Psychological Methods
2016, Vol. 21, No. 4, 583–602

1082-989X/16/$12.00 http://dx.doi.org/10.1037/met0000087

Finding Structure in Data Using Multivariate Tree Boosting

Patrick J. Miller
University of Notre Dame

Gitta H. Lubke
University of Notre Dame and VU University Amsterdam

Daniel B. McArtor and C. S. Bergeman
University of Notre Dame

Technology and collaboration enable dramatic increases in the size of psychological and psychiatric data collections, but finding structure in these large data sets with many collected variables is challenging. Decision tree ensembles such as random forests (Strobl, Malley, & Tutz, 2009) are a useful tool for finding structure, but are difficult to interpret with multiple outcome variables which are often of interest in psychology. To find and interpret structure in data sets with multiple outcomes and many predictors (possibly exceeding the sample size), we introduce a multivariate extension to a decision tree ensemble method called *gradient boosted regression trees* (Friedman, 2001). Our extension, *multivariate tree boosting*, is a method for nonparametric regression that is useful for identifying important predictors, detecting predictors with nonlinear effects and interactions without specification of such effects, and for identifying predictors that cause 2 or more outcome variables to covary. We provide the R package "*mvtboost*" to estimate, tune, and interpret the resulting model, which extends the implementation of univariate boosting in the R package "*gbm*" (Ridgeway, 2015) to continuous, multivariate outcomes. To illustrate the approach, we analyze predictors of psychological well-being (Ryff & Keyes, 1995). Simulations verify that our approach identifies predictors with nonlinear effects and achieves high prediction accuracy, exceeding or matching the performance of (penalized) multivariate multiple regression and multivariate decision trees over a wide range of conditions.

Keywords: boosting, multivariate, decision trees, nonparametric regression, model selection

Technology and collaboration enable dramatic increases in the size of psychological and psychiatric data collections, in terms of both the overall sample size and the number of variables collected. A major challenge is to develop and evaluate methods that can serve to leverage the information in these growing data sets to better understand human behavior. Big data can take many forms, such as audio and video recordings, web site logs, genetic sequences, and medical records (Chen, Chiang, & Storey, 2012; Howe et al., 2008; Manovich, 2012). In psychology and psychiatry, big data often take the form of large surveys with hundreds of individual questionnaire items comprised of many outcomes and predictors. In this context, however, it can be difficult to know what types of models to consider or even which variables to include in the model. As a result, it is often the case that many possible models need to be explored to find the model that most adequately captures the structure in the observed data. Although parametric models such as multivariate multiple regression and structural equation models (SEMs) can be used in this model selection, these methods make strong structural and distributional assumptions that limit exploration. To address this limitation, the current article describes *multivariate tree boosting,* a machine learning alternative to comparing different parametric models. It more easily allows discovery of important structural features in observed variables, such as nonlinear or interaction effects, and the detection of predictors that affect only some of the outcome variables.

Finding structure in observed data even in the absence of strong theory is particularly important for enhancing construct and external validity and for examining possible validity threats (Shadish, Cook, & Campbell, 2002). For instance, in the context of psychological testing, it is important to discover grouping variables (such as age or gender) that influence particular items in a test, indicating differential item functioning (Holland & Wainer, 1993). For observational studies, it is important to identify predictors with nonlinear effects or predictors that interact because presence of these effects makes the interpretation of main or linear effects misleading. In an experimental design, interactions between a

Patrick J. Miller, Department of Psychology, University of Notre Dame; Gitta H. Lubke, Department of Psychology, University of Notre Dame, and Department of Biological Psychology, VU University Amsterdam; Daniel B. McArtor and C. S. Bergeman, Department of Psychology, University of Notre Dame.

This research was based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant 1313583. Gitta H. Lubke is supported by NIDA R37 DA-018673. C. S. Bergeman is supported by a grant from the National Institute of Aging (1 R01 AG023571-A1-01). The computational work was done on clusters acquired through NSF MRI BCS-1229450.

We are grateful for the comments received from reviewers, Scott E. Maxwell, Ian Campbell, and Justin Luningham on earlier versions of this article.

Correspondence concerning this article should be addressed to Patrick J. Miller and Gitta H. Lubke, Department of Psychology, University of Notre Dame, 110 Haggar Hall, Notre Dame, IN 46656. E-mail: pmille13@nd.edu or glubke@nd.edu

treatment and other covariates can indicate limits of generalization (Shadish et al., 2002). More broadly, detecting and interpreting nonlinear and predictor specific effects can enhance the development of theory.

The usual approach for finding structure among many predictors and outcomes is to fit and compare a limited number of parametric models (Burnham & Anderson, 2002). These parametric models can involve latent variables (e.g., SEM, factor analysis) or only observed variables (e.g., canonical correlation analysis, multivariate multiple regression). However, there are significant problems with using parametric models for data exploration, including exploration in big data sets. First, in addition to distributional assumptions, models often make the strong assumption that a system of linear equations can sufficiently capture the important structure in the observed data. This ignores nonlinear effects and possible interactions, unless they are explicitly specified. Second, because the structure of the data is typically unknown, the number of models that must be included in a comparison for a thorough exploration can easily become unwieldy. Even if some nonlinear or interaction effects are included in the model, it can be difficult to specify all of the potentially relevant direct effects, nonlinear effects, and interactions in a parametric model a priori. In addition, it is impossible to estimate these effects simultaneously if the number of effects is larger than the sample size. Third, model selection is usually ad hoc rather than systematic and is not guaranteed to capture all of the important structural features. Even automatic procedures such as stepwise regression and best subsets analysis are known to be unable to identify a correct set of predictors of a given size and capitalize on sampling error (Hocking & Leslie, 1967; Thompson, 1995). Finally, conducting inference after this type of model selection inflates Type I error and leads to results that are difficult to replicate (Burnham & Anderson, 2002; Gelman & Loken, 2013).

An alternative to model selection using parametric models is to use a nonparametric approach such as decision trees (Breiman, Friedman, Stone, & Olshen, 1984). Decision trees are powerful because they can approximate nonlinear effects and interactions and handle missing data in the predictors. This is done without making parametric assumptions about the structure of the observed data. Decision trees are also easily interpretable in terms of decision rules, which are defined by the splitting variables. Each decision rule is in the form of a conditional effect (e.g., if $x_1 < c$ and $x_2 < d$ then . . .), which defines groups of observations with similar scores. Although decision trees are flexible and easy to interpret, the estimated structure varies considerably from sample to sample. Bagging (*b*ootstrap *agg*regation, Breiman, 1996) and random forests (Breiman, 2001), which includes bagging, improve on single decision trees by fitting many decision trees to bootstrap samples, forming an ensemble that is more robust against random sampling fluctuation (Strobl et al., 2009).

Decision trees and random forests have been successfully used in observational studies in psychology to identify predictors of mid- and later-life stress (Scott, Jackson, & Bergeman, 2011; Scott, Whitehead, Bergeman, & Pitzer, 2013), well-being in later life (Wallace, Bergeman, & Maxwell, 2002) and caregiver stability (Proctor et al., 2011). Decision trees have been used clinically to predict suicide attempts in psychiatric patients (Mann et al., 2008) and in experimental designs to analyze the effects of competence on depressive symptoms in children (Seroczynski, Cole, & Maxwell, 1997).

There are several ways to extend decision trees to multivariate outcomes, including many variants of multivariate decision trees (Brodley & Utgoff, 1995; Brown, Pittard, & Park, 1996; De'Ath, 2002; Dine, Larocque, & Bellavance, 2009; Franco-Arcega, Carrasco-Ochoa, Sánchez-Díaz, & Martínez-Trinidad, 2010; Hsiao & Shih, 2007; Struyf & Džeroski, 2006) and decision trees for longitudinal outcomes (Loh & Zheng, 2013; Segal, 1992; Sela & Simonoff, 2012). Decision tree methods can also be extended to multivariate outcomes by combining decision trees with parametric models, in which model parameters are allowed to differ in groups defined by split points on covariates (Zeileis, Hothorn, & Hornik, 2008). *Structural equation model trees* (SEM trees; Brandmaier, von Oertzen, McArdle, & Lindenberger, 2013) are an example of this approach, in which parameters of a structural equation model are allowed to differ in each partition. SEM trees can be used to address possible sources of measurement noninvariance, to detect group differences on factors, or to detect group differences in trajectories for longitudinal data (Brandmaier et al., 2013).

A natural extension to multivariate decision trees is a multivariate decision forest (Hothorn, Hornik, & Zeileis, 2006; Segal & Xiao, 2011). Like decision ensembles for single response variables, the multivariate random forest provides critical improvements to predictive performance compared to multivariate decision trees by combining predictions from many decision trees. Another example of a multivariate decision forest is a SEM forest with a fully saturated model for the covariance matrix, with the only difference being that the SEM forest uses a maximum likelihood criterion for split evaluation (Brandmaier et al., 2013). However, one of the limitations to ensembles of multivariate trees is that they are difficult to interpret. In general, decision tree ensembles exchange interpretability for prediction performance.

To address the limitations of exploratory analyses with parametric models and the difficulty of interpreting multivariate tree ensembles, we propose a new approach for exploratory data analysis with multivariate outcomes called *multivariate tree boosting*. Multivariate tree boosting is an extension of boosting for univariate outcomes (Bühlmann & Hothorn, 2007; Bühlmann & Yu, 2003; Freund & Schapire, 1997; Friedman, 2001, 2002). It fits univariate trees to multivariate outcomes by maximizing the covariance explained in the outcomes by predictors. Multivariate tree boosting addresses the problem of model selection with multivariate outcomes by smoothly approximating nonlinear effects and interactions by additive models of trees without requiring specification of these effects a priori. The method is suitable for truly big data scenarios in which the number of predictors is only limited by available memory and computation time. Its flexibility also makes it useful for exploratory analyses involving only a few variables. Our method differs from SEM trees or forests by not requiring a model to be specified for the outcome variables. It differs from multivariate decision forests (or saturated SEM forests) by allowing easier interpretation of nonlinear effects of predictors on individual outcome variables.

Our approach specifically addresses the issue of interpretation of multivariate tree ensembles because interpretation of the model is critical for applications in psychology and other sciences. We describe in detail how to use the model to select

important variables, visualize nonlinear effects, and detect departures from additivity. Multivariate tree boosting also allows estimation of the covariance explained in pairs of outcomes by predictors, a novel interpretation we think will be relevant for exploratory analyses in psychology. Our R package called "*mvtboost*" makes it easy to fit, tune, and interpret a multivariate tree boosting model and extends the implementation of univariate boosting in the R package "*gbm*" (Ridgeway, 2015) to multivariate outcomes.

Below, we introduce the approach by describing decision trees and univariate boosting, followed by introducing multivariate tree boosting. We demonstrate how to estimate, tune, and interpret the multivariate tree boosting model using functions in "*mvtboost.*" Specifically, we use multivariate tree boosting to identify predictors that contribute to specific aspects (subscales) of psychological well-being in aging adults. This example illustrates how multivariate tree boosting can be used to answer exploratory questions such as: Which predictors are important? What is the functional form of the effect of important predictors? How do predictors explain covariance in the outcomes? Finally, we use a simulation to evaluate the prediction error and predictor selection performance of our approach compared with other model-based and exploratory approaches.

Decision Trees and Ensembles

Decision Trees

Decision trees use a series of dichotomous splits on predictor variables to create groups of observations (nodes) that are maximally homogeneous with respect to the outcome variable. Nodes in a tree are created by finding both the predictor and the optimal split point on that predictor that result in maximally increased homogeneity in the daughter nodes. For continuous outcome variables, this means choosing predictors and split points that minimize the sums of squared errors within each daughter node. This process of finding a splitting variable and split point is then repeated within each daughter node, and is called recursive partitioning (Strobl et al., 2009).

There are several different ways to understand decision trees, which emphasize different properties. Decision trees are often represented as tree diagrams (Figure 1A), which show the set of variables and split points used to form the nodes. These diagrams represent the decision rules used to identify groups that are similar with respect to the outcome variable. Trees can also be viewed as models of conditional effects: The predictor with the largest main effect is selected for the first split, and each subsequent split is an effect conditional on all previously selected predictors. Thus, trees can capture interactions between predictors. From a geometric perspective, a decision tree is a piecewise function (Figure 1B). The decision rules or splits form regions or nodes (denoted R_j) in the predictor space, and the predicted values of a tree are the means within each of these regions. Algebraically, a tree can be represented by the piecewise function $T(\boldsymbol{X}, \theta) = \sum_{j=1}^{J} \gamma_j I(\boldsymbol{X} \in R_j)$ where θ contains the split points and predictors defining J regions R_j, and γ_j are the predictions in each region (Friedman, 2001). The indicator function, $I(\boldsymbol{X} \in R_j)$ denotes which observations in $\boldsymbol{X}$ fall into region R_j. Thus, the tree, $T(\boldsymbol{X}, \theta)$, is a piecewise approximation of

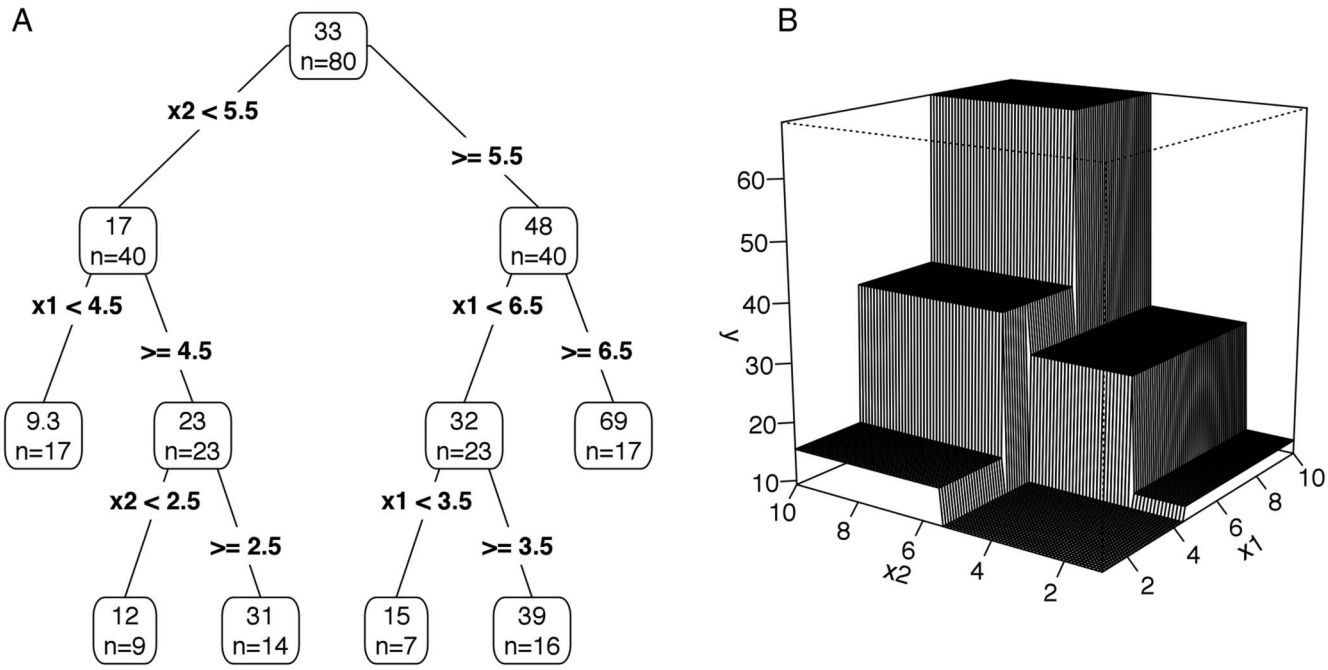

Figure 1. Representation of a decision tree as a tree diagram (Panel A) and as a surface in three dimensions for two predictors (Panel B). In the decision tree (Panel A) the means (top) and sample sizes (n = . . .) within each node are shown, and the split is shown in each branch. Panel B illustrates that decision trees are piecewise-constant approximations of nonlinear and interaction effects: each split divides the predictor space into rectangular regions, and the prediction of the tree is the mean of the response variable in each region. Plots following Hastie et al. (2009), Elith et al. (2008), and many others.

the unknown but potentially complex and nonlinear function $F(\boldsymbol{X})$ relating the outcome variable $\boldsymbol{y}$ to a set of predictors $\boldsymbol{X}$.

In addition to capturing interactions and approximating nonlinear functions, decision trees also handle missing values in the predictors using a procedure called "surrogate splitting." If there are missing values on the splitting variable, a second (or surrogate) variable is selected that best approximates the original split. Individuals with missing values are then classified according to the split on the surrogate rather than the original splitting variable.

One of the dangers of recursive partitioning is overfitting, which refers to a model fitting the idiosyncrasies of the sample in addition to the population structure. Overfitting is a result of fitting an overly complex model and results in high prediction error (Hastie et al., 2009). The prediction error is a function of the bias and variance of the tree, which are in turn a function of model complexity. Highly complex models have low bias but high variance, whereas low complexity models have high bias and low variance. For example, the most complex tree model is created by recursively partitioning the sample until only one observation remains in each node, achieving low bias. However, the structure of this tree changes drastically from sample to sample (it has high variance), which results in high prediction error. The complexity of a decision tree can be reduced by removing unnecessary splits after a full tree is fit (pruning) or by constraining the total number of splits of the tree (Hastie et al., 2009).

Decision Tree Ensembles

A better way to reduce the prediction error of individual trees is by using decision tree ensembles. Ensembles reduce the prediction error of individual trees by aggregating the predictions from many trees. There are two primary methods to create ensembles of decision trees: *bagging* (Breiman, 1996), as used in *random forests* (Breiman, 2001; Strobl et al., 2009), and *boosting* (Freund & Schapire, 1997; Friedman, 2001). In bagging, a set of trees is fit to bootstrap samples, and the prediction of the ensemble is the average prediction across all trees. This improves prediction error by increasing the stability of the model (decreasing variance) and decreasing the influence of extreme observations. In random forests, bagging is extended by randomly selecting the set of predictors evaluated for each split in the tree, making the trees less correlated. This has the effect of further reducing the variance of the ensemble compared with bagging. Both Breiman (2001) and Strobl, Malley, and Tutz (2009) describe random forests in more detail. We focus on a different approach for creating decision tree ensembles called boosting, which builds a tree ensemble by fitting trees that incrementally improve the predictions of the model.

Boosting

The intuition behind boosting is to iteratively create an ensemble of decision trees so that each subsequent tree focuses on observations poorly predicted by the previous trees. This is done by giving greater weight to observations that have been poorly predicted in previous trees and decreasing the weight of well predicted observations (Freund & Schapire, 1996, 1997). Subsequent trees in the model "boost" the performance of the overall model by selecting predictors and split points that better approximate the observations that are most poorly predicted. This procedure of iteratively fitting trees to the most poorly predicted observations was later shown to estimate an *additive model of decision trees by gradient descent* (Friedman, 2001, 2002; Friedman, Hastie, & Tibshirani, 2000). This additive model of decision trees is represented by:

$$\boldsymbol{y} = F(\boldsymbol{X}) = \sum_{m=1}^{M} T_m(\boldsymbol{X}, \theta_m)\nu \tag{1}$$

where the model for an outcome variable $\boldsymbol{y}$ is some unknown function of the predictors $F(\boldsymbol{X})$. The goal is to approximate $F(\boldsymbol{X})$ using an additive model of $m = 1, \ldots, M$ decision trees $T_m(\boldsymbol{X}, \theta_m)$ (Friedman et al., 2000). Each tree m has split points and splitting variables θ_m. Because model Equation 1 is a linear combination of decision trees, it retains their helpful properties: it can approximate complex, nonlinear functions $F(\boldsymbol{X})$, capture interactions among the predictors, and handle missing data.

The parameters that are estimated in Equation 1 are the splitting predictors and split points (θ_m) in each tree. The parameter ν is called the step-size and controls how quickly the model fits the observed data. The number of trees M, the depth of the trees, and the step-size ν are metaparameters that control the complexity of the model and are tuned to minimize prediction error (usually by cross-validation). In the following sections, we describe how this model is estimated, tuned, and interpreted.

Estimation by Gradient Descent

A critical problem with model Equation 1 is that the parameters, θ_m, cannot be estimated in all trees simultaneously (Friedman, 2001). This is because there is no closed formula or any procedure for estimating the best possible splitting variables and split points for a single tree except by an inexhaustible computational search (Hyafil & Rivest, 1976). Estimating parameters jointly in M trees is even more difficult. Because of this difficulty, the additive model of decision trees (Equation 1) is estimated by a stagewise approach called *gradient descent.* Stagewise procedures update the model one term at a time without updating the previous terms included in the model (Hastie et al., 2009). We illustrate gradient descent by first drawing a connection to multiple regression.

In multiple regression, the goal is to find the estimates of the regression weights β that minimize the sums of squared errors. This goal is equivalent to minimizing the squared error loss function:

$$\hat{\boldsymbol{\beta}} = \min_{\beta} L(\boldsymbol{y}, \boldsymbol{X}\boldsymbol{\beta}) = \min_{\beta} \sum_{i=1}^{N} (y_i - \boldsymbol{X}_i\boldsymbol{\beta})^2 \tag{2}$$

where the parameter $\boldsymbol{\beta}$ is a vector of regression weights, $\boldsymbol{y}$ is the dependent variable vector and $\boldsymbol{X}$ is a matrix of predictors for $i = 1, \ldots, N$ observations. The usual least-squares formulas that minimize the squared error loss function can be obtained by taking the first derivative (or *gradient*) of the loss function with respect to the parameters $\boldsymbol{\beta}$, setting it equal to zero, and solving for $\boldsymbol{\beta}$.

Estimating the additive model of decision trees (Equation 1) is done similarly by choosing the parameters θ_m so that the first derivative (or gradient) of the squared error loss function is minimized. In this case, the parameters are the splitting variables and split points of each tree in the model. Minimizing the loss function can be done stagewise by fitting each tree to the first derivative of

the loss function, or the *gradient* (Friedman, 2001). For continuous outcome variables with squared error loss, the gradient of the squared error loss function is the vector of residuals:

$$\frac{\partial}{\partial F(\boldsymbol{X})} L(\boldsymbol{y}, F(\boldsymbol{X})) = \frac{\partial}{\partial F(\boldsymbol{X})} \frac{1}{2} (\boldsymbol{y} - F(\boldsymbol{X}))^2 = \boldsymbol{y} - F(\boldsymbol{X}) \quad (3)$$

where $L(.)$ is the loss function, $\boldsymbol{y}$ is the vector of observations, and $F(\boldsymbol{X})$ is the unknown function mapping the predictors $\boldsymbol{X}$, to $\boldsymbol{y}$. The derivative of the squared error loss (times the constant ½) is taken with respect to $F(\boldsymbol{X})$ at the current step m, so that the loss function L is iteratively minimized by each tree (for details, see Friedman, 2001). Thus, estimation of the additive model of decision trees by gradient descent is equivalent to iteratively fitting decision trees to the residuals of the previous fit (Friedman et al., 2000; Friedman 2001). Intuitively, this corresponds to giving greater weight to the observations that are most poorly predicted. The estimation procedure can be summarized as:

Algorithm 1: Boosted Decision Trees for Continuous Outcomes Minimizing Squared Error Loss (Friedman et al., 2000; Friedman 2001)

For $m = 1, \ldots, M$ steps (trees) do:

1. Fit tree m to residuals
2. Update residuals by subtracting the predictions of tree m multiplied by step-size ν.

In the first step, the prediction of the model is the mean of the outcome variable, and the residuals are the deviations of the outcome around its mean.

In contrast to random forests in which trees are fully grown, individual trees are constrained to have a fixed number of splits. This is done to allow the user to directly control the degree of function approximation provided by each tree as well as the computational complexity. Empirical evidence suggests that a tree depth of five to 10 can capture the most important interactions (Friedman, 2001) while being relatively quick to estimate. Tuning the tree depth often improves performance. We recommend standardizing continuous predictors prior to estimation, which makes comparisons of the relative importance of these predictors to each other more interpretable (described in more detail later).

Resampling. Resampling is an important improvement to the boosting procedure in which each tree is fit to a subsample of the observations (that is, a sample of observations drawn without replacement) at each iteration of the algorithm. Friedman (2002) showed that incorporating this stochastic subsampling dramatically improves the performance of the algorithm. As with random forests, this improvement results from diminishing the impact of outlying observations (Friedman, 2002). However subsampling is an improvement over bootstrapping because it is valid under fewer regularity conditions (Politis & Romano, 1994). The fraction of the data used to fit each tree is called the *bag fraction* and is conventionally set at .5.

Tuning the model by choosing the number of trees and step-size. A critical part of successfully building this model by gradient descent is controlling the model complexity to achieve an optimal bias-variance tradeoff and low prediction error. The complexity of the model is a function of the number of trees (M), and the goal is to choose a minimally complex model that describes the data well. An overly complex model with many trees can fit the data too closely, resulting in high variance, whereas a model that is too simple will fail to approximate the underlying function well, resulting in high bias (Hastie et al., 2009).

There are two primary methods for choosing the best number of trees: splitting the sample into a training and test-set, or by k-fold cross-validation. In the first approach, the model is fit to the training set and then the number of trees is chosen to minimize prediction error on the test-set. But in this approach, the user needs to choose the fraction of the sample used for training and testing. It is often unclear how much of the sample should be used for each task. The second approach, k-fold cross-validation, provides a better estimate of the prediction error while still using the entire sample (Hastie et al., 2009). In k-fold cross-validation, the sample is divided into k groups (called *folds*, usually 5 or 10). The model is trained on $(k - 1)$ of the groups, and the prediction error is computed for the kth group. This is done for all k groups so that each observation is in the test set once. The cross-validation error is the average prediction error over all k groups at each step (or tree). The number of trees is then selected that minimizes the cross-validation error. Once the number of trees is chosen, the model is then retrained on the entire sample with the selected number of trees.

The step-size ν is a metaparameter set between 0 and 1 that indirectly affects the number of trees. It is sometimes called *shrinkage* because it shrinks the residuals at each iteration, and is sometimes called the *learning rate* because it affects how quickly the model approximates the observed data. Smaller step-sizes (e.g., .0001, .0005, or less) require many trees, but may provide better fit to the observed data (Friedman, 2001; Hastie et al., 2009). A larger step-size (e.g., .1, .5, or larger) requires fewer trees and fits the data more quickly, but can also more rapidly overfit. A typical strategy for choosing the step-size is to fix it to a small value (e.g., .001, .005, or .01) and then choose the optimal number of trees (Hastie et al., 2009). Note that the step size is a constant that cannot be factored out algebraically from Equation 1 because trees are fit sequentially, with each tree conditional on the previous trees. Because the step size affects the residual that serves as the outcome when fitting the next tree, changing the step size results in different boosting models. The step-size is fixed because computing an optimal step length based on the second derivative of the loss function (Equation 3) with respect to the parameters of tree m is very difficult.

Implementation. The procedure of fitting an ensemble of decision trees by gradient descent is referred to as "gradient boosted trees" (Elith, Leathwick, & Hastie, 2008) or "gradient boosting machines" (Friedman, 2001). For univariate outcomes, the model can be estimated and tuned using the R package "*gbm*" (Ridgeway, 2015), which is an open source version of Friedman's proprietary implementation MART and TreeNet procedures available through Salford Systems. Another open-source implementation is available in Python (Pedregosa et al., 2011).

Although outside the scope of this article, ensembles of decision trees can also be fit to binomial, Poisson, multinomial, and censored outcomes by choosing an appropriate loss function (Friedman, 2001; Ridgeway, 1999, 2015). The general procedure of fitting generalized additive models by gradient descent is known as "boosting" which has a rich literature with many developments

and extensions (see, e.g., Bühlmann & Hothorn, 2007; Bühlmann & Yu, 2003, 2006; Freund & Schapire, 1996, 1997; Friedman, 2001, 2002; Friedman et al., 2000; Groll & Tutz, 2011, 2012; Ridgeway, 1999). The R package "*mboost*" (Hothorn, Bühlmann, Kneib, Schmid, & Hofner, 2015) implements boosting algorithms to estimate a wide variety of high dimensional, generalized additive models by specifying different base procedures (e.g., splines and trees) and loss functions. For interested readers, a hands-on tutorial using "*mboost*" is also available (Hofner, Mayr, Robinzonov, & Schmid, 2014).

Multivariate Tree Boosting

One goal in learning about structure with multiple outcome variables is to understand which predictors explain correlations between outcome variables. This is an important question for psychologists when multiple items are used to measure unobserved latent constructs. In factor analysis and structural equation models, the covariance between outcome variables results from a dependence on unobserved latent variables. However, in a big data context when a potentially a large number of predictors have been measured, it may be the case that the covariance between outcomes results from a dependence on some of the measured predictors. Examining the associations between predictors and multivariate outcomes can reveal, for example, whether predictors have similar or unique effects across the different aspects of a construct. It may also provide a different way of understanding a construct in terms of other observed or latent variables and thus provide a basis for subsequently building large confirmatory SEMs.

We propose an extension of boosting to multivariate outcomes that maximizes the covariance explained between pairs of outcomes by predictors. This is done by maximizing a criterion called the covariance discrepancy, denoted by D, at each gradient descent step. Maximizing this criterion directly corresponds to selecting predictors that explain covariance in the outcomes. To motivate this criterion, note that a single gradient descent step with squared error loss corresponds to replacing an outcome, $y^{(q)}$, with its residual at each step (Algorithm 1). In the simplest case with one dichotomous predictor and no shrinkage, the gradient descent step removes the effect of that predictor from $y^{(q)}$. If the predictor has an effect on multiple outcomes (e.g., $y^{(1,2,3)}$), the covariance between these outcomes and $y^{(q)}$ will decrease after the gradient descent step. Thus, if a predictor causes multiple outcomes to covary, there will be a discrepancy between the sample covariance matrices before and after each gradient descent step.

Formally, the covariance discrepancy D is given by:

$$D_{m,q} = \|\hat{\Sigma}_{(m-1)} - \hat{\Sigma}_{(m,q)}\| \quad (4)$$

which is the discrepancy between sample covariance matrix of the outcomes at the previous step, $\hat{\Sigma}_{(m-1)}$, and the sample covariance matrix at step m, $\hat{\Sigma}_{(m,q)}$, after fitting a tree to outcome q. The discrepancy $D_{m,q}$ quantifies the amount of covariance explained in all outcomes by the predictor(s) selected by the tree fit to $y^{(q)}$ in step m. At the first step, $\hat{\Sigma}_{(0)} = S$, the sample covariance matrix. D corresponds to the improvement in how closely the model fits the sample covariance matrix at each step. There are many possible norms for Equation 4. We employ the L^2 norm, which is simply the sums of squared differences between all elements of the two covariance matrices. Maximizing D can be incorporated into the original boosting algorithm for squared error loss, giving Algorithm 2:

Algorithm 2: Multivariate Tree Boosting with Covariance Discrepancy Loss

For m in 1, . . ., M steps (trees) do:

1. For q in 1, . . ., Q outcome variables do:
   a. Fit tree $m^{(q)}$ to residuals, and compute the amount of covariance discrepancy $D_{m,q}$ (4)
2. Choose the outcome q^* corresponding to the tree that produced the maximum covariance discrepancy $D_{m,q}$ (4)
3. Update residuals by subtracting the predictions of the tree fit to outcome q^*, multiplied by step-size.

In the first step ($m = 1$), the predictions of the model are the means of the outcome variables, and the residuals are the deviations of the outcome variables from their means. As before, the decision trees are estimated for each outcome variable by minimizing squared error loss. At each gradient descent step m, one tree is chosen whose selected predictors maximize the covariance discrepancy $D_{m,q}$ (Equation 4). Equivalently, the tree is chosen that maximally explains covariance in the outcome variables, or maximally improves the model implied covariance matrix. The resulting model is an ensemble of trees where the selected predictors explain covariance in the outcomes. As noted previously, we recommend standardizing continuous predictors and outcomes for interpretability and numerical stability.

We have developed an R package (R Core Team, 2015) called "*mvtboost*" which efficiently maximizes the covariance discrepancy (Equation 4) indirectly by explaining variance in each outcome. Our work directly extends the implementation of univariate boosted decision trees in the R package "*gbm*" (Ridgeway, 2015) such that fitting an ensemble of decision trees to a single outcome variable corresponds to using the original *gbm* function directly. Both "*gbm*" and "*mvtboost*" are freely available on CRAN (https://cran.r-project.org).

In the following sections, we further describe methods to tune and interpret the multivariate tree boosting model using the "*mvtboost*" package in a step-by-step tutorial. For the tutorial, we use real data on the factors that predict aspects of psychological well-being in aging adults. We describe the well-being data and the research context in more detail below.

Application to Psychological Well-Being

Psychological Well-Being

Identifying the factors that impact well-being in aging adults is an important step to understanding successful aging and decreasing the risk for pathological aging (Wallace et al., 2002). Previous research has identified that high resilience, coping strategies, social support from family and friends, good physical health, and the

Table 1
Scale Sample Statistics, Missing Rates

Variable	*M*	*SD*	Range	α	% Missing	Scale
Psychological well-being						Psychological Well-Being (Ryff & Keyes, 1995)
Autonomy	2.9	.34	1.5–4	.82		
Environmental mastery	2.9	.41	1.5–4	.89		
Personal growth	3.0	.36	1.1–4	.87		
Positive relationships	2.9	.43	1.1–4	.89		
Purpose in life	2.9	.42	1.4–4	.90		
Self-acceptance	2.8	.47	1–4	.92		
Health						Measurement of Physical Health Scale (Belloc, Breslow, & Hochstim, 1971)
*Chronic health	1.7	1.4	0–7	.84	22%	
*Somatic health	3.0	2.0	1–11	.84	38%	
Self-report health	−.02	4.3	−7.9–12	.84	.84%	
Depression						CES-D (Devins & Orme, 1985)
Positive affect	7.4	3.2	4–16	.67	3.0%	
Negative affect	31	10	20–78	.86	1.6%	
Perceived social control	36	4.4	12–48	.79	1.2%	Desired Control Measure (Reid & Ziegler, 1981)
Control internal states	51	6.6	21–71	.91	2.5%	Perceived Control of Internal States Scale (Pallant, 2000)
Dispositional resilience						Hardiness (Bartone, Ursano, Wright, & Ingraham, 1989)
Commitment	49	5.8	17–60	.77	2.0%	
Control	48	4.9	21–59	.67	2.4%	
Challenge	40	4.5	24–54	.53	2.5%	
Ego resilience	42	6.3	14–56	.84	1.4%	Trait Ego Resilience (Block & Kremen, 1996)
Social support						Perceived Social Support from Friends and Family Scale (Procidano & Heller, 1983)
Friends	53	7.1	23–76	.94	2.9%	
Family	58	12	20–80	.96	1.5%	
Stress						Perceived Stress Scale (Cohen, Kamarck, & Mermelstein, 1983)
Problems	17	3.4	7–28	.85	1.7%	
Emotions	15	3.4	8–28	.85	1.7%	
Loneliness	39	11	20–77	.91	4.8%	UCLA Loneliness Scale (Russel, Peplau, & Cutrona, 1980)

Note. Predictors noted by * were not measured in the youngest cohort included in the analysis.

lack of stress and depression are important to successful aging (Wallace et al., 2002). In our exploratory analysis, we included these predictors as well as several additional ones—control of internal states, trait-ego resilience, and hardiness—and investigated the extent to which these predictors influenced particular aspects of well-being. Most research has focused on a well-being aggregate score, and little is known about whether the influence of these predictors varies across the different subscales of well-being.

A sample of 985 participants from the Notre Dame Study of Health and Well-Being (Bergeman & Deboeck, 2014) filled out the surveys that were used in this analysis. The data were cross-sectional, and the age of participants ranged from 19–91 with median age 55 and with 50% of the participants falling between the ages of 43–65. More of the participants were female (58%) than male (42%).

The Psychological Well-Being Scale (Ryff & Keyes, 1995) has six subscales: autonomy, environmental mastery, personal growth, positive relationships with others, purpose in life, and self-acceptance. These were used as dependent variables in the analysis. Gender, age, income, and education were included as demographic predictors. The primary predictors of interest were chronic, somatic, and self-reported health, depression (separated into positive and negative indicators), perceived social control, control of internal states, subscales of dispositional resilience (commitment, control, and challenge), ego resilience, social support (separately for friends and family), self-reported stress (problems, emotions), and loneliness. Scale summary statistics, reliability, and missingness rates are included in Table 1, and the correlations among the well-being subscales are shown in Table 2. Each subscale is continuous and approximately normally distributed. In total, 20 pre-

Table 2
Correlations Among Psychological Well-Being Subscales

	Autonomy	Environmental mastery	Personal growth	Positive relationships	Purpose in life	Self-acceptance
Autonomy	1					
Environmental mastery	.52	1				
Personal growth	.46	.57	1			
Positive relationships	.39	.65	.61	1		
Purpose in life	.51	.81	.71	.69	1	
Self-acceptance	.54	.82	.61	.67	.86	1

dictors were included in the analysis. All continuous predictors and the dependent variables were standardized.

Well-being items were 0.3%–1% missing, with 82.6% of the participants having measurements on all items. One participant who was missing on 95% of the well-being items was removed from the analysis. Well-being subscales were created by averaging scores over the items ignoring missingness. The predictors had low missingness rates (1%–5%), except for chronic and somatic health problems, which were not measured on the youngest cohort included in the study. Overall missingness rates were similar across gender, education, and income levels.

A modified version of the data set adding additional noise to all variables has been provided in the package *mvtboost* to illustrate use of the software while protecting privacy. The results reported here are from the original data and will differ slightly from the results obtained from the data provided in the package. The original data are available upon request.

Fitting the Model Using mvtboost

The *mvtboost* package can be installed and loaded directly in an interactive R session. The well-being data set described above is included in the package and can also be loaded into the workspace using the "data" command. These steps are shown below:

```
install.packages("mvtboost")
library(mvtboost)
data(wellbeing)
```

To fit the model, we first assign the dependent variables to the matrix "Y" and the predictors to the matrix "X" using their respective column indices. After standardizing the continuous outcomes and predictors ("Ys," "Xs," respectively) the multivariate tree boosting model can then be fit using the function `mvtb`:

```
res <- mvtb(Y=Ys,X=Xs)
```

Documentation for the function is available from the command `?mvtb`, which describes its use in greater detail.

As with the univariate procedure, the number of trees can be chosen to minimize a test or cross-validation estimate of the prediction error. For multiple outcomes, a useful criterion is the multivariate mean-squared error:

$$MSE = \frac{1}{nQ}\sum_{i=1}^{n}\left(\boldsymbol{Y}_i - \hat{\boldsymbol{Y}}_i\right)^2 \tag{5}$$

where $\boldsymbol{Y}_i$ is the vector of observations for each individual $i = 1, \ldots, n$ not used in training the model obtained for Q outcome variables, and $\hat{\boldsymbol{Y}}_i$ are the predicted values from the multivariate additive model of decision trees for these individuals.

The default number of trees and shrinkage values for the function mvtb are 100 and .01, respectively. These defaults are chosen to provide a quick initial fit, and further tuning is most likely necessary. For the well-being data we set the shrinkage to .005 and the maximum number of iterations to 10K. The best number of trees (2,482) was chosen by five-fold cross-validation and can be obtained using the summary function:

```
res5 <- mvtb(Y=Ys,X=Xs,n.trees=10000,
  shrinkage=.005, cv.folds=5)
summary(res)
```

Interpreting the Multivariate Additive Model of Decision Trees

One of the challenges of using multivariate decision tree ensembles is that the model is more difficult to interpret than a single tree. Although tree boosting can be used to build a very accurate predictive model, it is potentially more important for researchers to interpret the effects of predictors. Below, we describe approaches that have been developed to (a) identify predictors with effects on individual outcome variables, (b) identify groups of predictors that jointly influence one or more outcome variables, (c) visualize the functional form of the effect of important predictors, and (d) detect predictors with possible interactions or nonlinear effects.

Predictor selection by relative influence. The first goal in interpretation is to identify which predictors influence which outcome variables. The influence (or variable importance) of each predictor from the tree ensemble has been defined as the reduction in sums of squared error due to any split on that predictor, summed over all trees in the model (Friedman, 2001). Predictors can then be ranked by their influence or their *relative* influence, which is expressed as a percent of the total reductions in error attributed to all predictors. Predictors with large relative influence contribute more to the model than predictors with small influence. For the case of multivariate outcomes, the univariate influence is obtained for each predictor for each of the outcome variables. Summing the importance over all outcomes creates a global importance for the predictor across outcomes. To decide whether to retain a variable for further modeling, we suggest simply ranking the predictors using the influence score and retaining a practical number of predictors higher than a given rank for any or all outcome variables. In analyses of real data it is often clear which predictors should be considered for further modeling and may be a theoretical rather than an empirical decision.

To illustrate, consider the well-being data. The relative or raw influences of the predictors for each outcome variable can be computed using `summary` or the function `mvtb.ri`:

```
mvtb.ri(res5)
```

The results are shown in Figure 2. We see that control of internal states affects all aspects of psychological well-being except positive relationships with others. Like control of internal states, perceived stress-problems affects three aspects of well-being: self acceptance, purpose in life, and environmental mastery. Personal growth is driven by control of internal states and ego-resilience. Other patterns in the influences can be interpreted similarly and conform to theoretical expectations (e.g., Ryff & Keyes, 1995; Wallace et al., 2002).

There are some potential issues with selecting variables based solely on the relative influence, because variable selection in trees is biased (Strobl, Boulesteix, Zeileis, & Hothorn, 2007). This bias occurs because predictors with large variances or many categories will be selected more frequently than predictors with smaller variances or fewer categories even if the effect sizes are equal. This is because predictors with larger variances or more categories have a larger number of possible splits and can fit more readily to idiosyncrasies in the sample. There are various approaches to correct this in random forests, and the most common is to record the difference in accuracy before and after permuting a predictor (Strobl et al., 2007). Important predictors will show large discrep-

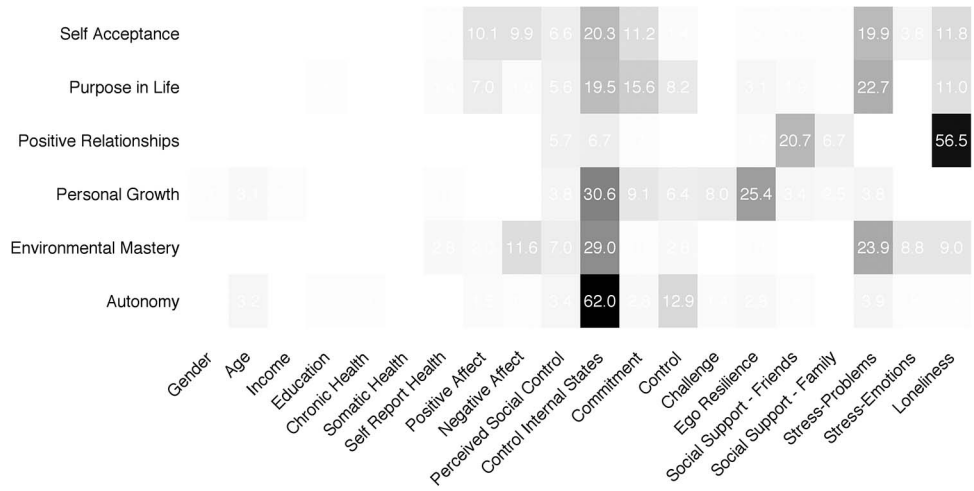

Figure 2. Relative influences from multivariate tree boosting. The relative influences are the sum of squared reductions in error attributable to splits on that predictor. They are reported as a percent of the total reduction in sums of squares for each well-being subscale. Control of internal states is important across well-being subscales. Control of internal states also has the single biggest contribution of any predictor, with a large effect for autonomy. Loneliness, stress problems, ego resilience, and social support from friends are also important to aspects of well-being.

ancies. Like with random forests, this permutation-based procedure is available for boosted tree ensembles as well. In addition to permutation, we suggest standardizing continuous predictors to ensure that they get selected with equal priority. Standardization can also improve the interpretability of the final model by ensuring the relative influences are on the same scale.

The second issue is the case when all the predictors have zero effects in the population. In this case, predictors will still be selected into the model and will report nonzero relative influence. As before, predictors with many categories or large variances may also be arbitrarily selected more frequently. Using the permutation procedure above can mitigate this, but the problem can be avoided altogether by assessing the fit of the model before variable selection. If the model explains little or no variance in the outcomes, there is no reason to use the model for variable selection. For the well-being data, we compute the R^2 for each dependent variable below. To do this, we obtain the predicted values of the model using the R function predict:

```
yhat <- predict(res5,newdata=Xs)
r2 <- diag(var(yhat)/var(Ys))
```

Computing the variance explained on a test-set of $n = 200$ observations, the results are as follows: autonomy (22%), environmental mastery (70%), personal growth (42%), positive relationships with others (51%), purpose in life (57%), and self acceptance (50%). Other measures of model fit for multivariate outcomes can be considered as well (e.g., η^2). The model explains substantial variance in all outcomes, further substantiating our interpretation of the relative influence scores.

Grouping predictors and outcomes by covariance explained. In addition to selecting predictors for inclusion into a subsequent multivariate model (e.g., a multivariate regression model or SEM), it may also be informative to select the outcome variables that are associated with the set of predictors. One criterion for selecting outcome variables is to choose the outcome variables whose covariance can be explained by a function of a common set of predictors. This approach, for example, could be used to (a) identify a set of demographic predictors that similarly affect particular symptoms of a disorder, or (b) indicate to what extent covariance in subscales of a construct is due to effects of predictors.

The covariance explained in the outcomes by a predictor can be estimated directly by Algorithm 2. At each gradient descent step, we record the covariance discrepancy (Equation 4) *without* taking the norm (resulting in a matrix) and the predictor X_j with the largest influence. Summing the raw discrepancy over all trees with each predictor approximates the *covariance explained* by each predictor. A *covariance-explained matrix* can then be organized as a $Q(Q + 1)/2 \times p$ table, where each element is the covariance explained in any pair of outcomes by predictor X_j, $j = 1, \ldots, p$. When the outcomes are standardized to unit variance, each element can be interpreted as the correlation explained in any pair of outcomes by predictor X_j. This decomposition is similar to decomposing R^2 in multiple regression. When the trees of the ensemble are limited to a single split and the predictors are independent, this decomposition is exact, otherwise it is approximate. The covariance-explained matrix can be used to identify groups of predictors that explain similar patterns of covariance in the outcomes. The covariance explained can be interpreted directly, or can be informative for building larger SEMs.

For the well-being data, the covariance explained matrix is obtained directly from the fitted model:

```
mvtb.covex(res5)
```

Figure 3 shows how the predictors explain correlation in pairs of subscales. We see that negative affect and stress problems have widespread effects on well-being. Control of internal states explains correlations across all dimensions and is the primary ex-

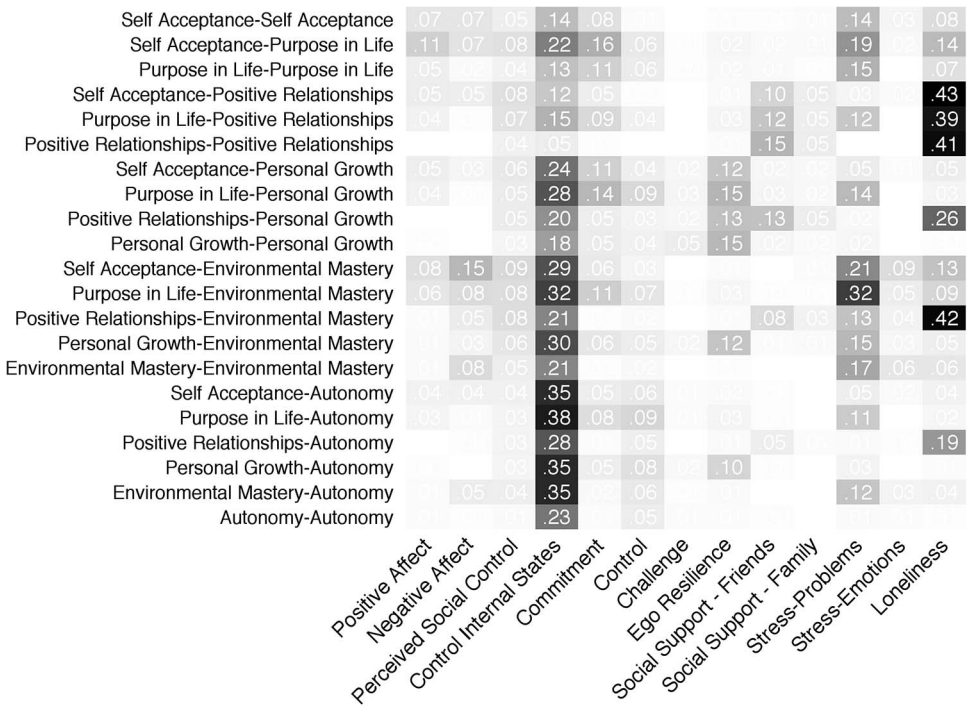

Figure 3. Correlation explained in pairs of subscales (rows) by a subset of the predictors (columns). Predictors with very small or zero effects included chronic/somatic health, income, education, gender, and age have been removed. Control of internal states explains correlation across almost all dimensions and is the primary explanatory predictor for autonomy. Stress problems primarily affects purpose in life and environmental mastery. Loneliness mainly affects positive relationships and the correlation between positive relationships and other factors. Ego-resilience mainly affects personal growth, while social support from friends is associated with positive relationships.

planatory predictor for autonomy. Similarly, stress, which can be detrimental to well-being, most strongly affects purpose in life and environmental mastery. Unsurprisingly, loneliness and social support from friends primarily affect positive relationships with others. Ego resilience mainly affects personal growth.

Clustering the covariance explained matrix. For a small number of outcomes or predictors, interpreting the covariance explained matrix is straightforward (e.g., Figure 3). However, when the number of predictors or outcomes becomes large, patterns become less obvious. It can be helpful to group predictors that explain similar patterns of covariance together using clustering procedures. The *covariance explained* matrix can be clustered by first computing the distance between columns (predictors) and the rows (pairs of outcomes), respectively. Predictors that explain similar patterns of covariance in the outcomes will be closer together (have smaller distance), as will pairs of outcomes that are functions of a similar set of predictors. The resulting distance matrices computed for the rows and columns can then be used to group rows or columns by hierarchical clustering (Johnson, 1967). This corresponds to grouping the predictors that explain covariance in similar pairs of outcomes and grouping pairs of outcomes dependent on similar sets of predictors.

Clustering the covariance explained matrix can be done via the function `mvtb.cluster`. This function allows different distance metrics to be used (e.g., Euclidean, Manhattan) and different ways to cluster the distance matrices. Heatmaps or network diagrams may be useful visual aids for further interpretation. A heatmap in which the rows and columns are clustered can be obtained using the function `mvtb.heat`. The commands to cluster the covariance explained matrix for the well-being data are shown below.

```
mvtb.cluster(res5)
mvtb.heat(res5)
```

Different clustering procedures can produce alternative arrangements of the predictors and outcomes, which may suggest novel interpretations of effects. We found that grouping the rows (pairs of outcomes) without clustering produced the most interpretable solution for well-being (see Figure 3) because the effects of several of the predictors concerned a single outcome variable. Examples of different clustering solutions for the covariance explained matrix are available in a package vignette. Additionally, we note that with many predictors and outcomes, it may also be helpful to cluster the matrix of relative influences. This can also be done using `mvtb.cluster` and `mvtb.heat`.

Visualizing nonlinear effects. Another important method for interpreting the additive model of trees is to visualize predictors with nonlinear effects or interactions. These plots effectively complement interpretations of relative influence by showing the direction and functional form of the effect of the predictor. Identifying nonlinear effects with plots can also help prevent model misspecification if a parametric model is the final goal.

Nonlinear effects can be inspected visually by plotting the fitted values of the model against individual predictors in a *partial dependence plot* (Friedman, 2001; Friedman & Meulman, 2003). In a partial dependence plot, the fitted values of the model are obtained by allowing one predictor to vary, while averaging over (or integrating out) the effects of the rest of the predictors. This plot can be extended to show the model implied effects of two variables jointly using a similar procedure. In this case, a three-dimensional perspective plot of the fitted values of the function is obtained. The fitted values are plotted jointly over a grid of the two predictors and are obtained by averaging over the other predictors. As others have noted, these plots do not perfectly represent the effects of individual predictors, but they are still useful for interpretation (Elith et al., 2008; Friedman & Meulman, 2003).

Univariate and multivariate plots can be easily obtained from the *mvtboost* package using the base R function `plot`. For the well-being data, we plot the effect of control of internal states on personal growth (Figure 4A). From the plot we see that above-average control of internal states corresponds to larger personal growth.

```
plot(res5,predictor.no=11,response.no=3)
```

Similarly `mvtb.perspec` can be used to produce a perspective plot involving two predictors. The following code produces Figure 4B, which shows the nonadditive effect of control of internal states and perceived stress problems on self-acceptance.

```
mvtb.perspec(res5,predictor.no=c(11,18),
  response.no=6)
```

Detecting nonadditive effects and possible interactions. Although decision trees are models of interactions, it is difficult to detect and interpret interaction effects from a decision tree ensemble. To address this issue, we again analyze the fitted values of the model. Following Elith, Leathwick, and Hastie (2008), possible two-way interactions can be detected by checking whether the fitted values of the approximation as a function of any pair of predictors deviates from a linear combination of the two predictors. Such departures indicate that the joint effect of the predictors is not additive and indicate a nonlinear effect or a possible interaction. A check of departures from additivity can be accomplished by computing the fitted values for any pair of predictors, over a grid of all possible levels for the two variables. For continuous predictors, 100 sample values are taken. The fitted values are then regressed onto the grid. Large residuals from this model indicate the fitted values are not a linear combination of the predictors, demonstrating nonlinearity or a possible interaction. For computational simplicity with many predictors, this might be done only for pairs of important variables.

Computing the departures from additivity from the multivariate tree boosting model can be accomplished using the `mvtb.nonlin` function:

```
mvtb.nonlin(res5,Y=Ys,X=Xs)
```

This produces a large table showing the departures from additivity involving all pairs of predictors (available in a *mvtboost* vignette). This table can be further interpreted by plotting pairs of predictors that produced the largest departures from

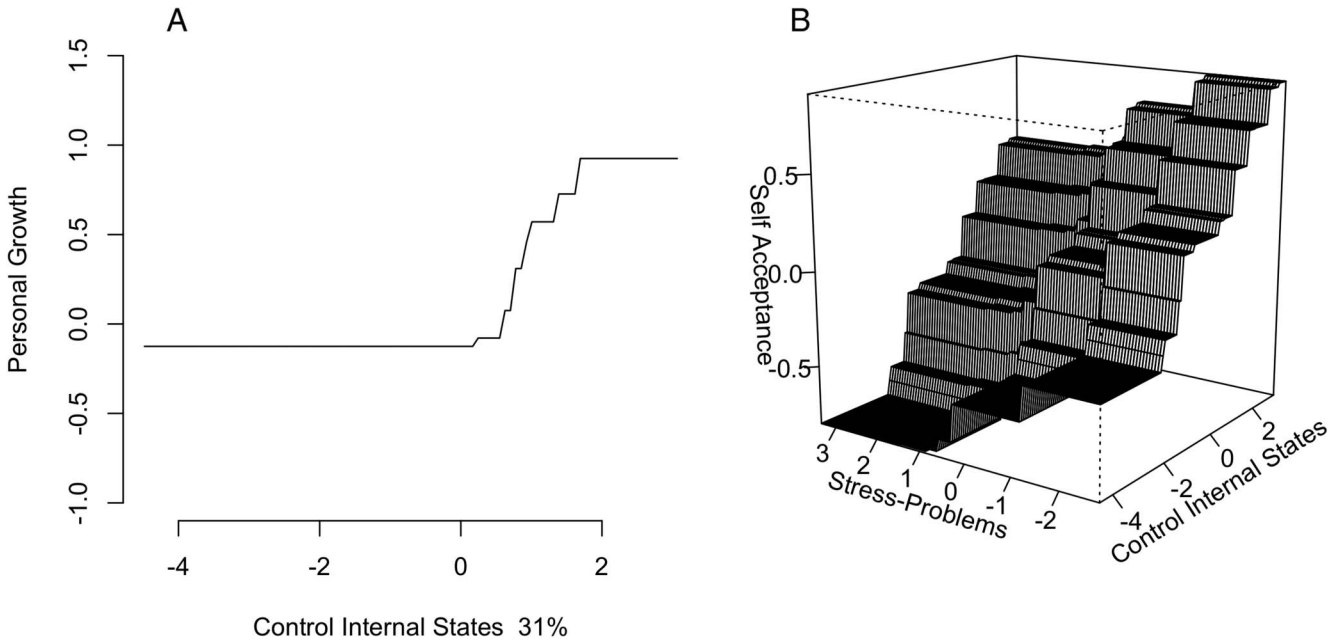

Figure 4. Model implied effects of control of internal states. Because the predictors and outcome were standardized, unit changes on the *x*, *y*, and *z* axes correspond to standard deviation changes in the predictors and outcomes. Panel A shows the predicted values of personal growth as a function of control of internal states (with % relative influence). Control of internal states shows a strong nonlinear effect—above average control is associated with larger personal growth. Panel B shows the model predicted values for self acceptance as a function of stress problems and control of internal states, indicating a possible multiplicative effect rather than simply an additive one.

additivity. In the well-being example, control of internal states and stress-problems produced a high ranking departure from additivity for the dependent variable self-acceptance. This is plotted in Figure 4B. We note that this approach is primarily a heuristic for interpreting the model. A variable with a nonadditive effect (e.g., a nonlinear effect like control of internal states) can produce bivariate departures from additivity which are not necessarily interactions.

Other Multivariate Boosting Approaches

To conclude the description of multivariate tree boosting, we describe here other approaches for boosting with multivariate outcomes. For example, it is possible to use boosting to estimate a high dimensional multivariate multiple regression model by updating one component of the regression weight matrix at a time (Hothorn, Bühlmann, Kneib, Schmid, & Hofner, 2010; Lutz & Bühlmann, 2006; Obozinski, Taskar, & Jordan, 2006). It is also possible to use boosting to estimate generalized additive mixed effect models (Groll & Tutz, 2012) and nonlinear time-series models (Robinzonov, Tutz, & Hothorn, 2012; Shafik & Tutz, 2009). These parametric approaches update the models one component at a time and can use splines to transform the predictors to capture nonlinear effects (Hastie & Tibshirani, 1990; Wood, 2006).

Other approaches for multivariate boosting consider the problem of classification with multiple related tasks, or *multitask* learning. Instead of viewing multiple classification tasks as separate problems, these algorithms seek to exploit commonalities between classification tasks to improve prediction performance (Faddoul, Chidlovskii, Torre, & Gilleron, 2010). Boosting algorithms in this setting can be used for web search ranking (Chapelle et al., 2010), facial recognition from images and videos (Wang, Zhang, & Zhang, 2009), and classification of documents or e-mail (Faddoul et al., 2010). A C++ software package called "*multiboost*" implements popular approaches (Benbouzid, Busa-Fekete, Casagrande, Collin, & Kégl, 2012). Our approach of estimating a model of decision trees for multivariate outcomes offers more flexibility than the parametric approaches and is more suitable for exploration with continuous, multivariate outcomes compared to these multitask approaches.

Variable Selection and Prediction Performance of Multivariate Tree Boosting

We have shown how to estimate, tune, and interpret the model using the R package "*mvtboost*" using the well-being data as an example. In this section we show how well the algorithm performs in comparison with other methods for effect and sample sizes that are common in psychology using simulated data. Additionally, we demonstrate the performance of the algorithm in a more traditional big data context in which the number of predictors exceeds the sample size.

The performance of the algorithm is quantified in terms of variable selection performance and prediction error. The performance of multivariate tree boosting is compared with model-based and exploratory approaches that are often used for data exploration. The model-based approaches are MANOVA and the multivariate Lasso. The exploratory approaches are multivariate classification and regression trees (De'Ath, 2002), as well as a bagged ensemble of these trees. The model-based approaches are expected to perform optimally when the model is correct, but to perform poorly when the model is specified incorrectly (i.e., in the presence of nonlinear effects). In these scenarios, multivariate tree boosting and (bagged) multivariate *classification and regression trees* (CART) are expected to perform better. Below we briefly review the methods used in the simulation to select variables and build predictive models with multiple outcomes when the number of predictors is larger than the sample size.

Approaches to Identifying Important Predictors With Multiple Outcomes

MANOVA. Multivariate analysis of variance (MANOVA) tests for mean differences in the outcomes due to predictors (e.g., Bray & Maxwell, 1985). This is done by specifying that the multivariate response $\boldsymbol{Y}$ is multivariate normally distributed with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Test statistics can be formed such as Wilk's Λ, which is a ratio of determinants of the within and total sums of squares and cross product matrices. For one predictor, the distribution of Λ is known and can be used to test whether the mean vectors are significantly different between the levels of the predictor. In high dimensional settings with many predictors, predictors can be tested one at a time. This approach has been recommended for genetic association studies with multiple outcomes in statistical genetics (Ferreira & Purcell, 2009). For high dimensional contexts, canonical correlation analysis (CCA) applied one predictor at a time is equivalent to this approach (van der Sluis, Posthuma, & Dolan, 2013).

Multivariate Lasso. The Lasso (Tibshirani, 1996) can be used to address the problem of estimating $\boldsymbol{\beta}$ in the multiple regression model $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}$ when the number of predictors exceeds the sample size. It obtains estimates of $\boldsymbol{\beta}$ using least squares plus an additional penalty on the sum of the absolute sizes of the estimates $\hat{\beta}_j$, which serves to shrink some coefficients to zero. A larger penalty results in more coefficients being reduced to 0, which is useful for variable selection and reduces the variance of the estimator (Hastie et al., 2009). The trade-off, however, is that all resulting estimates are biased (Tibshirani, 1996).

The Lasso for a single outcome variable can be generalized to the case of multiple outcomes by penalizing rows of $\boldsymbol{B}$ in the multivariate linear model $\boldsymbol{Y} = \boldsymbol{XB} + \boldsymbol{E}$. As with the univariate Lasso, the stringency of the penalization is controlled by the metaparameter λ. An implementation for the multivariate Lasso is available in the R package *glmnet* (Friedman, Hastie, & Tibshirani, 2010). Although the multivariate Lasso evaluates all predictors jointly even if the number of predictors exceeds the sample size, it still assumes that each outcome variable depends linearly on the predictors. To relax this assumption, spline transformations of the predictors and product terms can be added into the model to account for nonlinear effects and interactions, respectively. But because multivariate tree boosting does not require specification of these effects a priori, we do not include these terms in the comparison. That is, we compare the methods on the basis of identical a priori knowledge.

Multivariate CART. Multivariate CART (De'Ath, 2002) is a comparable exploratory procedure to multivariate tree boosting.[1] A multivariate decision tree is fit in a similar fashion to a univariate decision tree. However, instead of selecting predictors that minimize the univariate sums of squared errors, predictors are selected that minimize the sums of squared errors about the *multivariate* mean vector (De'Ath, 2002). The predictions of the multivariate tree are simply the mean of each outcome variable within each node. Because of the benefits of ensembles, a bagged version of multivariate CART was also employed, where multivariate trees were fit to 1,000 bootstrap samples. Predictions from the ensemble were averaged over all trees. Splits in multivariate trees were pruned by 10-fold cross-validation. As noted previously, other methods such as saturated SEM Trees (Brandmaier et al., 2013), or multivariate conditional inference trees (Hothorn et al., 2006) are also relevant comparisons. It should be noted, however, that only this implementation was both computationally feasible for big data sets and admitted an easy to compute measure of influence.

Simulation Experiments: Variable Selection and Prediction Error

Two simulations were carried out to quantify the performance of multivariate tree boosting for predictor selection and prediction error relative to these comparable methods. In each experiment, data were generated under a model linear in the predictors and three models that were not linear in the predictors. For each scenario, multivariate tree boosting was compared with MANOVA testing one predictor at a time, the multivariate Lasso, and (bagged) multivariate CART. The methods were compared in terms of their variable selection performance as well as their multivariate prediction error.

Metaparameter selection. For the multivariate Lasso, a value for the penalty parameter λ (which controls the amount of penalization) was chosen using 10-fold cross-validation. For (bagged) multivariate CART, splits in each tree were only considered if they improved the fit of the tree by a fixed amount. This amount was considered as a metaparameter and set to (.001, .0025, .005, .0075, .01, .015, .02). Trees were then pruned by 10-fold cross-validation. For multivariate tree boosting, the maximum number of trees was fixed to 20,000 and five different step-size values were used (.1, .01, .005, .001, .0005). The tree depth was set to either 1 or 3, resulting in $5 \times 2 = 10$ conditions. Each tree was fit to a randomly selected half the sample. The best number of trees was selected by fivefold cross-validation.

Data generation. Linear data were generated under the multivariate multiple regression model:

$$\boldsymbol{Y} = \boldsymbol{XB} + \boldsymbol{E} \tag{6}$$

Each of $p = 50$ or $p = 2{,}000$ predictors in the matrix $\boldsymbol{X}$ were independent, standard normal variables, and each error vector in the matrix $\boldsymbol{E}$ was distributed standard normal. The number of outcome variables in the matrix $\boldsymbol{Y}$ was five. For the case where $p = 50$, the matrix of regression weights $\boldsymbol{B}_{(50\times5)}$ was sparse: 15 rows each had two nonzero elements, so that each of these 15 predictors caused two outcomes to covary. When $p = 2{,}000$, the matrix of regression weights was generated similarly with 100 significant predictors. The pattern of nonzero coefficients in $\boldsymbol{B}$ was allowed to vary randomly across replications. The values of the coefficients were chosen to control the item-wise R^2. The sample size N was fixed at 1,000 and all observations were independent; 100 data sets were generated in this fashion.

Nonlinear data were generated under the multivariate multiple regression model Equation 6 for $p = 50$ predictors, with quadratic, cubic, and exponential transformations of the predictors. Plots of the nonlinear functions are shown in Figure 5. As shown, only the exponential transformation can be well approximated by a linear model. As was the case in the linear model, the nonzero coefficients in $\boldsymbol{B}$ were chosen randomly, and each predictor affected two randomly chosen outcomes. Instead of choosing the values of $\boldsymbol{B}$ to control the effect size, the error variance was chosen to control a given item-wise R^2.

Variable selection performance. For each method, the following statistics were used for variable selection:

- MANOVA: The p value from the F Test of Wilk's Λ for each predictor
- Multivariate Lasso: Penalized regression coefficients
- Multivariate CART: relative influence (the reduction in multivariate SSE attributed to splits on each predictor)
- Bagged Multivariate CART: Influence averaged over 1,000 trees
- Multivariate tree boosting: The influence summed over all outcome variables.

If the test statistic was larger than a cutoff τ (smaller for MANOVA) the variable was selected. Comparing this indicator with the known true predictors produces a 2×2 table with true positives, false positives, true negatives, and false negatives for a particular cutoff value (see Table 3). The *true positive rate* is the ratio of true predictors that were identified by the model over the total number of true predictors. The *false positive rate* is the number of incorrectly selected predictors over the total number of predictors with truly zero effects. Ideally, a method will have a true positive rate close to 1 as well as a false positive rate close to 0. Most empirical analyses are less ideal and choosing the cutoff τ becomes important: A liberal cutoff will lead to a high true positive rate, but also a higher false positive rate. A conservative cutoff will lead to a low false positive rate, but also a lower true positive rate.

Measuring variable selection performance using the area under the ROC curve. To summarize the variable selection performance independent of the threshold chosen, an ROC curve can be used (Hanley & McNeil, 1982). The curve is created by computing the true positive rate and false positive rates resulting from allowing the cutoff to take all realized values of the statistic. The true positive rates are then plotted against the false positive rates. If a procedure selects variables according to chance, the curve will be a line along the diagonal of the plot, with the true positive rate increasing along with the false positive rate. If the procedure selects variables perfectly, the true positive rate will be 1 (all true predictors selected) whereas the false positive rate is zero (no true zero predictors selected). The ROC curve can be summarized as a single number by

[1] The implementation of multivariate CART described in (De'Ath, 2002) was not available from CRAN at the time of publication. The archived version 1.6-2 was compiled and used for the following simulations.

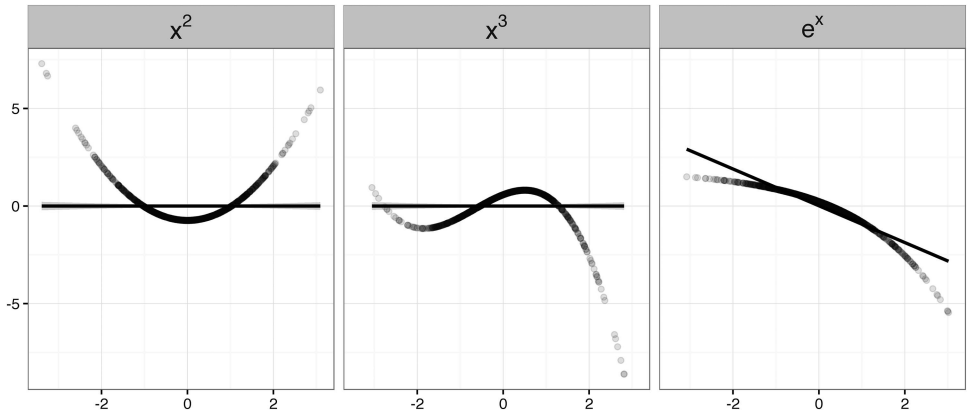

Figure 5. Simulated nonlinear effects for a single predictor: x^2, x^3, e^x for 1,000 samples. Only e^x is well approximated by a linear model.

computing the area under the curve (AUC; Bradley, 1997). An AUC of .5 corresponds to chance variable selection performance, and an AUC of 1 corresponds to perfect variable selection performance. Methods can then be compared based on their AUC values, which is an indicator of performance across all possible cutoffs (Bradley, 1997).

Prediction performance. Prediction performance was assessed under the same data generating models used to assess variable selection performance. For each of the 100 replications, a test set was generated from the model by drawing new errors from the same distribution. The sample size for this test set was n = 1,000. The design matrix $\boldsymbol{X}$ was the same for the test set as the original sample. The multivariate mean squared error (Equation 5) was computed for each method on the n new observations in the test set. The mean squared prediction error was computed directly for multivariate tree boosting, (bagged) multivariate CART, and the multivariate Lasso. For MANOVA, variables were first selected and then included in a multivariate multiple regression model, and the mean squared prediction error was computed from this model.

Table 3
Confusion Matrix for Variable Selection

		Effect of predictor	
		$\beta_j > 0$	$\beta_j = 0$
Selected by method	$\hat{I}_j > \tau$	TP	FP
	$\hat{I}_j < \tau$	FN	TN

Note. Columns: Predictor has a true effect in the population if β_j is greater than zero, or does not have a true effect if β_j is equal to zero. Rows: Predictor j is labeled as having an effect in the population if statistic $\hat{I}_j$ is greater than a cutoff τ, and no effect if $\hat{I}_j$ is less than τ. In the simulation, the statistic is the influence measure from the decision tree methods (boosting, multivariate CART, and bagged multivariate CART), p-value from MANOVA, or the estimated regression coefficient from the Lasso. For MANOVA, the rows of this matrix are reversed because variables are selected when $p < \tau$, where $\tau = \alpha$. TP = true positive; FP = false positive; FN = false negative; TN = true negative.

Results

The simulation results confirm theoretical expectations that if predictors have nonlinear effects multivariate tree boosting performs best relative to other methods. In these cases, the methods based on the linear model are incorrectly specified. When predictors have linear effects, multivariate tree boosting matches the performance of MANOVA and the Lasso (showing that little power is lost) and out performs multivariate CART.

Variable Selection Performance

The AUCs averaged over all 100 replications are shown in Figure 6. Higher AUC values indicate improved variable selection performance across all possible cutoffs. For data not well approximated by a linear model, multivariate tree boosting exceeded the performance of all methods (including bagged multivariate CART) for even very small effect sizes. When the true model was linear or could be easily approximated by a linear model, multivariate tree boosting performed as well as the linear model methods and better than multivariate CART or bagged multivariate CART. A similar pattern holds when selecting predictors with linear effects when p = 2,000 (see Figure 8). We note that simple multivariate CART performed very poorly in this case.

Prediction Performance

The mean-square prediction error is shown in Figure 7. When predictors had nonlinear effects, multivariate tree boosting had much lower prediction error than the Lasso or MANOVA. When the responses are truly linear functions of the predictors (or approximately linear), multivariate tree boosting performs just as well as both the Lasso and MANOVA, and better than (bagged) multivariate CART. A similar pattern holds with p = 2,000 predictors (see Figure 8), though multivariate tree boosting has much lower prediction error than multivariate CART and even bagged multivariate CART.

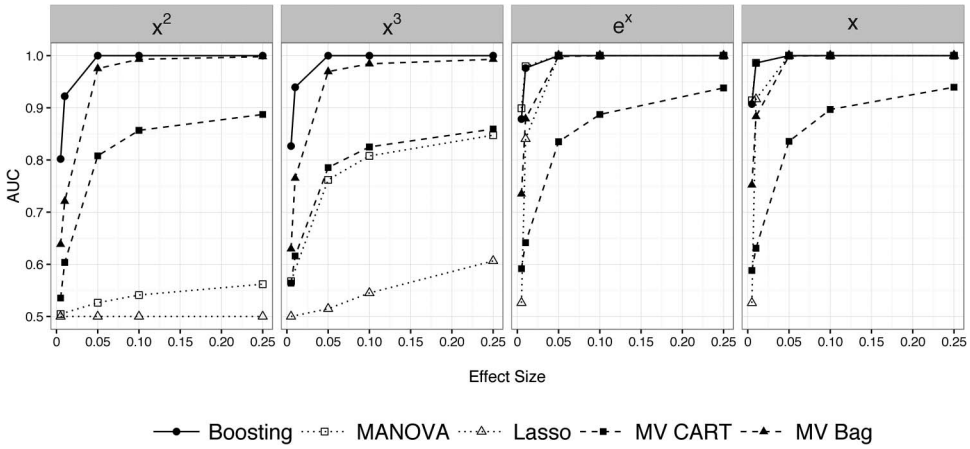

Figure 6. Variable selection performance of multivariate tree boosting compared to MANOVA, the Lasso, multivariate CART, and bagged multivariate CART. The performance (higher is better) is shown for a given nonlinear effect (e^x, x^2, x^3, x is the linear model) for a range of effect sizes. Higher AUCs indicate better performance: an AUC of 1 is perfect and an AUC of .5 corresponds to variable selection no better than chance. For effects that are not linear (x^2, x^3), boosting dominates all other methods followed by bagged multivariate CART. For e^x and x (linear effect), boosting does not perform significantly worse than MANOVA or the Lasso.

Discussion

Finding structure in large data collections with many predictors and outcomes is important because it can enhance content or external validity for experimental designs and provide a starting point for specifying complex parametric models in observational studies. Even with smaller data sets involving fewer variables, exploratory procedures can be helpful for detecting nonlinear effects and interactions, which help to correctly specify subsequent parametric models. Finding structure requires flexible statistical methods suitable for many observed variables and little theory. Although model selection using factor models, CCA, and multivariate multiple regression models can be useful, these models make strong structural assumptions and are unwieldy with a large number of predictors. It can also be difficult or impossible to estimate parameters in a large model if the number of parameters exceeds the sample size, or to specify all possible models in order to perform a systematic model search. It can also be difficult to specify all necessary transformations of predictors to capture potential nonlinear effects. Incomplete model searches likely result in ignoring predictors with important effects. Finding structure in any data set involves selecting important predictors, seeing how these predictors influence some or all outcome variables, and identifying predictors that possibly interact and have nonlinear effects. To accomplish all of these things simultaneously, a highly flexible and interpretable model building approach is necessary.

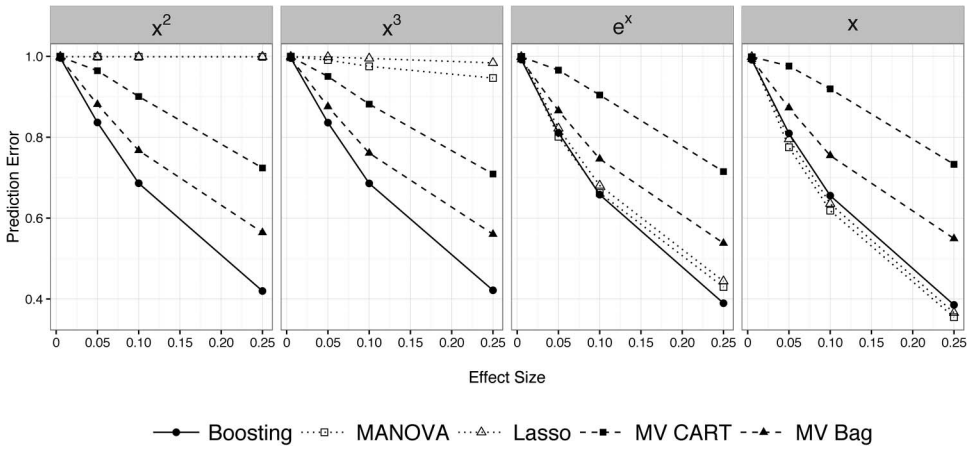

Figure 7. Prediction performance of multivariate tree boosting, MANOVA, the Lasso, multivariate CART, and bagged multivariate CART. Lower prediction error is better. The performance is shown for a given nonlinear effect (e^x, x^2, x^3, x is the linear model) for a range of effect sizes. Multivariate tree boosting has lower prediction error than bagged multivariate CART, MANOVA or the Lasso for conditions with nonlinear effects (x^2, x^3). It has comparable performance when effects are linear (x) or approximated well by a linear model (e^x).

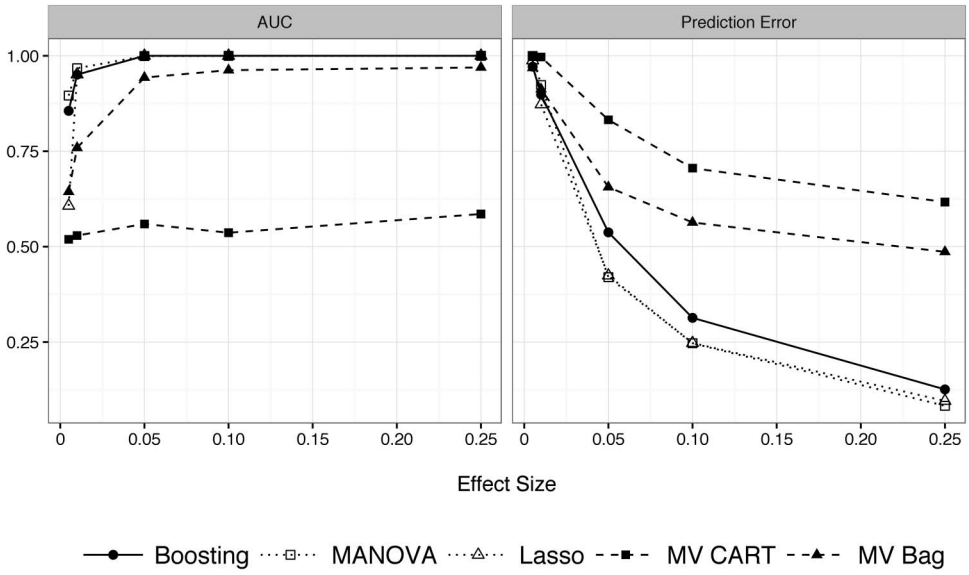

Figure 8. Prediction performance of multivariate tree boosting, MANOVA, the Lasso, multivariate CART, and bagged multivariate CART when $p > n$. The performance is shown when predictors have linear effects only. Multivariate tree boosting does not perform significantly worse than MANOVA or the Lasso, and performs much better than multivariate CART and bagged multivariate CART.

Multivariate tree boosting is one such approach. In this approach, the outcome variables are assumed to depend on an arbitrary function of the predictors, which is then approximated by an additive model of decision trees. Decision trees provide the necessary flexibility for approximating nonlinear effects and interactions without a priori specification, and handle missing data by surrogate splitting. Multivariate tree boosting complements existing multivariate decision tree procedures by providing methods to easily interpret the resulting ensemble. Predictors can be selected by ranking their importance in predicting each outcome, nonlinear or interaction effects between pairs of predictors can be detected by testing for departures from additivity, and plots can be used to visualize these effects. Finally, we showed how to identify predictors that explain covariance in the outcomes. All of these methods contribute to a better understanding of the structure between a set of outcome variables and a set of predictors. Our model can also be used as a "black box" for prediction, if prediction rather than an investigation of variable importance is the ultimate goal.

Our simulations verified that multivariate tree boosting has better predictor selection performance and lower prediction error than other model-based and exploratory procedures when predictors have nonlinear effects. The improved performance of multivariate tree boosting is dramatic even with relatively small effect sizes. Our simulations also show that when linear models are correctly specified, multivariate tree boosting performs nearly as well as methods that explicitly search for this type of effect.

Multivariate tree boosting can be used for exploratory analyses in lieu of model selection with linear models because it systematically and robustly explores existing structure in data. Both our simulations and the applied example of psychological well-being highlight the benefits of this structured exploration, which includes the discovery of predictors with nonlinear effects. Multivariate tree boosting is also unambiguously interpreted as exploratory—the final results are importance scores or plots, which are suggestive and not inferential.

Multivariate tree boosting complements SEM trees and other model-based recursive partitioning approaches by being maximally exploratory, highly predictive, easy to use, and still interpretable. SEM trees are best used for identifying ways to modify a known model: for example, by identifying groups with different trajectories, or groups that are not measurement invariant. SEM trees and forests also provide a "global" measurement of variable importance in terms of the log-likelihood discrepancy between model implied covariance matrices due to splits on a predictor (Brandmaier et al., 2013). However, SEM trees still make strong assumptions—all of the assumptions of SEM (even fully saturated models) must still hold within each region. In contrast, multivariate tree boosting suggests ways to build parametric models by discovering important structural features in the data while making few assumptions.

Several important practical questions remain: How large of a sample is necessary, and how many predictors can be included in the model? A sample size recommendation is difficult to make because it depends on the true model, the pattern and size of effects, and the number of variables (outcomes and predictors) under consideration. Our initial simulation results at $n = 1{,}000$, are a useful guideline. Specific sample size limits can be investigated further using Monte Carlo simulations. More generally, statistical learning-based analyses tend to need larger samples than parametric models because the data replace the information in the structural relations specified by the parametric model.

With respect to the number of predictors, our simulation results show that boosting performs well and is computationally feasible with a very large number of predictors (i.e., 2,000). Given a large enough sample and/or the presence of sufficiently large effects,

investigating even larger numbers of predictors is possible given enough computation time. We note that though the number of false positives is expected to increase with a larger number of predictors, our results show that the rate of false positives can be well controlled. We caution that the variables included in the model should be selected based on their potential theoretical relevance. For questionnaire data, either items or factor scores can be included depending on the research question.

An important limitation of multivariate tree boosting that is common to all decision tree ensembles is that the estimation of individual trees and the ensemble is not optimal. An optimal approach would require an exhaustive computational search of all possible splitting variables and split points, without conditioning on a previous split or on splits in a previous tree. This limitation means that individual trees can fail to capture complex dependency structures. Further research is necessary to understand the limits of the approximation provided by decision tree ensembles and whether these limits are of practical importance.

There are several other limitations concerning the methods we developed for model interpretation. First, as discussed in the text, the relative influence score in this implementation is biased in favor of predictors with large variances and many categories. However, this bias can be mitigated by using alternative importance measures based on permutations of the predictors. The performance of these alternatives in the context of boosting should still be explored. Second, the departure from additivity is a heuristic, not a statistic. Further research along the lines of Mentch and Hooker (2014) is necessary to understand how well it performs in detecting interactions from boosted tree ensembles. Third, the "covariance explained" in pairs of outcomes by predictors is only an exact decomposition for uncorrelated predictors and for decision trees with a single split. On the positive side, even these approximations can be helpful in understanding the structure in observed data. Fourth, partial dependence plots can hide heterogeneous predictor effects. Plots of independent conditional expectation may be more informative in some cases (Goldstein, Kapelner, Bleich, & Pitkin, 2015).

In addition to methodological limitations, there are shortcomings concerning the scope of the present work. For instance, our framework does not account for missingness in the outcomes. As a result, imputation by singular value decomposition, *k*-nearest neighbors (Troyanskaya et al., 2001), the Expectation-Maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977), or by data augmentation (Tanner & Wong, 1987) is necessary. These approaches are all likely reasonable to the extent that their assumptions about the data distribution and the missingness mechanism hold. Further research could provide a method of missing value imputation using the boosting model itself. Another limitation in scope is that our approach does not directly accommodate longitudinal data. Fitting a boosting model to factor scores in a latent growth model is one way to link an exploratory boosting model to a parametric growth model. Additionally, pilot studies have indicated that using principal components of the outcomes as dependent variables can result in better prediction of the outcomes than boosting them directly. Our ongoing work focuses on quantifying this improvement under a wide range of conditions. Finally, though several representative methods of variable selection were compared in the simulation, little is known about how multivariate tree boosting influence scores compare to other variable selection methods from other models and methods not included here (for instance, variable importance in CCA and multiple regression: Grömping, 2015; Huo & Budescu, 2009; Lambert, Wildt, & Durand, 1988; Nathans, Oswald, & Nimon, 2012; Nimon, Henson, & Gates, 2010; Thompson, 2005). In general, however, we expect variable selection performance to depend on the assumptions and statistical power of the model from which the influence scores are computed.

In summary, multivariate tree boosting is a useful approach for finding structure in data for large data sets in psychology. Its flexibility makes it a compelling tool to discover and clarify important theoretical relationships that would be otherwise difficult or impossible to detect by model selection with parametric models. We hope that this work will open future developments and improvements in exploratory analyses for big data in psychology.

References

Bartone, P. T., Ursano, R. J., Wright, K. M., & Ingraham, L. H. (1989). The impact of a military air disaster on the health of assistance workers. A prospective study. *Journal of Nervous and Mental Disease, 177,* 317–328. http://dx.doi.org/10.1097/00005053-198906000-00001

Belloc, N. B., Breslow, L., & Hochstim, J. R. (1971). Measurement of physical health in a general population survey. *American Journal of Epidemiology, 93,* 328–336.

Benbouzid, D., Busa-Fekete, R., Casagrande, N., Collin, F.-D., & Kégl, B. (2012). MultiBoost: A multi-purpose boosting package. *Journal of Machine Learning Research, 13,* 549–553.

Bergeman, C. S., & Deboeck, P. R. (2014). Trait stress resistance and dynamic stress dissipation on health and well-being: The reservoir model. *Research in Human Development, 11,* 108–125. http://dx.doi.org/10.1080/15427609.2014.906736

Block, J., & Kremen, A. M. (1996). IQ and ego-resiliency: Conceptual and empirical connections and separateness. *Journal of Personality and Social Psychology, 70,* 349–361. http://dx.doi.org/10.1037/0022-3514.70.2.349

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition, 30,* 1145–1159. http://dx.doi.org/10.1016/S0031-3203(96)00142-2

Brandmaier, A. M., von Oertzen, T., McArdle, J. J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods, 18,* 71–86. http://dx.doi.org/10.1037/a0030001

Bray, J. H., & Maxwell, S. E. (1985). *Multivariate analysis of variance* (Vol. *54*). Newbury Park, CA: Sage.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24,* 123–140. http://dx.doi.org/10.1007/BF00058655

Breiman, L. (2001). Random forests. *Machine Learning, 45,* 5–32. http://dx.doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. London, UK: CRC press.

Brodley, C. E., & Utgoff, P. E. (1995). Multivariate decision trees. *Machine Learning, 19,* 45–77. http://dx.doi.org/10.1007/BF00994660

Brown, D. E., Pittard, C. L., & Park, H. (1996). Classification trees with optimal multivariate decision nodes. *Pattern Recognition Letters, 17,* 699–703. http://dx.doi.org/10.1016/0167-8655(96)00033-5

Bühlmann, P., & Hothorn, T. (2007). Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science, 22,* 477–505. http://dx.doi.org/10.1214/07-STS242

Bühlmann, P., & Yu, B. (2003). Boosting with the L_2 loss: Regression and classification. *Journal of the American Statistical Association, 98,* 324–339. http://dx.doi.org/10.1198/016214503000125

Bühlmann, P., & Yu, B. (2006). Sparse boosting. *Journal of Machine Learning Research, 7,* 1001–1024.

Burnham, K. P., & Anderson, D. R. (2002). *Model selection and multimodel inference: A practical information-theoretic approach*. New York, NY: Springer Science & Business Media.

Chapelle, O., Shivaswamy, P., Vadrevu, S., Weinberger, K., Zhang, Y., & Tseng, B. (2010). Multi-task learning for boosting with application to web search ranking. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1189–1198). New York, NY: ACM.

Chen, H., Chiang, R. H., & Storey, V. C. (2012). Business Intelligence and Analytics: From Big Data to Big Impact. *Management Information Systems Quarterly, 36,* 1165–1188.

Cohen, S., Kamarck, T., & Mermelstein, R. (1983). A global measure of perceived stress. *Journal of Health and Social Behavior, 24,* 385–396. http://dx.doi.org/10.2307/2136404

De'Ath, G. (2002). Multivariate regression trees: A new technique for modeling species-environment relationships. *Ecology, 83,* 1105–1117.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the Expectation-Maximization (EM) algorithm. *Journal of the Royal Statistical Society Series B. Methodological, 39,* 1–38.

Devins, G. M., & Orme, C. M. (1985). Center for epidemiologic studies depression scale. *Test Critiques, 20,* 144–160.

Dine, A., Larocque, D., & Bellavance, F. (2009). Multivariate trees for mixed outcomes. *Computational Statistics & Data Analysis, 53,* 3795–3804. http://dx.doi.org/10.1016/j.csda.2009.04.003

Elith, J., Leathwick, J. R., & Hastie, T. (2008). A working guide to boosted regression trees. *Journal of Animal Ecology, 77,* 802–813. http://dx.doi.org/10.1111/j.1365-2656.2008.01390.x

Faddoul, J. B., Chidlovskii, B., Torre, F., & Gilleron, R. (2010, December). Boosting multi-task weak learners with applications to textual and social data. In *Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on* (pp. 367–372). Washington, DC: IEEE. http://dx.doi.org/10.1109/ICMLA.2010.61

Ferreira, M. A. R., & Purcell, S. M. (2009). A multivariate test of association. *Bioinformatics, 25,* 132–133. http://dx.doi.org/10.1093/bioinformatics/btn563

Franco-Arcega, A., Carrasco-Ochoa, J. A., Sánchez-Díaz, G., & Martínez-Trinidad, J. F. (2010). Multivariate decision trees using different splitting attribute subsets for large datasets. In A. Farzindar & V. Kešelj (Eds.), *Advances in artificial intelligence* (Vol. 6085, pp. 370–373). Berlin, Germany: Springer. http://dx.doi.org/10.1007/978-3-642-13059-5_49

Freund, Y., & Schapire, R. E. (1996). *Experiments with a new boosting algorithm* (Vol. *96,* pp. 148–156). Presented at the International Conference on Machine Learning Bari, Italy.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences, 55,* 119–139. http://dx.doi.org/10.1006/jcss.1997.1504

Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics, 29,* 1189–1232. http://dx.doi.org/10.1214/aos/1013203451

Friedman, J. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis, 38,* 367–378. http://dx.doi.org/10.1016/S0167-9473(01)00065-2

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting (With discussion and a rejoinder by the authors). *Annals of Statistics, 28,* 337–407. http://dx.doi.org/10.1214/aos/1016218223

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software, 33,* 1–22. http://dx.doi.org/10.18637/jss.v033.i01

Friedman, J. H., & Meulman, J. J. (2003). Multiple additive regression trees with application in epidemiology. *Statistics in Medicine, 22,* 1365–1381. http://dx.doi.org/10.1002/sim.1501

Gelman, A., & Loken, E. (2013). *The garden of forking paths: Why multiple comparisons can be a problem, even when there is no "fishing expedition" or "p-hacking" and the research hypothesis was posited ahead of time*. Unpublished manuscript.

Goldstein, A., Kapelner, A., Bleich, J., & Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics, 24,* 44–65. http://dx.doi.org/10.1080/10618600.2014.907095

Groll, A., & Tutz, G. (2011). *Variable selection for generalized additive mixed models by likelihood-based boosting*. Munich, Germany: Ludwig-Maximilians-University.

Groll, A., & Tutz, G. (2012). Regularization for generalized additive mixed models by likelihood-based boosting. *Methods of Information in Medicine, 51,* 168–177. http://dx.doi.org/10.3414/ME11-02-0021

Grömping, U. (2015). Variable importance in regression models. *Wiley Interdisciplinary Reviews: Computational Statistics, 7,* 137–152. http://dx.doi.org/10.1002/wics.1346

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology, 143,* 29–36. http://dx.doi.org/10.1148/radiology.143.1.7063747

Hastie, T., & Tibshirani, R. (1990). *Generalized additive models* (Vol. 43). London, UK: CRC Press.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., & Tibshirani, R. (2009). *The elements of statistical learning* (Vol. 2). New York, NY: Springer. http://dx.doi.org/10.1007/978-0-387-84858-7

Hocking, R. R., & Leslie, R. N. (1967). Selection of the best subset in regression analysis. *Technometrics, 9,* 531–540.

Hofner, B., Mayr, A., Robinzonov, N., & Schmid, M. (2014). Model-based boosting in R: A hands-on tutorial using the R package mboost. *Computational Statistics, 29,* 3–35. http://dx.doi.org/10.1007/s00180-012-0382-5

Holland, P. W., & Wainer, H. (Eds.). (1993). *Differential item functioning*. New York, NY: Routledge.

Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M., & Hofner, B. (2015). *mboost: Model-Based Boosting, R package version R package version 2.5–0*. Retrieved from http://CRAN.R-project.org/package=mboost

Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M., & Hofner, B. (2010). Model-based boosting 2.0. *Journal of Machine Learning Research, 99,* 2109–2113.

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics, 5,* 651–674.

Howe, D., Costanzo, M., Fey, P., Gojobori, T., Hannick, L., Hide, W., . . . Rhee, S. Y. (2008). Big data: The future of biocuration. *Nature, 455,* 47–50. http://dx.doi.org/10.1038/455047a

Hsiao, W. C., & Shih, Y. S. (2007). Splitting variable selection for multivariate regression trees. *Statistics & Probability Letters, 77,* 265–271. http://dx.doi.org/10.1016/j.spl.2006.08.014

Huo, Y., & Budescu, D. V. (2009). Erratum to "An extension of dominance analysis to canonical correlation analysis." *Multivariate Behavioral Research, 44,* 859. http://dx.doi.org/10.1080/00273170903467679

Hyafil, L., & Rivest, R. L. (1976). Constructing optimal binary decision trees is NP-complete. *Information Processing Letters, 5,* 15–17. http://dx.doi.org/10.1016/0020-0190(76)90095-8

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika, 32,* 241–254. http://dx.doi.org/10.1007/BF02289588

Lambert, Z. V., Wildt, A. R., & Durand, R. M. (1988). Redundancy analysis: An alternative to canonical correlation and multivariate multiple regression in exploring interset associations. *Psychological Bulletin, 104,* 282–289. http://dx.doi.org/10.1037/0033-2909.104.2.282

Loh, W. Y., & Zheng, W. (2013). Regression trees for longitudinal and multiresponse data. *The Annals of Applied Statistics, 7,* 495–522. http://dx.doi.org/10.1214/12-AOAS596

Lutz, R. W., & Bühlmann, P. (2006). Boosting for high-multivariate responses in high-dimensional linear regression. *Statistica Sinica, 16,* 471–494.

Mann, J. J., Ellis, S. P., Waternaux, C. M., Liu, X., Oquendo, M. A., Malone, K. M., . . . Currier, D. (2008). Classification trees distinguish suicide attempters in major psychiatric disorders: A model of clinical decision making. *The Journal of Clinical Psychiatry, 69,* 23–31. http://dx.doi.org/10.4088/JCP.v69n0104

Manovich, L. (2012). Trending: The promises and the challenges of big social data. In M. K. Gold (Ed.), *Debates in the digital humanities* (pp. 460–475). Minneapolis, MN: University of Minnesota Press. http://dx.doi.org/10.5749/minnesota/9780816677948.003.0047

Mentch, L., & Hooker, G. (2014). Ensemble trees and CLTs: Statistical inference for supervised learning. *arXiv:1404.6473*.

Nathans, L. L., Oswald, F. L., & Nimon, K. (2012). Interpreting multiple linear regression: A guidebook of variable importance. *Practical Assessment, Research & Evaluation, 17,* 1–19.

Nimon, K., Henson, R. K., & Gates, M. S. (2010). Revisiting interpretation of canonical correlation analysis: A tutorial and demonstration of canonical commonality analysis. *Multivariate Behavioral Research, 45,* 702–724. http://dx.doi.org/10.1080/00273171.2010.498293

Obozinski, G., Taskar, B., & Jordan, M. I. (2006). Multi-task feature selection. *Statistics Department, UC Berkeley, Tech. Rep.* Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.951&rep=rep1&type=pdf

Pallant, J. F. (2000). Development and validation of a scale to measure perceived control of internal states. *Journal of Personality Assessment, 75,* 308–337. http://dx.doi.org/10.1207/S15327752JPA7502_10

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12,* 2825–2830.

Politis, D. N., & Romano, J. P. (1994). Large sample confidence regions based on subsamples under minimal assumptions. *Annals of Statistics, 22,* 2031–2050. http://dx.doi.org/10.1214/aos/1176325770

Procidano, M. E., & Heller, K. (1983). Measures of perceived social support from friends and from family: Three validation studies. *American Journal of Community Psychology, 11,* 1–24. http://dx.doi.org/10.1007/BF00898416

Proctor, L. J., Van Dusen Randazzo, K., Litrownik, A. J., Newton, R. R., Davis, I. P., & Villodas, M. (2011). Factors associated with caregiver stability in permanent placements: A classification tree approach. *Child Abuse & Neglect: The International Journal, 35,* 425–436. http://dx.doi.org/10.1016/j.chiabu.2011.02.002

R Core Team. (2015). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.

Reid, D. W., & Ziegler, M. (1981). The desired control measure and adjustment among the elderly. In H. M. Lefcourt (Ed.), *Research with the locus of control construct: Vol. 1, Assessment methods* (pp. 127–159). New York, NY: Academic Press. http://dx.doi.org/10.1016/B978-0-12-443201-7.50008-7

Ridgeway, G. (1999). The state of boosting. *Computing Science and Statistics, 31,* 172–181.

Ridgeway, G. (2015). gbm: Generalized boosted regression models. [Computer software]. Retrieved from http://CRAN.R-project.org/package=gbm

Robinzonov, N., Tutz, G., & Hothorn, T. (2012). Boosting techniques for nonlinear time series models. *Advances in Statistical Analysis, 96,* 99–122. http://dx.doi.org/10.1007/s10182-011-0163-4

Russell, D., Peplau, L. A., & Cutrona, C. E. (1980). The revised UCLA Loneliness Scale: Concurrent and discriminant validity evidence. *Journal of Personality and Social Psychology, 39,* 472–480. http://dx.doi.org/10.1037/0022-3514.39.3.472

Ryff, C. D., & Keyes, C. L. M. (1995). The structure of psychological well-being revisited. *Journal of Personality and Social Psychology, 69,* 719–727. http://dx.doi.org/10.1037/0022-3514.69.4.719

Scott, S. B., Jackson, B. R., & Bergeman, C. S. (2011). What contributes to perceived stress in later life? A recursive partitioning approach. *Psychology and Aging, 26,* 830–843. http://dx.doi.org/10.1037/a0023180

Scott, S. B., Whitehead, B. R., Bergeman, C. S., & Pitzer, L. (2013). Combinations of stressors in midlife: Examining role and domain stressors using regression trees and random forests. *The Journals of Gerontology Series B, Psychological Sciences and Social Sciences, 68,* 464–475. http://dx.doi.org/10.1093/geronb/gbs166

Segal, M. R. (1992). Tree-structured methods for longitudinal data. *Journal of the American Statistical Association, 87,* 407–418. http://dx.doi.org/10.1080/01621459.1992.10475220

Segal, M., & Xiao, Y. (2011). Multivariate random forests. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1,* 80–87.

Sela, R. J., & Simonoff, J. S. (2012). RE-EM trees: A data mining approach for longitudinal and clustered data. *Machine Learning, 86,* 169–207. http://dx.doi.org/10.1007/s10994-011-5258-3

Seroczynski, A. D., Cole, D. A., & Maxwell, S. E. (1997). Cumulative and compensatory effects of competence and incompetence on depressive symptoms in children. *Journal of Abnormal Psychology, 106,* 586–597. http://dx.doi.org/10.1037/0021-843X.106.4.586

Shadish, W. R., Cook, T. D., & Campbell, D. T. (2002). *Experimental and quasi-experimental designs for generalized causal inference*. Belmont, CA: Wadsworth Cengage Learning.

Shafik, N., & Tutz, G. (2009). Boosting nonlinear additive autoregressive time series. *Computational Statistics & Data Analysis, 53,* 2453–2464. http://dx.doi.org/10.1016/j.csda.2008.12.006

Strobl, C., Boulesteix, A.-L., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics, 8,* 25. http://dx.doi.org/10.1186/1471-2105-8-25

Strobl, C., Malley, J., & Tutz, G. (2009). An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological Methods, 14,* 323–348. http://dx.doi.org/10.1037/a0016973

Struyf, J., & Džeroski, S. (2006). Constraint based induction of multi-objective regression trees. In F. Bonchi & J. F. Boulicaut (Eds.), *Knowledge discovery in inductive databases* (Vol. 3933, pp. 222–233). Berlin, Germany: Springer. http://dx.doi.org/10.1007/11733492_13

Tanner, M. A., & Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association, 82,* 528–540. http://dx.doi.org/10.1080/01621459.1987.10478458

Thompson, B. (1995). Stepwise regression and stepwise discriminant analysis need not apply. *Educational and Psychological Measurement, 55,* 524–534.

Thompson, B. (2005). Canonical correlation analysis. In B. S. Everitt & D. Howell (Eds.), *Encyclopedia of statistics in behavioral science*. Hoboken, NJ: Wiley, Ltd. http://dx.doi.org/10.1002/0470013192.bsa068

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B. Methodological, 58,* 267–288.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., . . . Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics, 17,* 520–525. http://dx.doi.org/10.1093/bioinformatics/17.6.520

van der Sluis, S., Posthuma, D., & Dolan, C. V. (2013). TATES: Efficient multivariate genotype-phenotype analysis for genome-wide association studies. *PLOS Genetics, 9,* e1003235. http://dx.doi.org/10.1371/journal.pgen.1003235

Wallace, K. A., Bergeman, C. S., & Maxwell, S. E. (2002). Predicting well-being outcomes in later life: An application of classification and regression tree (CART) analysis. In S. P. Shohov (Ed.), *Advances in psychology research* (Vol. 17, pp. 71–92). Hauppauge, NY: Nova Science Publishers.

Wang, X., Zhang, C., & Zhang, Z. (2009). Boosted multi-task learning for face verification with applications to web image and video search. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition* (pp. 142–149). Washington, DC: IEEE.

Wood, S. (2006). *Generalized additive models: An introduction with R*. London, UK: CRC press.

Zeileis, A., Hothorn, T., & Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics, 17,* 492–514. http://dx.doi.org/10.1198/106186008X319331

Received May 27, 2015
Revision received January 20, 2016
Accepted March 6, 2016 ■