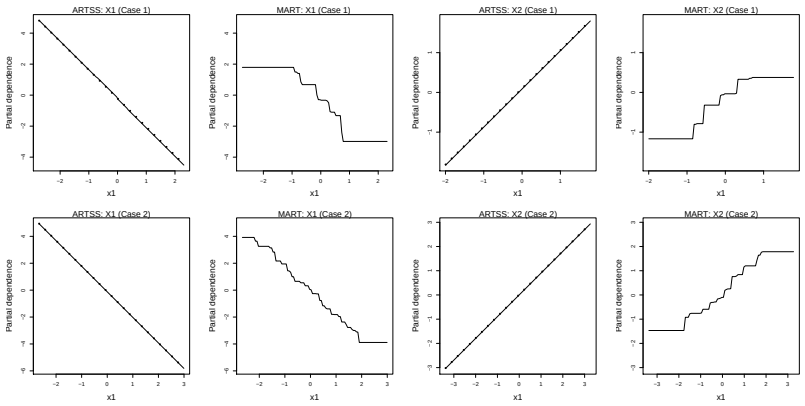

FIGURE 2. Low probability effect on MART. The upper (lower) row shows the estimated partial dependence in Case 1 (2). The black dot lines are the estimated marginal effects defined in (4.1). The gray solid lines are estimated partial dependence.

Ideally, both ARTSS and MART should recover the additive linear structure for the underlying true function. Figure 2 compares the estimated marginal effects of x_1 and x_2 from ARTSS and MART, and shows the low probability effect in MART. Based on Figure 2, we conclude that (1) the estimated marginal effects by ARTSS approximate the underlying truth well; (2) the estimated partial dependence defined in (4.2) is very close to the corresponding estimate of marginal effect defined in (4.1), because the underlying function is additive and ARTSS approximates the underlying function almost entirely in the first phase; (3) MART fail to recover the linear partial dependence faithfully, e.g., in the sparse area $|x_j| > 2$, $j = 1, 2$, the estimated partial dependence is constant; (4) due to the curse of dimensionality, we expect the low probability effect to be even more common and severe in high-dimensional problems. On the other hand, since ARTSS estimates the marginal effect in the first phase, it will be less affected in high dimension.

4.2. Test Significance of Non-additive Effect. Consider an "ANOVA" expansion of a function

$$f(\mathbf{x}) = \sum_j f_j(x_j) + \sum_{j,k} f_{jk}(x_j, x_k) + \sum_{j,k,l} f_{jkl}(x_j, x_k, x_l) + \dots. \tag{4.4}$$

The first sum consists of functions that each depends on only one input variable. The particular functions $\{f_j(x_j)\}_1^p$ in ARTSS provide the "closest" approximation to $f(\mathbf{x})$ under the additive constraint. The rest sums consists of functions that each depends on more than one input variables. In ARTSS, these sums are approximated in the second phase by using MART. In order to test the significance of non-additive effect, we present an approach based on Friedman and Popescu's work (2005).

Consider the model is fitted on the training set, while a validation set is used to determine where to stop the algorithm. First, a collection of artificial data sets $\{\mathbf{x}_i, \tilde{y}_i\}$ are generated from the training and validation data sets as follows.

$$\tilde{y}_i = \hat{f}^A(\mathbf{x}_i) + (y_{p(i)} - \hat{f}^A(\mathbf{x}_{p(i)})), \quad i = 1, \dots, n. \tag{4.5}$$

Here $\{p(i)\}$ represents a random permutation of the integers $\{1, 2, \dots, n\}$ (n is number of observations in the training and validation set), and $\hat{f}^A(\mathbf{x})$ is the main effects estimate from the first phase (ARTSS without second phase). Note that every data set generated in (4.5) contains no interaction effects, and $\hat{f}^A(\mathbf{x})$ is the underlying true function. In addition, the joint distribution of predictor variables is same as that of the original data. Then, for each artificial data set and the original data, we apply ARTSS and calculate $P2$, the proportion of reduction in validation error from Phase 2:

$$P2 = \frac{\text{reduction of the validation error in Phase 2}}{\text{total reduction of the validation error}}. \tag{4.6}$$

The collection of these $P2$ values can be considered as a reference distribution under the null hypothesis that the underlying function is additive. Finally, an empirical p-value, the proportion of artificial data sets that resulted in $P2$ greater than that for the original data, provides a test statistic for the significance of non-additive effects.

As an illustration, we did the following simulations. The data is simulated from the true model

$$y = f(\mathbf{x}) + \sigma\epsilon, \quad \epsilon \sim N(0, 1), \tag{4.7}$$

where σ was chosen to obtain specified values of two-to-one SNR. Two examples are presented here. In each example, it consists of a training set and an independent validation set. The joint distributions of input variables are the same in two examples. The first five input variables are continuous variables generated from the uniform $(0, 1)$ distribution, whereas the rest five are categorical variables, all of which have four categories, denoted as $1 \sim 4$, with equal probabilities.

Example 1: Additive case. The data set consists of 200 observations in both training and validation set. The underlying function $f(\mathbf{x})$ is

$$\begin{aligned} f(\mathbf{x}) &= 10(x_1 - 0.5)^2 - x_2 + sin(2\pi x_2) - 3x_3 \\ &+ 1.5I\{(x_6 = 3) \cup (x_6 = 4)\}, \end{aligned} \tag{4.8}$$

where $I\{A\}$ is an indicator function, equal to one when A holds, otherwise zero.

Example 2: Non-additive case. The data set consists of 1000 observations in the training set and has the same amount in the validation set. The underlying function is

$$f(\mathbf{x}) = exp(-x_1 - x_2 + 2x_3) + 3(x_4 - x_5) + 2I\{(x_6 \geq 3) \cap (x_7 \leq 2)\}. \tag{4.9}$$

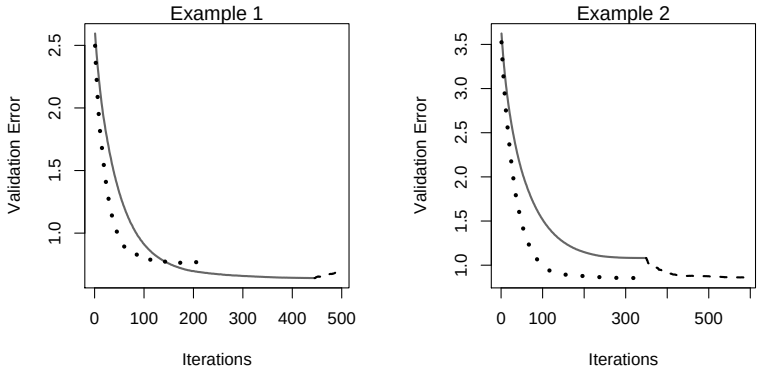

FIGURE 3. Validation MSE for ARTSS (solid/dash corresponds to the first/second phase) and MART (dot).

Figure 3 shows the decreasing patterns in validation errors for ARTSS and MART on two examples. In the first example, where the underlying function is additive, the first phase of ARTSS contributes all the reduction of validation errors. In the second phase, the validation errors even increase slightly before stopping. In the second example, we see an obvious dip from the second phase, which accounts for about 9% of total reduction in validation error. This agrees with the non-additive nature of the underlying function. By applying the proposed test procedure based on 100 random permutations, we have the empirical p-value for Example 1 is 0.28, whereas zero in Example 2.

4.3. Relative Variable Importance. In general, there are at least three ways to measure the importance of input variables. The first is causality, which can only be inferred if the data set comes from a well-designed experiment. A second

approach is based on face value interpretation, i.e., to look at the variables that are important in determining the face value of the estimated function $\hat{F}(x)$, (see e.g., Breiman 2001). Face value interpretation method tends to ignore the dependence structure among predictors. The third way is based on the out-of-sample predictions with/without a certain variable. This is commonly addressed by fitting the model multiple times using different subsets of predictors, and comparing the prediction accuracy on a test set (see e.g., John, Kohavi and Pfleger 1994).

By the two-phase procedure in ARTSS, we can measure the relative variable importance (RVI) on main (mRVI) and interaction (iRVI) effects, separately. Here we suggest a way to measure the mRVI based on the first phase in ARTSS. Although our approach is also based on out-of-sample predictions, it doesn't require fitting the model multiple times employing different subsets of the input variables. To measure the RVI on interaction effect, we use Friedman (2001) approach of estimating RVI in MART.

Let v_0 be the validation error for the initial estimate and v_m be the validation error at the m^{th} iteration in the first phase of ARTSS, $m = 1, \ldots, M$. We measure the mRVI for x_l as follows.

$$mRVI(x_l) = \frac{\sum_{m=1}^{M} \mathbf{1}(j_m = l)(v_{m-1} - v_m)}{\sum_{m=1}^{M}(v_{m-1} - v_m)}. \tag{4.10}$$

From (4.10), we see that mRVI of x_l is essentially the proportion of reduction in validation error accounted by x_l in the first phase.

Figure 4 displays the estimated RVI in Example 1 and 2. In Example 1, we see that x_1, x_2, x_3 and x_6 dominate the main effects. However, x_4 and x_7 are also selected in the first phase, but their mRVIs are very close to zero (note that their mRVIs are even negative). In Example 2, only the first seven variables have positive mRVI. Among them, x_4 and x_5 have the largest mRVI. However, their iRVI is smaller than the other five variables. This agrees with the fact that x_4 and x_5 are additive, while the other five are non-additive in the underlying function. Note that it is important to know the sensitivity of the inference with respect to the tuning parameters, including the number of iterations in both phases. Figure 5 shows the estimated mRVI paths as a function of number of iterations (starting with the 21^{st} iteration) in Example 1 and 2. We see that the estimate of mRVI have stabilized much sooner than the end of the Phase 1. We expect to conduct further studies on sensitivity to other parameters.

4.4. Variable Selection. Like stagewise regression, ARTSS tends to select only a small subset of variables in the first phase, when the true model is sparse. However, variables not in the true model can also be selected with small or even negative mRVI. In order to reduce the false discovery rate (FDR), we can select the variable via thresholding on mRVI, e.g., using τ/p as the threshold where $0 \leq \tau \leq 1$. Although we don't have any theoretical justification for selecting the threshold value as of now, heuristically, using the suggested threshold can guarantee to have at least $1 - \tau$ percent of the mRVI retained after thresholding. In this paper, we set τ at 0.1.

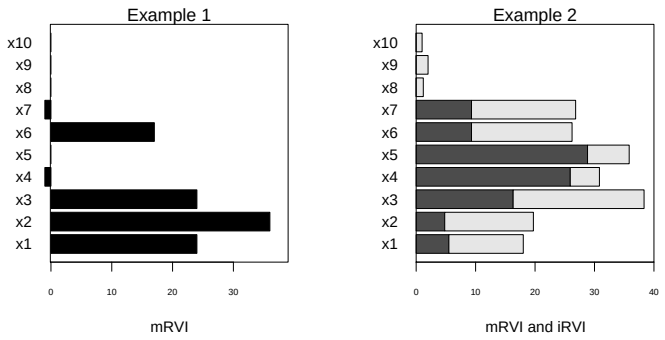

FIGURE 4. (Left) The mRVI in Example 1. (Right) The mRVI (black) and iRVI (gray) in Example 2.

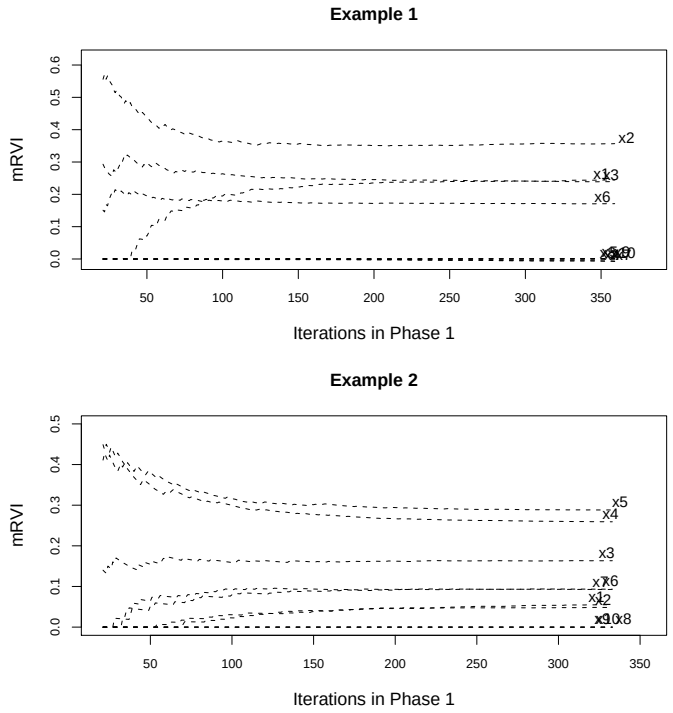

FIGURE 5. The mRVI paths in Example 1 and 2.

To illustrate, we simulated 100 replications of the data from the following true model.

$$y = \sum_{j=1}^{20} a_j \cdot x_j \cdot \mathbf{1}(j \in \mathbf{S}) + \sigma\epsilon, \quad \epsilon \sim N(0,1), \tag{4.11}$$

where $a_j \overset{iid}{\sim} Unif(0.5 \leq |a| \leq 1)$, $\mathbf{S}$ is a randomly selected subset of $\{1, \ldots, 20\}$ with size $|\mathbf{S}|$, and σ is chosen to have SNR=2. The size of $\mathbf{S}$ itself is random,

$|\mathbf{S}| = \lfloor 1.5 + u \rfloor$, with u being drawn from an exponential distribution with mean equal to five. Thus, the expected number of selected input variables into $\mathbf{S}$ is between six and seven. The joint distribution of $\mathbf{x}$ is twenty dimensional multivariate normal distribution with mean zero and covariance matrix $\mathbf{\Sigma}$, where $\Sigma(a, b) = \rho^{|a-b|}$. ρ is randomly generated from a uniform distribution between -0.5 and 0.5. Within each replication, all the parameters related to (4.11) are randomly generated with sample size 500. We apply five-fold cross-validation, and take the average as our estimate of mRVI. Table 2 shows the FDR and NDR (nondiscovery rate) in simulation. The results indicate that thresholding on mRVI makes the selected variables closer to the underlying true model (without suffering from the NDR). Interestingly, it is known that thresholding LASSO estimators achieves consistency in variable selection, see e.g., Li and Goel (2006). This seems to be what is happening here too. Since measuring the mRVI is done in the Phase 1 of ARTSS, we can choose the variables into the second phase based on mRVI, i.e., the hierarchical structure can be naturally incorporated in ARTSS.

	Before Thresholding	After Thresholding
FDR	0.66	0.0025
NDR	0	0

TABLE 2. FDR and NDR in Simulation Study

5. ARTSS in Classification

Consider the two-class classification problem, i.e., $y_i \in \{-1, 1\}$, $i = 1, \ldots, n$. The ARTSS for classification is based on the gradient boosting algorithm described in Section 2.1. Instead of squared error loss, negative log-likelihood for the binomial model is used as the loss function:

$$(5.1)\; L(y, \hat{f}) = log(1 + exp(-2y\hat{f})), \text{ where } \hat{f}(\mathbf{x}) = \frac{1}{2} log \left[\frac{Pr(y = 1|\mathbf{x})}{Pr(y = -1|\mathbf{x})} \right].$$

More specifically, within each iteration of Phase 1, the pseudo-responses are computed based on the above loss function. Then, a variable and its corresponding basis function is selected by the least-squares function minimization on the pseudo responses. The detailed algorithm for ARTSS in classification is the following.

ARTSS Algorithm in Classification

Phase 1.

(1) Initialize $\hat{f}_0(\mathbf{x}_i) = \frac{1}{2} log \left(\frac{1+\bar{y}}{1-\bar{y}} \right)$.
(2) Repeat for $m = 1, 2, \ldots, M$:
   (a) Set the *pseudo-responses* $\tilde{y}_i = - \left[\frac{\partial L(y_i, \hat{f}_{m-1}(\mathbf{x}_i))}{\partial \hat{f}_{m-1}(\mathbf{x}_i)} \right]$, $i = 1, \ldots, n$.
   (b) $\{\pi(i)\}_1^n$ = bootstrap sample of $\{i\}_1^n$.

(c) $(\mathbf{a}^*, j^*) = \underset{\mathbf{a}, j \in \{1,\ldots,p\}}{\arg\min} \sum_{i=1}^{n} [\tilde{y}_{\pi(i)} - h(x_{\pi(i)j}; \mathbf{a})]^2;$.

(d) $\hat{f}_m(\mathbf{x}) = \hat{f}_{m-1}(\mathbf{x}) + \nu h(x_{j^*}; \mathbf{a}^*)$.

(3) $\tilde{y}_i = -\left[\frac{\partial L(y_i, \hat{f}_M(\mathbf{x}_i))}{\partial \hat{f}_M(\mathbf{x}_i)}\right]$, $i = 1, \ldots, n$.

Phase 2.
Apply MART on $\{\mathbf{x}_i, \tilde{y}_i\}_1^n$.

6. Applications to Real Data

In this section, ARTSS procedure is further illustrated on two public domain data sets. In both examples, we split the data set into two parts, training and validation set. The model was fitted on the training set and evaluated on the validation set.

6.1. California Housing Data. This data set, available at the Carnegie-Mellon *StatLib* repository (http://lib.stat.cmu.edu/datasets/), was originally used by Pace and Barry (1997). It consists of aggregated data from each of 20,640 neighborhoods (1990 census block groups) in California. The response variable is the median house value in each neighborhood measured in units of \$100,000. There are eight continuous input variables: median income (denoted as Med), house age (Age), average number of rooms per person (Rms), average number of bedrooms per person (Bdrm), population (Pop), average occupancy in each house (Occ), latitude of the location of each neighborhood (Lat) and longitude (Lon). Since Rms, Bdrm, Pop and Occ have some extremely large outliers, these variables are winsorized at 99.5 percentile, i.e., all the observations above 99 percentile are set to their 99 percentile value. The logarithm transformation is applied to the response variable (see Pace and Barry, 1997) before the analysis. One third of data was randomly selected as the validation set, and the model was trained on the remaining two third.

Figure 6(a) shows the validation error for the ARTSS and MART. We see an obvious dip in the second phase, which accounts about 9.6% of total reduction in validation error. Based on the validation set, ARTSS achieves $R^2 = 82\%$, whereas MART achieves $R^2 = 81\%$. Pace and Barry (1997) applied a sophisticated spatial autoregression method and achieved $R^2 = 85\%$ for the training set. To test the significance of non-additive effect, we did the permutation test described in Section 4.2. All the one hundred $P2$ generated from randomly permutated data sets are much smaller than observed $P2$, based on the original data. Panel (b) shows the RVI on main and interaction effects. We see that median income is the most relevant predictor in main effects, whereas the spatial variables (longitude and latitude) are the two most relevant predictors in interaction effects. Panel (c) and (d) show the estimated marginal effects for income and occupancy, respectively. Not surprisingly, the estimated marginal effect of income increases linearly at low level, and stabilized at high level.

Based on the RVI on main and interaction effects, we decide to keep the income, occupancy, latitude and longitude in the main effects, and consider only the

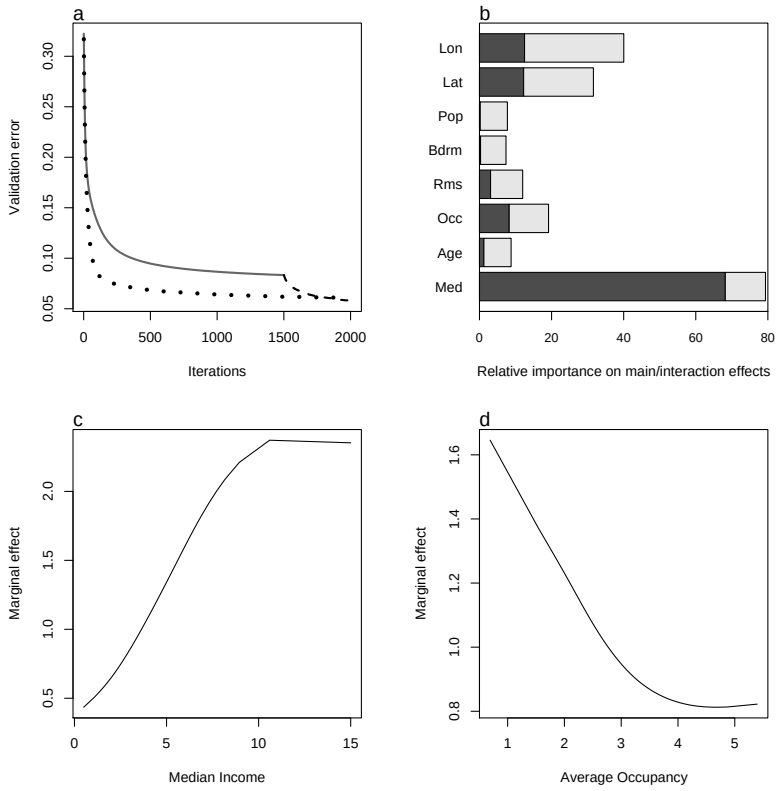

FIGURE 6. (a) Validation MSE for ARTSS (solid/dash corresponds to the first/second phase) and MART (dot). (b) The relative variable importance on main (black) and interaction (gray) effects. (c) and (d) Estimated main effects for median income and average occupancy.

interaction between longitude and latitude. However, house age is commonly considered as an important factor on house value. Thus, we enforce the house age into the main effects. Then we applied ARTSS on fewer variables (five in the first phase and two in the second phase). The new (sparse) model achieves $R^2 = 80\%$ on the validation set. Figure 7 shows the estimated contour plot on the joint values of longitude and latitude from the sparse model. It represents the effect of location after accounting for the effects of the other variables, i.e., an extra premium one pays for location. We see that the premium is larger near the Pacific coast especially in the Bay Area and Los Angeles-San Diego regions than the one in the northern, central valley, and south-eastern desert regions of California.

6.2. Spam Data. This data set, available at UCI Machine Learning Repository (http://www.ics.uci.edu/∼mlearn/ MLSummary.html), consists of information from 4,601 email messages, in a study to try to predict whether the email was junk email, or "spam". The response variable is binary, with values *email* (coded as -1) or *spam* (coded as 1), and there are 57 continuous predictors as follows.

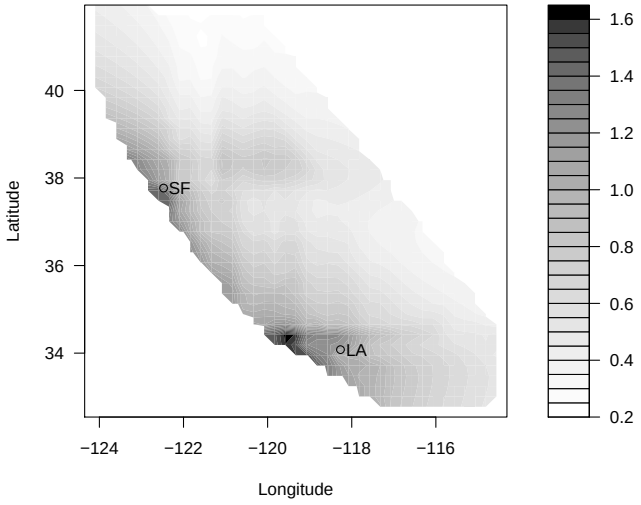

FIGURE 7. Estimated contour plot on joint values of longitude and latitude. SF: San Francisco; LA: Los Angeles.

- 48 quantitative predictors - percentage of words in the e-mail that match a given word.
- 6 quantitative predictors - percentage of words in the e-mail that match a given character.
- The average length of uninterrupted sequences of capital letters (denoted as average).
- The length of longest uninterrupted sequence of capital letters (longest) .
- The total number of capital letters in the e-mail (total).

A validation set of size 1,536 was randomly chosen, leaving 3,065 observations in the training set. Applying ARTSS to this data resulted in a validation error rate (0-1 loss) 5.0%, whereas MART achieved 5.7%. Figure 8(a) shows the validation errors, defined in (4.9) for ARTSS and MART. The second phase accounts 21.9% of total reduction in validation error, which, as expected, indicates the strong evidence of interaction effects. In the first phase of ARTSS, only 28 predictors were selected, out of where 12 have mRVI less than 1%. Figure 8(b) shows the iRVI and mRVI for some leading predictors. We see the imbalance of variable importance on main and interaction effects. Panel (c) and (d) show the marginal effects of log-odds of *spam* on two important predictors.

7. Acknowledgements

The authors would like to thank the referee for pointing out related work, and for constructive comments and suggestions that helped to improve the presentation of the paper.

References

[1] Breiman, L. (1999), "Using adaptive bagging to debias regression," Technical Report, Department of Statistics, University of California, Berkeley.

[2] Breiman, L. (2001), "Random forest," *Machine Learning*, 45, 5-32.

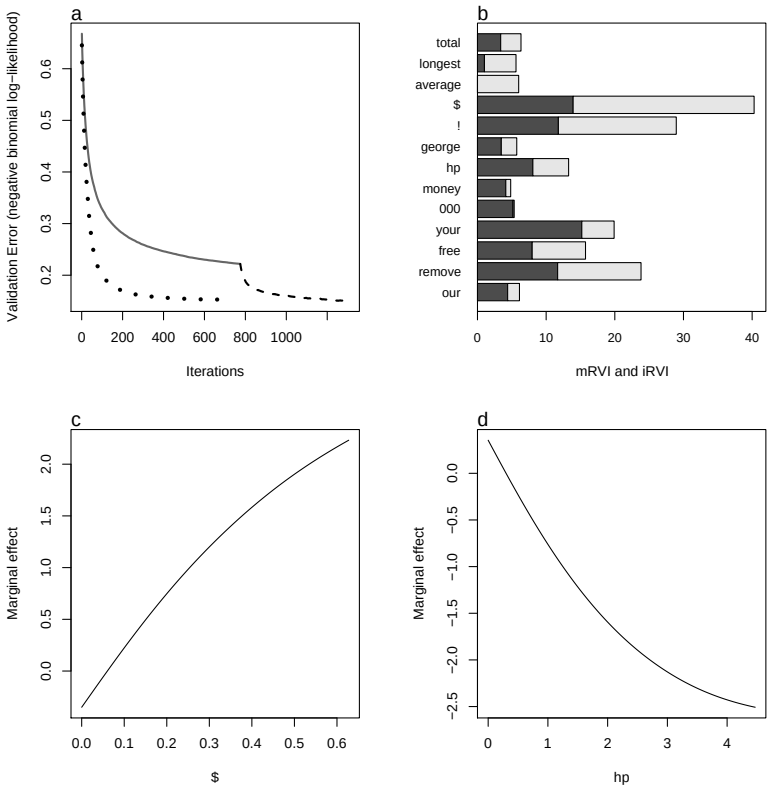

FIGURE 8. (a) Validation errors for ARTSS and MART. (b) The RVI on main (black) and interaction (gray) effects. (c) and (d) Marginal effects of two important predictors.

[3] Bühlmann, P. (2002), "Consistency for L_2Boosting and matching pursuit with trees and tree-type basis functions." Available at: http://citeseer.ist.psu.edu/611931.html

[4] Bühlmann, P., and Yu, B. (2003), "Boosting with the L2 loss: regression and classification," *Journal of the American Statistical Association*, 98, 324-340.

[5] Friedman, J. (1991), "Multivariate adaptive regression splines," *The Annals of Statistics*, 19, 1-141.

[6] Friedman, J. (2001), "Greedy function approximation: a gradient boosting machine," *The Annals of Statistics*, 29, 1189-1232.

[7] Friedman, J. (2002), "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, 38, 367-378.

[8] Friedman, J., Hastie, T., and Tibshirani, R. (2000), "Additive logistic regression: a statistical view of boosting" (with discussion), *The Annals of Statistics*, 28, 337-407.

[9] Friedman, J., and Popescu, B. E. (2005), "Predictive learning via rule ensembles," Stanford University, Dept. of Statistics. Available at: http://www-stat.stanford.edu/∼jhf/ftp/RuleFit.pdf

[10] Gu, C. (2002), *Smoothing Spline ANOVA Models*, New York: Springer.

[11] Hutchison, M. F., and de Hoog, F. R. (1985), "Smoothing noisy data with spline functions," *Numerische Mathematik*, 47, 99-106.

[12] John, G. H., Kohavi, R., and Pfleger, K. (1994), "Irrelevant features and the subset selection problem," in *Machine Learning: Proceedings of the Eleventh International Conference*, pp. 121-129.

[13] Li, B., and Goel, P. K. (2006), "Regularized optimization in statistical learning: a Bayesian perspective," *Statistica Sinica*, 16, 411-424.
[14] Pace, R. K., and Barry, R. (1997), "Sparse spatial autoregressions," *Statistics and Probability Letters*, 33, 291-297.
[15] Stone, C. (1985), "Additive regression and other nonparametric models," *The Annals of Statistics*, 13, 689-705.
[16] Wahba, G. (1990), *Spline Models for Observational Data*, SIAM.

Department of Statistics, The Ohio State University, Columbus, OH, 43210-1247, USA
Current address: Department of Experimental Statistics, Louisiana State University, Baton Rouge, LA, 70803-5606, USA
E-mail address: bli@lsu.edu

Department of Statistics, The Ohio State University, Columbus, OH, 43210-1247, USA
E-mail address: goel@stat.ohio-state.edu