Contrast trees and distribution boosting

Jerome H. Friedman[a,1]

[a]Department of Statistics, Stanford University, Stanford, CA 94305

Contributed by Jerome H. Friedman, June 8, 2020 (sent for review December 9, 2019; reviewed by Edward I. George and Mark R. Segal)

A method for decision tree induction is presented. Given a set of predictor variables $\mathbf{x} = (x_1, x_2, \cdots, x_p)$ and two outcome variables y and z associated with each $\mathbf{x}$, the goal is to identify those values of $\mathbf{x}$ for which the respective distributions of $y \mid \mathbf{x}$ and $z \mid \mathbf{x}$, or selected properties of those distributions such as means or quantiles, are most different. Contrast trees provide a lack-of-fit measure for statistical models of such statistics, or for the complete conditional distribution $p_y(y \mid \mathbf{x})$, as a function of $\mathbf{x}$. They are easily interpreted and can be used as diagnostic tools to reveal and then understand the inaccuracies of models produced by any learning method. A corresponding contrast-boosting strategy is described for remedying any uncovered errors, thereby producing potentially more accurate predictions. This leads to a distribution-boosting strategy for directly estimating the full conditional distribution of y at each $\mathbf{x}$ under no assumptions concerning its shape, form, or parametric representation.

machine learning | prediction diagnostics | boosting | quantile regression | conditional distribution estimation

In statistical (machine) learning one has a system under study with associated attributes or variables. The goal is to estimate the unknown value of one of the variables y, given the known joint values of other (predictor) variables $\mathbf{x} = (x_1 \cdots, x_p)$ associated with the system. It is seldom the case that a particular set of $\mathbf{x}$ values gives rise to a unique value for y. There are quantities other than those in $\mathbf{x}$ that influence y whose values are neither controlled nor observed. Specifying a particular set of joint values for $\mathbf{x}$ results in a probability distribution of possible y values, $p_y(y \mid \mathbf{x})$, induced by the varying values of the uncontrolled quantities. Given a sample $\{y_i, \mathbf{x}_i\}_{i=1}^N$ of previous solved cases, the goal is to estimate the distribution $p_y(y \mid \mathbf{x})$, or some of its properties, as a function of the predictor variables $\mathbf{x}$. These can then be used to predict likely values of y realized at each $\mathbf{x}$.

Usually only a single property of $p_y(y \mid \mathbf{x})$ is used for prediction, namely a measure of its central tendency such as the mean or median. This provides no information concerning prediction accuracy at each $\mathbf{x}$. Only average accuracy over a set of $\mathbf{x}$ values can be estimated using cross-validation. In order to estimate individual prediction accuracy at each $\mathbf{x}$ one needs additional properties of $p_y(y \mid \mathbf{x})$ such as various quantiles, or the distribution itself. These can be estimated as functions of $\mathbf{x}$ using maximum-likelihood or minimum-risk techniques. Such methods, however, do not provide a measure of accuracy (goodness of fit) for their respective estimates as functions of $\mathbf{x}$. There is no way to know how well the results actually characterize the distribution of y at each $\mathbf{x}$.

Contrast trees can be used to assess lack of fit of any estimate of $p_y(y \mid \mathbf{x})$, or its properties (mean or quantiles), as a function of $\mathbf{x}$. In cases where the fit is found to be lacking, contrast boosting applied to the output can often improve accuracy. A special case of contrast boosting, distribution boosting, can be used to estimate the full conditional distribution $p_y(y \mid \mathbf{x})$ under no assumptions. Contrast trees can also be used to uncover concept drift and reveal discrepancies in the predictions of different learning algorithms.

Building Contrast Trees

The data consist of N observations $\{\mathbf{x}_i, y_i, z_i\}_{i=1}^N$ each with a joint set of predictor variable values $\mathbf{x}_i$ and two outcome variables y_i and z_i. Contrast trees are constructed from this data in an iterative manner. At the Mth iteration the tree partitions the space of $\mathbf{x}$ values into M disjoint regions $\{R_m\}_{m=1}^M$ each containing a subset of the data $\{\mathbf{x}_i, y_i, z_i\}_{\mathbf{x}_i \in R_m}$. At the first iteration there is a single region containing the entire dataset. Associated with any data subset is a discrepancy measure between the y and z values of the subset:

$$d_m = D(\{y_i\}_{\mathbf{x}_i \in R_m}, \{z_i\}_{\mathbf{x}_i \in R_m}). \quad [1]$$

Choice of a particular discrepancy measure depends on the specific application, as discussed below.

At the next $(M+1)$st iteration each of the regions R_m defined at the Mth iteration $(1 \le m \le M)$ is provisionally partitioned (split) into two regions $R_m^{(l)}$ and $R_m^{(r)}$ $(R_m^{(l)} \cup R_m^{(r)} = R_m)$. Each of these "daughter" regions contains its own data subset with associated discrepancy measure $d_m^{(l)}$ and $d_m^{(r)}$ (Eq. **1**).

Within each separate region the quality of a split is defined as the product of two factors:

$$Q_m(l, r) = (f_m^{(l)} \cdot f_m^{(r)}) \cdot \max(d_m^{(l)}, d_m^{(r)})^\beta. \quad [2]$$

In the first, $f_m^{(l)}$ and $f_m^{(r)}$ are the fraction of observations in the "parent" region R_m associated with each of the two daughters. This factor discourages highly asymmetric splits in anticipation of further splitting. The second factor attempts to isolate daughter regions with high discrepancy. The parameter β regulates the relative influence of the two factors. Results are insensitive to its value. In all examples below the default $\beta = 2$ was used.

The types of splits considered here are the same as in ordinary regression trees (1). Each involves one of the predictor

Significance

Often machine learning methods are applied and results reported in cases where there is little to no information concerning accuracy of the output. Simply because a computer program returns a result does not ensure its validity. If decisions are to be made based on such results it is important to have some notion of their veracity. Contrast trees represent an approach for assessing the accuracy of many types of machine-learning estimates that are not amenable to standard validation methods. In situations where inaccuracies are detected boosted contrast trees can often improve performance. A special case, distribution boosting, provides an assumption-free method for estimating the full probability distribution of an outcome variable given any set of joint input predictor variable values.

Author contributions: J.H.F. designed research, performed research, contributed new reagents/analytic tools, analyzed data, and wrote the paper.

Reviewers: E.I.G., University of Pennsylvania; and M.R.S., University of California San Francisco Medical Center.

The author declares no competing interest.

[1]Email: jhf@stanford.edu.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.1921562117/-/DCSupplemental.

First published August 19, 2020.

variables x_j. For numeric variables splits are specified by a particular value of that variable (split point) s. The corresponding daughter regions are defined by

$$\mathbf{x} \in R_m \,\&\, x_j \le s \Longrightarrow \mathbf{x} \in R_m^{(l)}$$
$$\mathbf{x} \in R_m \,\&\, x_j > s \Longrightarrow \mathbf{x} \in R_m^{(r)}. \quad [3]$$

For categorical variables (factors) the respective levels are ordered by discrepancy Eq. **1**. The discrepancy at each respective level of the factor for the observations in the mth region is computed. Splits are then considered in this order.

Within each current region R_m all possible splits are performed and the one maximizing Eq. **2** is associated with that region. Then, the region whose associated split maximizes actual improvement

$$I_m = \max(d_m^{(l)}, d_m^{(r)}) - d_m \quad [4]$$

is ultimately chosen to create the two new regions at that iteration. These new regions replace the corresponding parent producing $M+1$ total regions. Splitting stops when no estimated improvement (Eq. **4**) is greater than zero, the tree reaches a specified size, or the observation count within all regions is below a specified threshold.

Tree size (number of regions) is generally specified by the user. It involves a trade-off between discrepancy and interpretability. Smaller trees give rise to larger regions defined by simpler conjunctive rules and are thereby easier to interpret. Larger trees have the potential to uncover smaller regions of higher discrepancy defined by more complex rules. Pruning strategies analogous to those in classification and regression trees (1) based on cross-validation can also be employed to guide choice of tree size.

Discrepancy Measures

By defining different discrepancy measures contrast trees can be applied to a variety of different problems. Even within a particular type of problem there may be a number of different appropriate discrepancy measures that can be used.

When the two outcomes are simply functions of $\mathbf{x}$, $y = f(\mathbf{x})$ and $z = g(\mathbf{x})$, any quantity that reflects their difference in values at the same $\mathbf{x}$ can be used to form a discrepancy measure such as

$$d_m = \frac{1}{N_m} \sum_{\mathbf{x}_i \in R_m} |\, y_i - z_i \,|. \quad [5]$$

Here N_m is the number of observations in the region R_m. If y is a random variable and z is an estimate for the mean of its conditional distribution at $\mathbf{x}$, $z_i = \hat{E}(y \,|\, \mathbf{x}_i)$, a natural discrepancy measure is

$$d_m = \frac{1}{N_m} \left| \sum_{\mathbf{x}_i \in R_m} (y_i - z_i) \right|. \quad [6]$$

This discrepancy (Eq. **6**) reflects the absolute difference between the empirical mean of the outcomes $\{y_i\}_{\mathbf{x}_i \in R_m}$ and that of the corresponding predictions $\{z_i\}_{\mathbf{x}_i \in R_m}$ in the region. Alternatively, if z is an estimate for the pth quantile at $\mathbf{x}$, $z_i = \hat{Q}_p(y \,|\, \mathbf{x}_i)$, a natural discrepancy measure would be lack of coverage in the region

$$d_m = \left| p - \frac{1}{N_m} \sum_{\mathbf{x}_i \in R_m} I(y_i < z_i) \right|. \quad [7]$$

If $y \sim p_y(y \,|\, \mathbf{x})$ and $z \sim p_z(z \,|\, \mathbf{x})$ are both independent random variables associated with each $\mathbf{x}$, a discrepancy measure reflects the distance between their respective distributions. There are many proposed empirical measures of distribution distance. Every two-sample test has one. For the examples below a variant of the Anderson–Darling (2) statistic is used. Let $\{t_i\} = \{y_i\} \cup \{z_i\}$ represent the pooled (y, z) sample in a region R_m. Then, discrepancy between the distributions of y and z is taken to be

$$d_m = \frac{1}{2N_m - 1} \sum_{i=1}^{2N_m - 1} \frac{\left| \hat{F}_y(t_{(i)}) - \hat{F}_z(t_{(i)}) \right|}{\sqrt{i \cdot (2N_m - i)}}, \quad [8]$$

where $t_{(i)}$ is the ith value of t in sorted order and $\hat{F}_y$ and $\hat{F}_z$ are the respective empirical cumulative distributions of y and z. Note that this discrepancy measure (Eq. **8**) can be employed in the presence of arbitrarily censored or truncated data simply by employing a nonparametric method to estimate the respective *cumulative distribution functions* (CDFs) such as ref. 3.

Discrepancy measures can be, and often are, customized to particular applications. In this sense they are similar to loss criteria in prediction problems. However, in the context of contrast trees (and boosting) there is no requirement that they be convex or even differentiable. Moreover, discrepancies need not be expressible as a sum of terms each involving a single observation as in Eq. **5**. Examples are Eqs. **6–8**.

Boosting Contrast Trees

As indicated above, and illustrated in the examples presented below and in *SI Appendix*, contrast trees can be employed as diagnostics to examine the lack of accuracy of predictive models. To the extent that inaccuracies are uncovered, boosted contrast trees can be used to attempt to mitigate them, thereby producing more accurate predictions. Contrast boosting derives successive modifications to an initially specified z, each reducing its discrepancy with y over the data. Prediction then involves starting with the initial value of z and then applying the modifications to produce the resulting estimate.

Estimation Contrast Boosting. In this case z is taken to be an estimate of some parameter of $p_y(y \,|\, \mathbf{x})$. The z values within each region $R_m^{(1)}$ of a contrast tree can be modified $z \to z^{(1)} = z + \delta_m^{(1)}$ $(\mathbf{x} \in R_m^{(1)})$ so that the discrepancy Eq. **1** with y is zero in that region:

$$D(\{y_i\}_{\mathbf{x}_i \in R_m^{(1)}}, \{z_i^{(1)}\}_{\mathbf{x}_i \in R_m^{(1)}}) = 0. \quad [9]$$

This in turn yields zero average discrepancy between y and $z^{(1)}$ over the regions defined by the terminal nodes of the corresponding contrast tree. However, there may well be other partitions of the $\mathbf{x}$ space defining different regions $\{R_m^{(2)}\}_1^M$ for which this discrepancy is not small. These may be uncovered by building a second tree to contrast y with $z^{(1)}$ producing updates

$$z^{(2)} = z^{(1)} + \delta_m^{(2)} \,(\mathbf{x} \in R_m^{(2)}). \quad [10]$$

These in turn can be contrasted with y to produce new regions $\{R_m^{(3)}\}_1^M$ and corresponding updates $\{\delta_m^{(3)}\}_1^M$. Such iterations can be continued K times until the updates become small. As with gradient boosting (4) performance accuracy is often improved by imposing a learning rate. At each step k the computed update $\delta_m^{(k)}$ in each region $R_m^{(k)}$ is reduced by a factor $0 < \alpha \le 1$. That is, $\delta_m^{(k)} \leftarrow \alpha\, \delta_m^{(k)}$ in Eq. **10**.

Each tree k in the boosted sequence $1 \le k \le K$ partitions the $\mathbf{x}$ space into a set of regions $\{R_m^{(k)}\}$. Any point $\mathbf{x}$ lies within one region $m_k(\mathbf{x})$ of each tree with corresponding update $\delta_{m_k(\mathbf{x})}^{(k)}$. Starting with a specified initial value $z(\mathbf{x})$ the estimate $\hat{z}(\mathbf{x})$ at $\mathbf{x}$ is then

$$\hat{z}(\mathbf{x}) = z(\mathbf{x}) + \sum_{k=1}^{K} \delta_{m_k(\mathbf{x})}^{(k)}. \quad [11]$$

Distribution Contrast Boosting. Here y and z are both considered to be random variables independently generated from respective distributions $p_y(y \mid \mathbf{x})$ and $p_z(z \mid \mathbf{x})$. The purpose of a contrast tree is to identify regions of $\mathbf{x}$ space where the two distributions most differ. The goal of distribution boosting is to estimate a (different) transformation of z at each $\mathbf{x}$, $g_{\mathbf{x}}(z)$, such that the distribution of the transformed variable is the same as that of y at $\mathbf{x}$. That is,

$$p_{g_{\mathbf{x}}}(g_{\mathbf{x}}(z) \mid \mathbf{x}) = p_y(y \mid \mathbf{x}). \quad [12]$$

Thus, starting with z values sampled from a known distribution $p_z(z \mid \mathbf{x})$ at each $\mathbf{x}$, one can use the estimated transformation $\hat{g}_{\mathbf{x}}(z)$ to obtain an estimate $\hat{p}_y(y \mid \mathbf{x})$ of the y distribution at that $\mathbf{x}$. Note that the transformation $g_{\mathbf{x}}(z)$ is usually a different function of z at each different $\mathbf{x}$.

The z values within each region $R_m^{(1)}$ of a contrast tree can be transformed $z^{(1)} = g_m^{(1)}(z)$ $(\mathbf{x} \in R_m^{(1)})$ so that the discrepancy (Eq. **8**) with y is zero in that region. The transformation is given by

$$g_m^{(1)}(z) = \hat{F}_y^{-1}\left(\hat{F}_z(z)\right), \quad [13]$$

where $\hat{F}_y(y)$ is the empirical cumulative distribution of y for $\mathbf{x} \in R_m^{(1)}$ and $\hat{F}_z(z)$ is the corresponding distribution of z for $\mathbf{x} \in R_m^{(1)}$. This transformation function is represented by the quantile–quantile (QQ) plot of y versus z in the region.

As with estimation boosting, the distribution of the modified (transformed) variable $z^{(1)}$ can then be contrasted with that of y using another contrast tree. This produces another region set $\{R_m^{(2)}\}_1^M$ where the distributions of y and $z^{(1)}$ differ. This discrepancy (Eq. **8**) can be removed by transforming $z^{(1)}$ to match the distribution of y in each new region $z^{(2)} = g_m^{(2)}(z^{(1)})$ $(\mathbf{x} \in R_m^{(2)})$. These in turn can be contrasted with y producing new regions each with a corresponding transformation function. Such distribution boosting iterations can be continued K times until the discrepancy between the distributions of $z^{(K)}$ and y becomes small in each new region. As with estimation, moderating the learning rate by shrinking each estimated transformation function toward identity $g_m^{(k)}(z) \leftarrow (1-\alpha)\, z + \alpha\, g_m^{(k)}(z)$ usually increases accuracy at the expense of computation (more transformations).

Predicting $p_y(y \mid \mathbf{x})$ starts with a sample $\{z_i\}_1^n$ drawn from the specified distribution of z, $p_z(z \mid \mathbf{x})$, at $\mathbf{x}$. This $\mathbf{x}$ lies within one of the regions $m_k(\mathbf{x})$ of each contrast tree k with corresponding transformation function $g_{m_k(\mathbf{x})}^{(k)}(\cdot)$. A given value of z can be transformed to a estimated value for y, $\hat{y} = \hat{g}_{\mathbf{x}}(z)$, where

$$\hat{g}_{\mathbf{x}}(z) = g_{m_K(\mathbf{x})}^{(K)}(g_{m_{K-1}(\mathbf{x})}^{(K-1)}(g_{m_{K-2}(\mathbf{x})}^{(K-2)} \cdots g_{m_1(\mathbf{x})}^{(1)}(z))). \quad [14]$$

That is, the transformed output of each successive tree is further transformed by the next tree in the boosted sequence. A different transformation is chosen at each step depending on the region of the corresponding tree containing the particular joint values of the predictor variables $\mathbf{x}$. With K trees each containing M regions (terminal nodes) there are M^K potentially different transformations $\hat{g}_{\mathbf{x}}(z)$ each corresponding to different values of $\mathbf{x}$. To the extent the overall transformation estimate $\hat{g}_{\mathbf{x}}(z)$ is accurate, the distribution of the transformed sample $\{\hat{y}_i = \hat{g}_{\mathbf{x}}(z_i)\}_1^n$ can be regarded as being similar to that of y at $\mathbf{x}$, $p_y(y \mid \mathbf{x})$. Statistics computed from the values of $\hat{y}$ estimating selected properties of its distribution, or the distribution itself, can be regarded as estimates of the corresponding quantities for $p_y(y \mid \mathbf{x})$.

Diagnostics

In this section we illustrate use of contrast trees as diagnostics for uncovering and understanding the lack of fit of predictive models for classification and conditional distribution estimation. Quantile regression models are examined in *SI Appendix*. All predictive models used for illustration were applied using their respective default procedure parameter settings.

Classification. Contrast tree classification diagnostics are illustrated on the census income data obtained from the Irvine Machine Learning Repository (5). This data sample, taken from 1994 US census data, consists of observations from 48,842 people divided into a training set of 32,561 and an independent test set of 16,281. The outcome variable y is binary and indicates whether or not a person's income is greater than \$50,000 per year. There are 14 predictor variables $\mathbf{x}$ consisting of various demographic and financial properties associated with each person. Here we use contrast trees to diagnose the classification predictions of gradient-boosted regression trees (4).

The predictive model produced by the gradient-boosting procedure applied to the training data set produced an error rate of 13% on the test data. This quantity is the expected error as averaged over all test set predictions. It may be of interest to discover certain $\mathbf{x}$ values for which expected error is much higher or lower. This can be ascertained by contrasting the binary outcome variable y with the model prediction z.

A natural discrepancy measure for this application is misclassification risk (error rate) in each region R_m:

$$d_m = \frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq z_i). \quad [15]$$

The goal in applying contrast trees is to uncover regions in $\mathbf{x}$ space with exceptionally high values of Eq. **15**. For this purpose the test dataset was randomly divided into two parts of 10,000 and 6,281 observations. A 10-region contrast tree was built on the 10,000 test dataset. Fig. 1 summarizes these regions using the separate 6,281-observation dataset. The upper bar plot shows the misclassification risk of the gradient boosting classifier in each region ordered from largest to smallest. The lower bar plot indicates the observation count in each respective region. The number below each bar is simply the contrast tree node identifier for that region. The horizontal (red) line indicates the 13% average error rate.

As Fig. 1 indicates, the contrast tree has uncovered many regions with substantially higher error rates than the overall average and several others with substantially lower error rates. The lowest-error region covers 43% of the test set observations with an average error rate of 2.7%. The highest-error region covering 5.6% of the data has an average error rate of 41%.

Each of the regions represented in Fig. 1 is easily described. For example, the rule defining the lowest-error region is

Node 4
relationship $\in$ {Own-child, Husband, Not-in-family, Other-relative}
&
education $\leq$ 12

Predictions satisfying that rule suffer only a 2.7% average error rate. Predictions satisfying the rule defining the highest-error region

Node 30
relationship $\notin$ {Own-child, Husband, Not-in-family, Other-relative}
&
occupation $\in$ { Exec-managerial, Transport-moving, Armed-Forces }
&
education $\leq$ 12

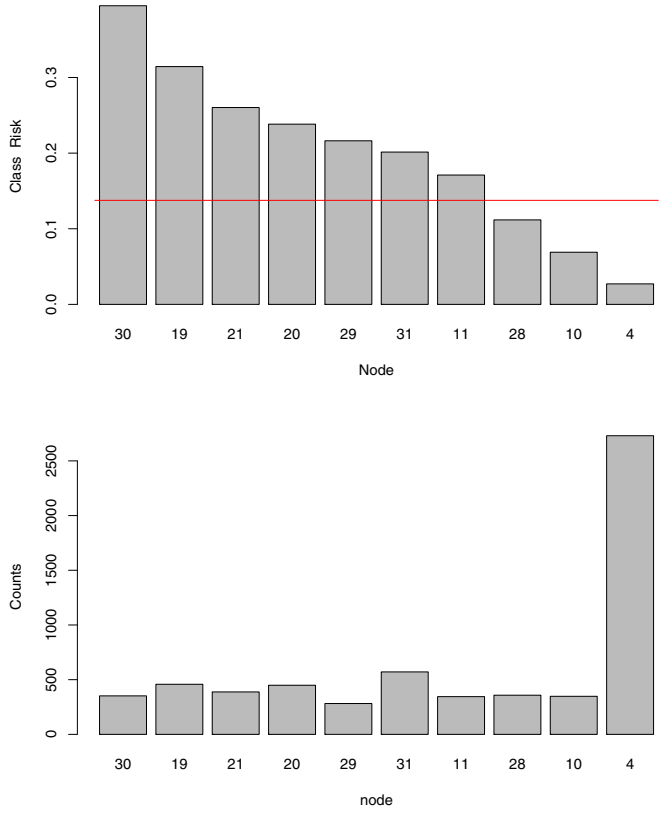

Fig. 1. Misclassification risk (error rate) (*Upper*) and observation count (*Lower*) of classification contrast tree regions on census income data.

have a 41% average error rate. Thus, confidence in salary predictions for people in node 4 might be higher than for those in node 30.

Probability Estimation. The discrepancy measure Eq. **15** is appropriate for procedures that predict a class identity and the corresponding contrast tree attempts to identify **x** values associated with high levels of misclassification. Some procedures such as gradient boosting return estimated class probabilities at each **x** which are then thresholded to predict class identities. In this case the probability estimate contains information concerning expected classification accuracy. The closer the respective class probabilities are to each other the higher is the likelihood of misclassification. This shifts the issue from classification accuracy to probability estimation accuracy which can be assessed with a contrast tree.

For binary classification a natural discrepancy for probability estimation is Eq. **6** where $y \in \{0, 1\}$ is the binary outcome variable and $0 \le z \le 1$ is its predicted probability $\widehat{\Pr}(y = 1)$. This measures the difference between the empirical probability of $y = 1$ in region R_m and the corresponding average probability prediction z in that region. The gradient boosting probability estimates were based on the training dataset. A 10-terminal-node contrast tree was built on the census income data using the 10,000-observation test dataset with corresponding node statistics evaluated on the separate 6,281-observation test dataset.

Fig. 2, *Upper* shows the empirical probability $y = 1$ (blue) and the average gradient boosting prediction z (red) within each region of the resulting contrast tree. Fig. 2, *Lower* shows the number of counts in each corresponding region. One sees a general trend of oversmoothing. The largest probability is being underestimated, whereas the smaller ones are substantially overestimated by the gradient-boosting procedure. As above each of these regions is defined by simple rules based on the values of a few predictor variables.

A convenient way to summarize the overall results of a contrast tree is through its corresponding lack-of-fit contrast curve. For each region R_m containing N_m counts, the observation weighted average of its discrepancy d_m and those with higher discrepancy

$$\bar{d}_m = \sum_{d_j \ge d_m} d_j N_j \Big/ \sum_{d_j \ge d_m} N_j \qquad [16]$$

is plotted on the vertical axis. The fraction of observations in those same regions

$$f_m = \frac{1}{N} \sum_{d_j \ge d_m} N_j \qquad [17]$$

is plotted along the horizontal axis. The leftmost point on each curve thus represents the discrepancy value of the largest discrepancy region of its corresponding tree. The rightmost point gives the discrepancy averaged over all regions. Intermediate points give average discrepancy over the highest discrepancy regions containing the corresponding fraction of observations.

The black curve in Fig. 3 shows the lack-of-fit contrast curve for the gradient boosting estimates based on a 50-node contrast tree built in the same manner as the one shown in Fig. 2. Its error in estimated probability averaged over all test set predictions is seen to be 0.066 (extreme right). The error corresponding to the

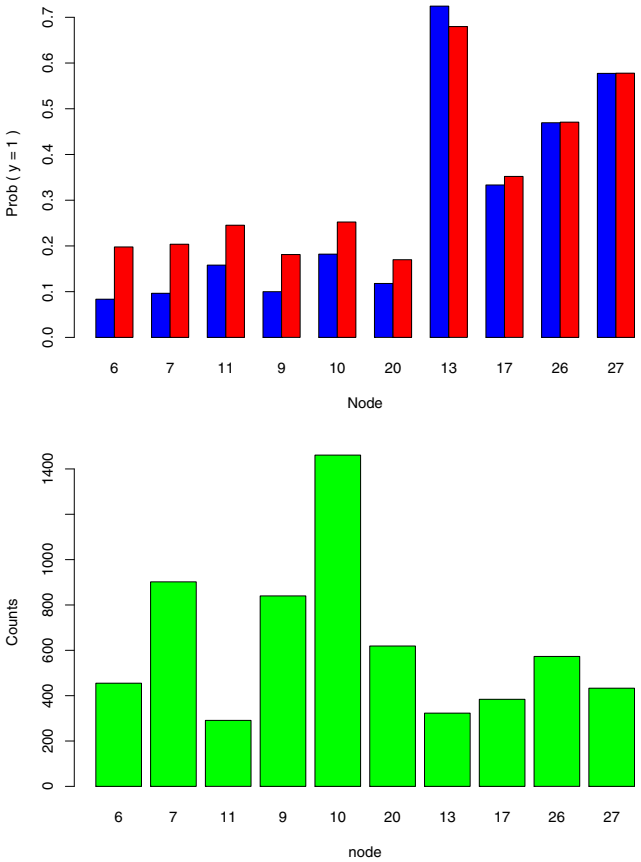

Fig. 2. Census income data. (*Upper*) Fraction of positive observations (blue) and mean probability prediction (red) for probability contrast tree regions. (*Lower*) Observation count in each region.

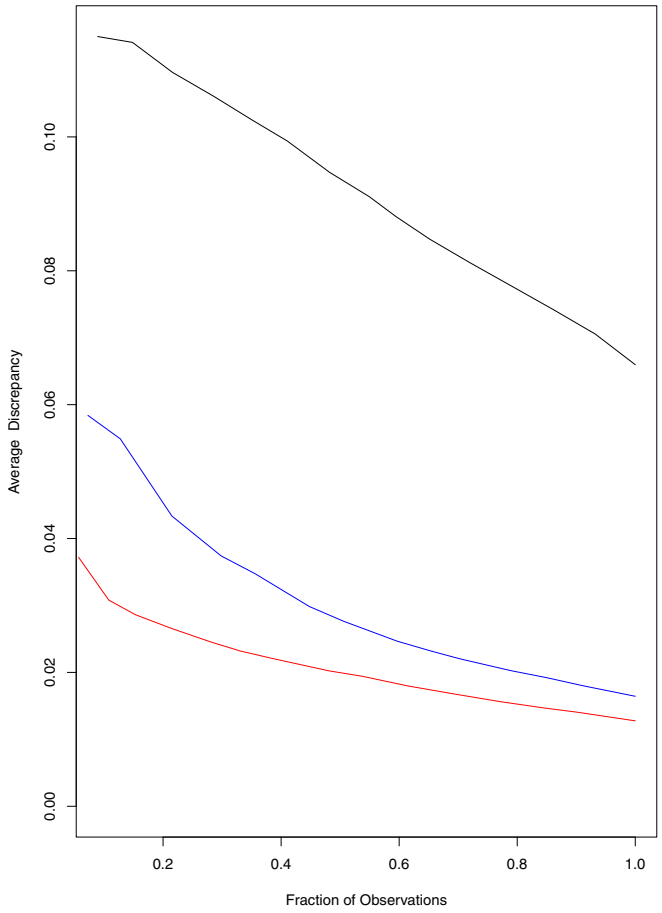

Fig. 3. Census income data. Lack-of-fit contrast curves comparing accuracy of Pr($y = 1$) estimates by logistic gradient boosting (black), random forests (blue), and probability gradient boosting (red).

largest discrepancy region (extreme left) is 0.115. The blue curve is the corresponding lack-of-fit contrast curve for random forest probability prediction (6). Its average error is less than one third of that for gradient boosting and its worst error is 50% less.

The contrast tree as represented in Fig. 2 suggests that the problem with the gradient boosting procedure here is oversmoothing. It is failing to accurately estimate the extreme probability values. Gradient boosting for binary probability estimation generally uses a negative Bernoulli log-likelihood loss function based on a logistic distribution. The logistic transformation to modeling on the log-odds scale inhibits the estimation of extreme probability values. Random forests use regression trees that model directly on the probability scale using squared-error loss. This suggests that using a similar approach with gradient boosting for this problem may improve performance, especially at the extreme values.

The red curve in Fig. 3 shows the corresponding lack-of-fit contrast curve for direct probability estimation with gradient boosting using squared-error loss. This change has dramatically improved accuracy of gradient-boosting probability estimates. Both its average and maximum discrepancies are seen to be at least four times smaller than those using the approach based on logistic regression.

Fig. 4 shows the corresponding test data results of applying contrast boosting to the training data output of each of the methods shown in Fig. 3. Comparing the two figures one sees that the accuracy of logistic gradient boosting is dramatically improved while that of random forest is substantially improved. The improvement to probability gradient boosting using squared-error loss is seen to be moderate.

Table 1 shows classification error rate for each of the three original methods plus that of the best contrast-boosting result. They are all seen to be very similar. This illustrates that prediction error on the random outcome variable can be a very poor proxy for estimation accuracy of the distribution mean ($\Pr(y = 1)$). Here the oversmoothing of probability estimates caused by modeling log odds does not change many class assignments. In some applications accurate estimation of extreme probabilities is important, such as with highly asymmetric misclassification losses. In such cases directly estimating on the probability scale may be superior to indirectly estimating on the log-odds scale.

Conditional Distributions. Here we consider the case in which both y and z are considered to be random variables independently drawn from respective distributions $p_y(y \mid \mathbf{x})$ and $p_z(z \mid \mathbf{x})$. Interest is in contrasting these two distributions as functions of $\mathbf{x}$. Specifically, we wish to uncover regions of $\mathbf{x}$ space where the distributions most differ. For this we use contrast trees with discrepancy measure Eq. **8**.

A well-known way to approximate $p_y(y \mid \mathbf{x})$ under the assumption of homoskedasticity is through the residual bootstrap (7). One obtains a location estimate such as the conditional median $\hat{m}(y \mid \mathbf{x})$ and forms the data residuals $r_i = y_i - \hat{m}(y \mid \mathbf{x}_i)$ for each observation $1 \leq i \leq N$. Under the assumption that the conditional distribution of r, $p_r(r \mid \boldsymbol{x})$, is independent of $\mathbf{x}$

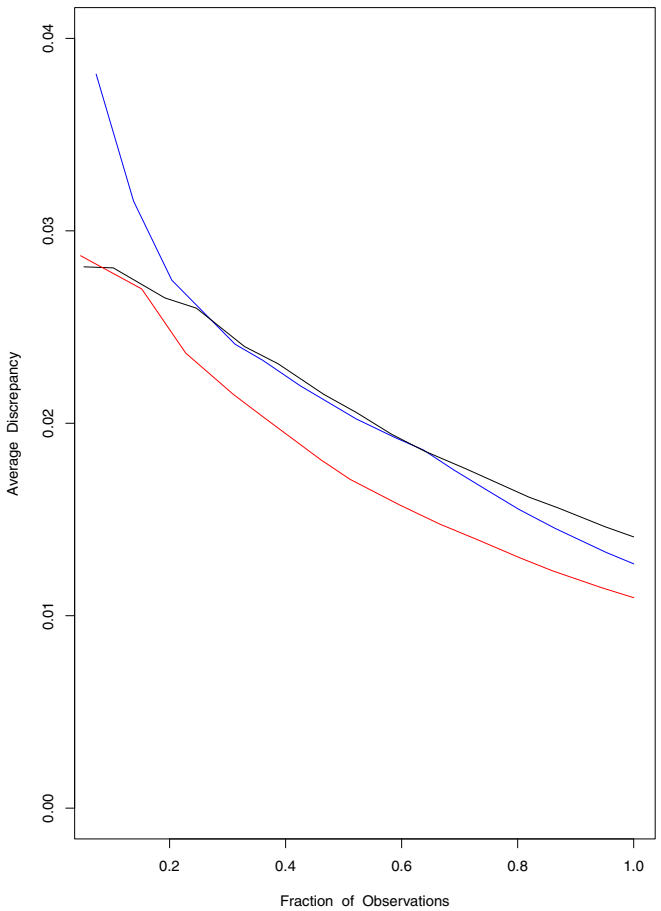

Fig. 4. Census income data. Lack-of-fit contrast curves comparing accuracy of Pr($y = 1$) estimates after applying contrast boosting to the output of logistic gradient boosting (black), random forests (blue), and probability gradient boosting (red).

Table 1. Classification error rates corresponding to several probability estimation methods

Method	Error rate
Logistic gradient boosting	13.0
Probability gradient boosting	12.9
Random forest	13.6
Probability gradient boosting + contrast	12.8

(homoskedasticity) one can draw random samples from $p_y(y \mid \mathbf{x}_i)$ as $y_i = \hat{m}(y \mid \mathbf{x}_i) + r_{\pi(i)}$ where $\pi(i)$ is random permutation of the integers $i \in [1, N]$. These samples can then be used to derive various regression statistics of interest.

A fundamental ingredient for the validity of residual bootstrap approach is the homoskedasticity assumption. Here we test this on the online news popularity dataset (8) also available from the Irvine Machine Learning Data Repository. It summarizes a heterogeneous set of features about articles published by the Mashable website over a period of 2 y. The goal is to predict the number of shares y in social networks (popularity). There are $N = 39{,}797$ observations (articles). Associated with each are $p = 59$ attributes to be used as predictor variables $\mathbf{x}$. These are described at the download website. Gradient boosting was used to estimate the median function $\hat{m}(y \mid \mathbf{x})$, and $\{z_i\}_{i=1}^{N}$ was taken as a corresponding residual bootstrap sample to be contrasted with y.

Fig. 5 shows QQ plots of y versus z for the nine highest discrepancy regions of a 50-node contrast tree. The red line represents equality. One sees that there are $\mathbf{x}$ values (regions) where the distribution of y is very different from its residual bootstrap approximation z; homoskedasticity is rather strongly violated. The average discrepancy (Eq. **8**) over all 50 regions is 0.19.

The outcome variable y (number of shares) is strictly positive and its marginal distribution is highly skewed toward larger values. In such situations it is common to model its logarithm. Fig. 6 shows the corresponding results for contrasting the distribution of $\log_{10}(y)$ with its residual bootstrap counterpart. Homoskedasticity appears to more closely hold on the logarithm scale but there are still regions of $\mathbf{x}$ space where the approximation is not good. Here the average discrepancy (Eq. **8**) over all 50 regions is 0.13. A null distribution for average discrepancy under the hypothesis of homoskedasticity can be obtained by repeatedly contrasting pairs of randomly generated $\log_{10}(y)$ residual bootstrap distributions. Based on 50 replications, this distribution had a mean of 0.078 with an SD of 0.003.

Distribution Boosting: Simulated Data

The notion of distribution boosting is sufficiently unusual that we first illustrate it on simulated data where the estimates $\hat{p}_y(y \mid \mathbf{x})$ can be compared to the true data generating distributions $p_y(y \mid \mathbf{x})$. Distribution boosting applied to the online news popularity data are presented in *SI Appendix*.

Data. There are N = 25,000 training observations each with a set of $p = 10$ predictor variables $\mathbf{x}_i$ randomly generated from a standard normal distribution. The outcome variable $y \mid \mathbf{x}$ is generated from a transformed asymmetric logistic distribution (9)

$$y = h(f(\mathbf{x}) + \eta(\mathbf{x})) \qquad [18]$$

with the random component being $\eta(\mathbf{x}) = -|\varepsilon| \cdot s_l(\mathbf{x})$ with probability $P_l = s_l(\mathbf{x})/(s_l(\mathbf{x}) + s_u(\mathbf{x}))$ and $\eta(\mathbf{x}) = +|\varepsilon| \cdot s_u(\mathbf{x})$ with probability $s_u(\mathbf{x})/(s_l(\mathbf{x}) + s_u(\mathbf{x}))$. Here ε is a standard logistic random variable. The transformation $h(z)$ is taken to be

$$h(z) = sign(z)\,(0.5\,|\,z\,| + 1.5\,z^2). \qquad [19]$$

The untransformed mode $f(\mathbf{x})$ and lower/upper scales $s_l(\mathbf{x})/\,s_u(\mathbf{x})$ are each different functions of the 10 predictor variables $\mathbf{x}$. The simulated mode function is taken to be

$$f(\mathbf{x}) = \sum_{j=1}^{10} c_j\, B_j(x_j)\,/\,std_{x_j}(B_j(x_j)) \qquad [20]$$

with the value of each coefficient c_j being randomly drawn from a standard normal distribution. Each basis function takes the form

$$B_j(x_j) = sign(x_j)\,|\,x_j\,|^{r_j} \qquad [21]$$

with each exponent r_j being separately drawn from a uniform distribution $r_j \sim U(0, 2)$. The denominator in each term of Eq. **20** prevents the suppression of the influence of highly nonlinear terms in defining $f(\mathbf{x})$.

The scale functions are taken to be $s_l(\mathbf{x}) = 0.2 + \exp(t_l(\mathbf{x}))$ and $s_u(\mathbf{x}) = 0.2 + \exp(t_u(\mathbf{x}))$ where the log-scale functions $t_l(\mathbf{x})$ and $t_u(\mathbf{x})$ are constructed in the same manner as Eqs. **20** and **21** but with different randomly drawn values for the 20 parameters $\{c_j, r_j\}_1^{10}$ producing different functions of $\mathbf{x}$. The average pairwise absolute correlation between the three functions is 0.18. The overall resulting distribution $p(y \mid \mathbf{x})$ (Eqs. **18–21**) has location, scale, asymmetry, and shape being highly dependent on the joint values of the predictors $\mathbf{x}$ in a complex and unrelated way.

Conditional Distribution Estimation. Distribution boosting is applied to this simulated data to estimate its distribution $p_y(y \mid \mathbf{x})$ as a function of $\mathbf{x}$. For each observation the contrasting

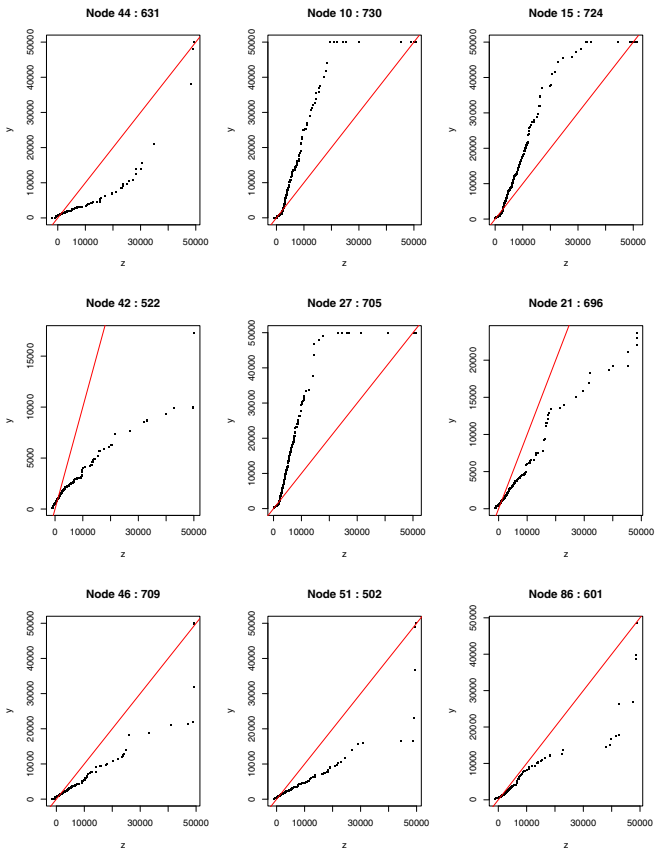

Fig. 5. QQ plots of *y* versus parametric bootstrap *z* distributions for the nine highest discrepancy regions of a 50-node contrast tree using online news popularity data. The red line represents equality.

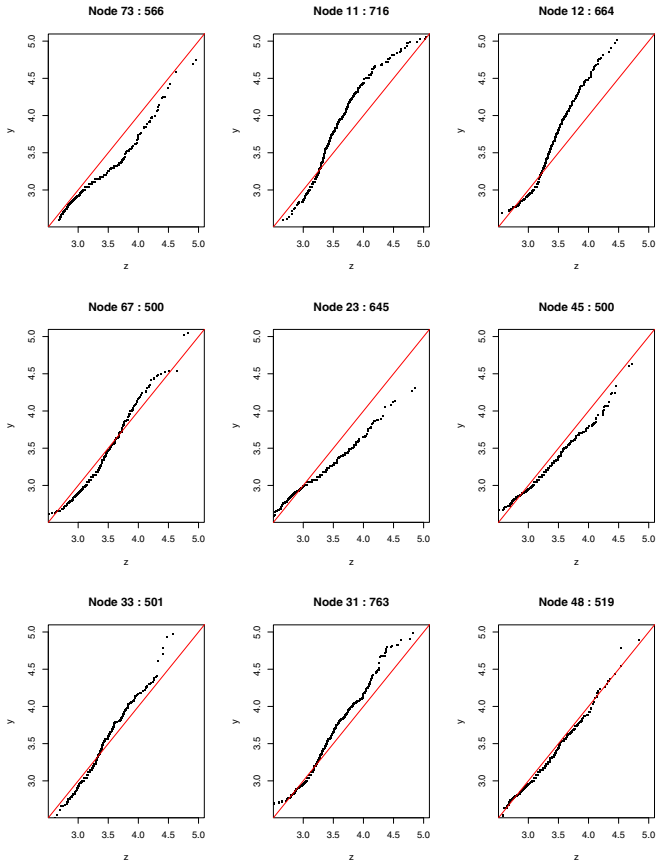

Fig. 6. QQ plots of $\log_{10}(y)$ versus corresponding parametric bootstrap z distributions for the nine highest-discrepancy regions of a 50-node contrast tree using online news popularity data. The red line represents equality.

random variable z is taken to be independently generated from the same normal distribution, $z \,|\, \mathbf{x} \sim N(\bar{y}, \sigma_y^2)$, independent of $\mathbf{x}$. Here $\bar{y}$ and σ_y^2 are the mean and variance of the marginal y distribution. The goal is to produce an estimated transformation of z, $\hat{y} = \hat{g}_{\mathbf{x}}(z)$, at each $\mathbf{x}$ such that $p_{\hat{y}}(\hat{y} \,|\, \mathbf{x}) = p_y(y \,|\, \mathbf{x})$. To the extent the estimate $\hat{g}_{\mathbf{x}}(z)$ accurately reflects the true transformation function $g_{\mathbf{x}}(z)$ at each $\mathbf{x}$ one can apply it to a sample drawn from $z \sim N(\bar{y}, \sigma_y^2)$ to produce a corresponding sample drawn from the distribution $y \sim p_y(y \,|\, \mathbf{x})$. This sample can then be used to plot that distribution or compute the value of any of its properties.

Fig. 7 plots the average terminal node discrepancy Eq. **8** for 400 iterations of distribution boosting applied to the training data, as evaluated on a 25,000 observation independent "test" dataset generated from the same joint $(\mathbf{x}, y)$ distribution (Eqs. **18**–**21**). Results are shown for the first and then every 10th successive tree. The red line is a running median smooth. The test set discrepancy is seen to generally decrease with increasing number of trees. There is a diminishing return after about 200 iterations (trees).

Note that with contrast boosting average tree discrepancy on test or even training data does not necessarily decrease monotonically with successive iterations (trees). Each contrast tree represents a greedy solution to a nonconvex optimization with multiple local optima. As a consequence, the inclusion of an additional tree can, and often does, increase average discrepancy of the current ensemble. Boosting is continued as long as there is a general downward trend in average tree discrepancy.

Lack of fit to the data of any model for the distribution $p_y(y \,|\, \mathbf{x})$ can be assessed by contrasting y with a sample drawn from that distribution. Fig. 8 shows QQ plots of y versus initial z (everywhere the same normal) for the nine highest discrepancy regions of a 10-node tree contrasting the two quantities on the test data set. The red lines represent equality. One sees that $p_y(y \,|\, \mathbf{x})$ is here far from being everywhere the same normal.

For the distribution boosted model $\hat{y} = \hat{g}_{\mathbf{x}}(z)$ lack-of-fit can be assessed by contrasting the distributions of y and $\hat{y}$ with a contrast tree using the test dataset. Fig. 9 shows QQ plots of y versus $\hat{y}$ for the nine highest-discrepancy regions of a 10-node tree contrasting the two quantities on the test dataset. The red lines represent equality. The transformation $\hat{g}_{\mathbf{x}}(z)$ at each separate $\mathbf{x}$ value was evaluated using the 400-tree model built on the training data. The nine highest-discrepancy regions shown in Fig. 9 together cover 27% of the data. They show that while the transformation model fits most of the test data quite well, it is not everywhere perfect. There are minor departures between the two distributions in some small regions. However, these discrepancies appear in sparse tails where QQ plots themselves can be unstable.

A measure of the difference between the estimated and true CDFs at each $\mathbf{x}$ can be defined as

$$Diff\ (\mathbf{x}) = \sqrt{\frac{1}{100}\sum_{j=1}^{100}(CDF_{\mathbf{x}}(u_j) - \widehat{CDF}_{\mathbf{x}}(u_j))^2}, \qquad \textbf{[22]}$$

where $CDF_{\mathbf{x}}$ is the true cumulative distribution of $y \,|\, \mathbf{x}$ computed from Eqs. **18**–**21** and $\widehat{CDF}_{\mathbf{x}}$ is the corresponding estimate

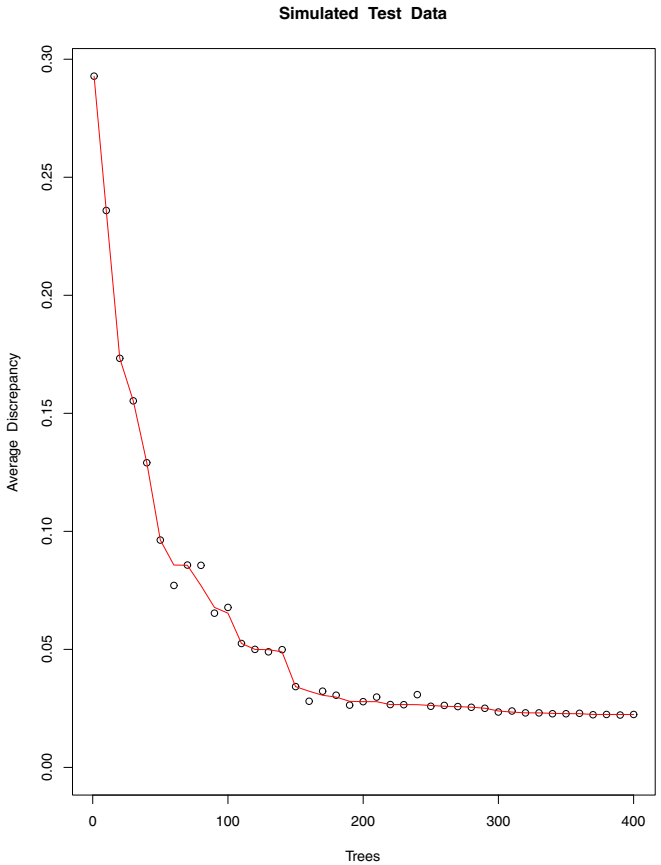

Fig. 7. Test data discrepancy averaged over the terminal nodes (regions) of successive contrast trees for the first and then every 10th iteration for 400 iterations of distribution boosting on simulated training data. The solid red curve is a running median smooth.

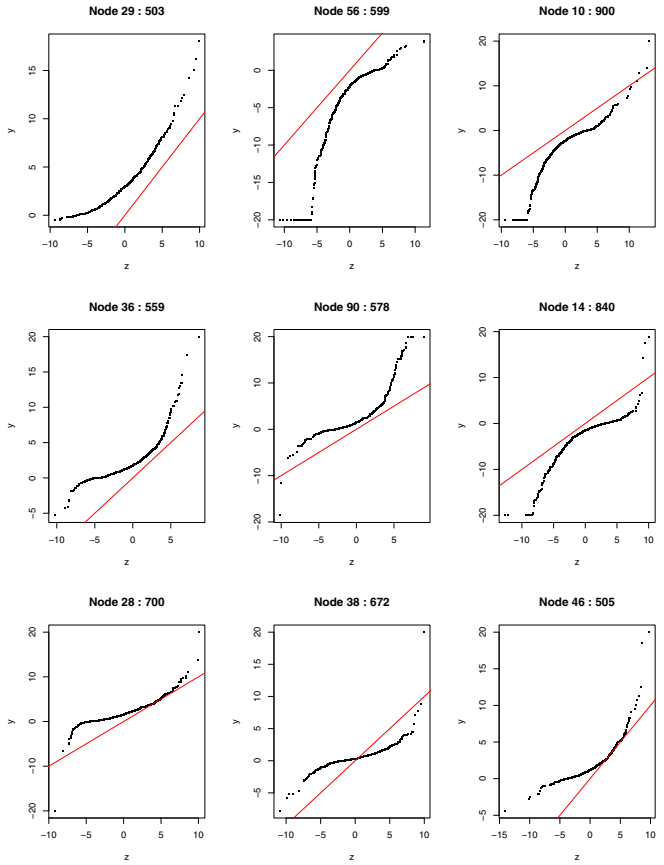

Fig. 8. QQ plots of *y* versus *z* (normal) for the nine highest discrepancy regions of a 10-node contrast tree on the simulated test dataset. The red lines represent equality.

from the distribution boosting model. The 100 evaluation points $\{u_j\}_1^{100}$ are a uniform grid between the 0.001 and 0.999 quantiles of the true distribution $CDF_{\mathbf{x}}$.

Fig. 10 summarizes the overall accuracy of the distribution boosting model. The upper left frame shows a histogram of the distribution of Eq. **22** for observations in the test dataset. The 50th, 75th, and 90th percentiles of this distribution are, respectively, 0.0352, 0.0489, and 0.0773, indicated by the red marks. The remaining plots show estimated (black) and true (red) distributions for the three observations with Eq. **22** equal to these respective percentiles. Thus 50% of the estimated distributions are closer to the truth than that shown in the upper right frame. Seventy-five percent are closer than that shown in the lower left frame, and 90% are closer than that seen in the lower right frame.

Distribution boosting produces an estimate for the full distribution of $y \mid \mathbf{x}$ by providing a function $\hat{g}_{\mathbf{x}}(z)$ that transforms a random variable z with a known distribution $p_z(z \mid \mathbf{x})$ to the estimated distribution $\hat{p}_y(y \mid \mathbf{x})$. One can then easily compute any statistic $\hat{S}(\mathbf{x}) = S[\,\hat{p}_y(y \mid \mathbf{x})]$, which can be used as an estimate for the value of the corresponding quantity $S(\mathbf{x}) = S[\,p_y(y \mid \mathbf{x})]$ on the actual distribution. For some quantities $S(\mathbf{x})$, an alternative is to directly estimate them by minimizing empirical prediction risk based on an appropriate loss function

$$\hat{S}(\mathbf{x}) = \arg\min_{f \in \Im} \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(\mathbf{x}_i)), \qquad [23]$$

where $\Im$ is the function class associated with the learning method. Here we compare distribution boosting estimates of the quartiles $Q_p(\mathbf{x})$, $p \in [0.25, 0.5, 0.75]$, with those of gradient boosting quantile regression, which uses loss

$$L_p(y, z) = (1-p)\,(z-y)_+ + p\,(y-z)_+, \qquad [24]$$

on the simulated dataset where the truth is known.

Fig. 11 shows true versus predicted values for each of the two methods (rows) on the three quartiles (columns). The red lines represent a running median smooth and the blue lines show equality. The average absolute error AAE associated with each of these plots is

$$AAE(h, v) = mean(|\,h - v\,|)/mean(|\,v - median(v)\,|), \qquad [25]$$

where h is the quantity plotted on the horizontal and v the vertical axes. The quantile values derived from the estimates of the full distribution (Fig. 11, Lower) are here seen to be somewhat more accurate than those obtained from gradient boosting quantile regression (Fig. 11, *Upper*).

With quantile regression each quantile is estimated separately without regard to estimates of other quantiles. Distribution boosting quantile estimates are all derived from a common probability distribution and thus have order constraints imposed among them. For example, two quantile estimates have the property $\hat{Q}_p(\mathbf{x}) < \hat{Q}_{p'}(\mathbf{x})$ for all $p < p'$ at any $\mathbf{x}$. These implicit constraints can improve accuracy especially when the quantile estimates are being used to compute quantities derived from them.

There is an additional advantage of computing quantities such as means or quantiles from the estimated conditional

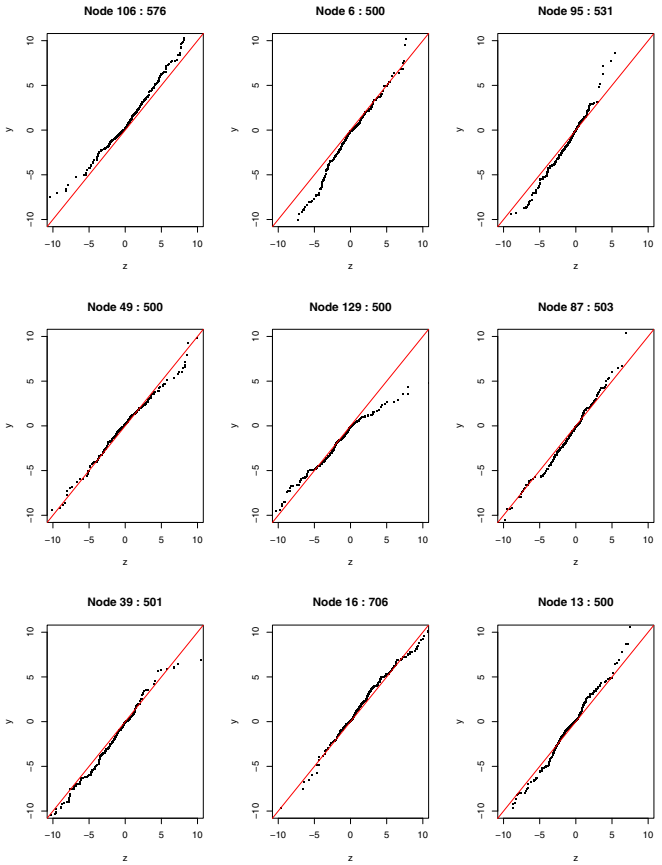

Fig. 9. QQ plots of *y* versus $\hat{y} = \hat{g}_{\mathbf{x}}(z)$ for the nine highest-discrepancy regions of a 10-node contrast tree on the simulated test dataset. The red lines represent equality.

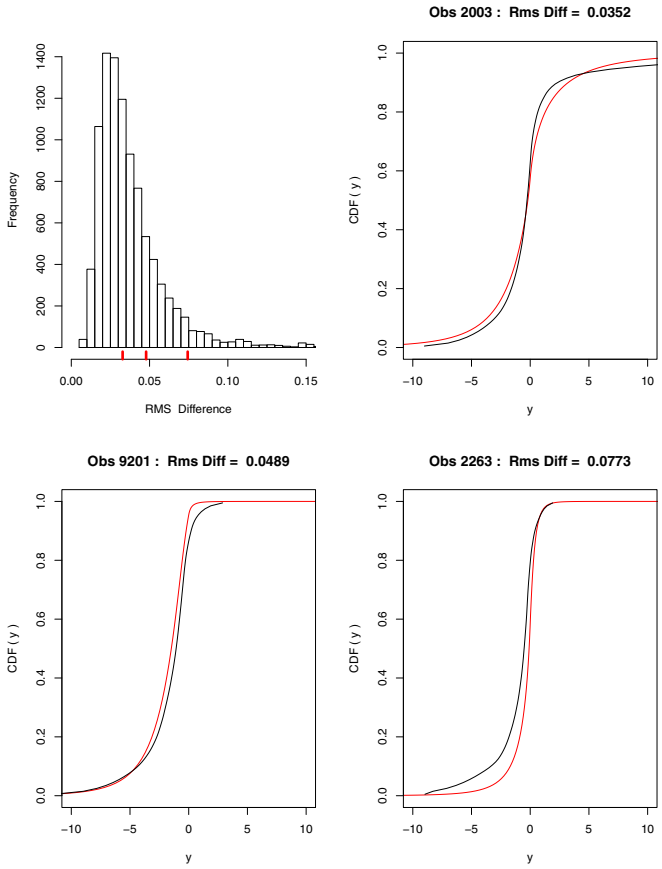

Fig. 10. (*Upper Left*) CDF error Eq. **22** distribution for simulated data. (*Upper Right*) Estimated (black) and true (red) CDFs for observation with median error. (*Lower*) Corresponding plots for 75% and 90% decile errors.

distributions $\hat{S}(\mathbf{x}) = S[\,\hat{p}_y(y\,|\,\mathbf{x})]$. Distribution contrast trees can be constructed in the presence of arbitrary censoring or truncation. This extends to contrast boosted distribution estimates $\hat{p}_y(y\,|\,\mathbf{x})$ and any quantities derived from them. This in turn allows application to ordinal regression which can be considered a special case of interval censoring (9).

Discussion

When a discrepancy measure takes the special form of an average over individual observation losses, such as Eq. **5** or model residuals Eq. **15**, one can use an ordinary regression tree (or other standard learning methods) to directly model the discrepancy as a function of **x**. This may uncover **x** values corresponding to relatively high discrepancy. However, such a strategy is not focused on this task but rather on trying to approximate discrepancy over entire distribution of **x** values. The contrast tree splitting strategy Eq. **4** directly seeks high-discrepancy regions regardless of local data density, thereby largely ignoring the **x** distribution. Besides increased sensitivity to high discrepancy, this property has the additional effect of rendering contrast tree-based methods more robust against distribution drift. Standard leaning methods are not applicable to discrepancy measures that are *not* simple averages of single observation loss criterion, such as Eqs. **6**–**8**.

The fitting paradigm of contrast trees is somewhat different from that of ordinary machine learning. The goal of the latter is data fitting, that is, to capture as much structure as possible in the relation between y and **x**. The more structure captured the better the model, subject to overfitting considerations. Overfitting occurs when the model captures nongeneralizable data-specific relationships. Contrast trees attempt to uncover lack of fit. The more structure they capture, the worse the model fits the data.

This reversal of emphasis has consequences for interpretation. With regular machine learning evaluating the quality of a model on its own training data generally produces an overoptimistic measure of model quality. With contrast trees this gives a conservative overly pessimistic assessment of model accuracy, especially for large trees built with small samples. For small trees and/or large samples the effect is usually small. Using different data to construct the tree and evaluate its node statistics eliminates this bias at the cost of increased variance.

Related Work

Regression trees have a long history in statistics and machine learning. Since their first introduction (10) many proposed modifications have been introduced to increase accuracy and extend applicability. See ref. 11 for a nice survey. More recent extensions include MediBoost (12) and the additive tree (13). All of these proposals are focused toward estimating the properties of a single outcome variable. There has been work on using trees for simultaneous estimation of several outcome variables (14) but there seems to have been little to no work related to applications involving contrasting two such variables.

Although not directly involving trees, Friedman and Fisher (15) proposed using recursive partitioning strategies to identify interpretable regions in **x** space within which the mean of a single outcome y was relatively large ("hot spots"). With a similar

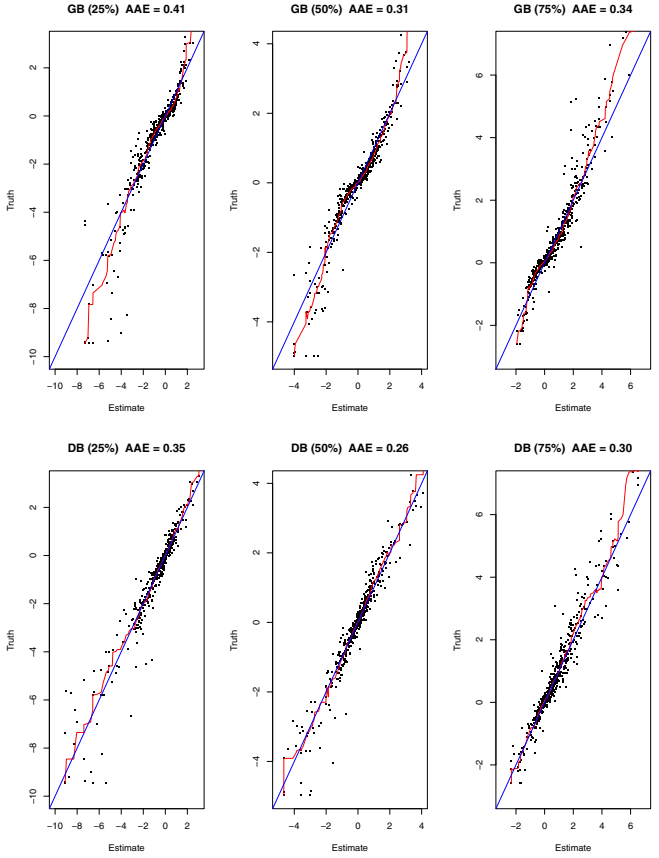

Fig. 11. Predicted versus true values for the three quartiles as functions of **x** (columns) for gradient boosting quantile regression (*Upper*) and distribution boosting (*Lower*) on the simulated data. The red lines represent a running median smooth and the blue lines show equality.

goal Buja and Lee (16) proposed using ordinary regression trees with a splitting criterion based on the maximum of the two daughter-node means.

Classification tree boosting was proposed by Freund and Schapire (17). Extension to regression trees was developed by Friedman (4). Since then there has been considerable research attempting to improve accuracy and extend its scope. See ref. 18 for a good summary.

Although boosted contrast trees have not been previously proposed they are generally appropriate for the same types of applications as gradient-boosted regression trees, such as classification, regression, and quantile regression. They can be beneficial in applications where a contrast tree indicates lack of fit of a model produced by some estimation method. In such situations applying contrast boosting to the model predictions often provides improvement in accuracy.

Tree ensembles have also been applied to nonparametric conditional distribution estimation. Meinshausen (19) used classic random forests to define local neighborhoods in $\mathbf{x}$ space. The empirical conditional distribution of y in each such defined local region around a prediction point $\mathbf{x}$ is taken as the corresponding conditional distribution estimate at $\mathbf{x}$. Athey et al. (20) noted that since the regression trees used by random forests are designed to detect only mean differences the resulting neighborhoods will fail to adequately capture distributions for which higher moments are not generally functions of the mean. They proposed modified tree-building strategies based on gradient-boosting ideas to customize random-forest tree construction for specific applications including quantile regression.

Boosted regression trees have been used as components in procedures for parametric fitting of conditional distributions and transformations. A parametric form for the conditional distribution or transformation is hypothesized and the parameters as functions of $\mathbf{x}$ are estimated by regression tree gradient boosting using negative log likelihood as the prediction risk [see, e.g., Mayr et al. (21), Friedman (9), Pratola et al. (22), Hothorn (23), and Mukhopadhyay and Wang (24)]. Some differences between these previous methods and the corresponding approaches proposed here include use of contrast rather than regression trees, and no parametric assumptions.

The principal benefit of the contrast tree-based procedures is a lack-of-fit measure. As seen in Table 1 and in *SI Appendix*, values of negative log likelihoods or prediction risk need not reflect actual lack of fit to the data. The values of their minima can depend upon other unmeasured quantities. The goal of contrast trees as illustrated in this paper is to provide such a measure. Contrast trees can be applied to assess lack of fit of estimates produced by any method, including those mentioned above. If discrepancies are detected, contrast boosting can be employed to remedy them and thereby improve accuracy.

Summary

Contrast trees are designed to provide interpretable goodness-of-fit diagnostics for estimates of the parameters of $p_y(y \mid \mathbf{x})$, or the full distribution. Examples involving classification, probability estimation, and conditional distribution estimation were presented. A quantile regression example is presented in *SI Appendix*. Two-sample contrast trees for detecting discrepancies between separate datasets are also described in *SI Appendix*.

Boosting of contrast trees is a natural extension. Given an initial estimate $\hat{z}(\mathbf{x})$ from any learning method a contrast tree can assess its goodness or lack of fit to the data. If found lacking, the boosting strategy attempts to improve the fit by successively modifying $\hat{z}(\mathbf{x})$ to bring it closer to the data. As seen in Fig. 4 this strategy can substantially improve prediction accuracy for some methods. *SI Appendix* provides such an example involving quantile regression.

Contrast boosting the full conditional distribution is illustrated on simulated data and on actual data in *SI Appendix*. Note that the conditional distribution procedure can be applied in the presence of arbitrarily censored or truncated data by employing Turnbull's (3) algorithm to compute CDFs and corresponding quantiles.

Contrast trees and boosting inherit all of the data analytic advantages of classification and regression trees. These include handling categorical variables and missing values, invariance to monotone transformations of the predictor variables, resistance to irrelevant predictors, variable importance measures, and few tuning parameters.

Materials and Methods

The data used in the examples was obtained online from the UCI Machine Learning repository (25). An R package (conTree) implementing the methods described herein is available from statweb.stanford.edu/~jhf/conTree.

ACKNOWLEDGMENTS. Important discussions with Trevor Hastie and Rob Tibshirani on the subject of this work are gratefully acknowledged.

1. L. Breiman, J. Friedman, R. Olshen, C. Stone, *Classification and Regression Trees* (Chapman & Hall, 1984).
2. T. Anderson, D. Darling, Asymptotic theory of certain "goodness-of-fit" criteria based on stochastic processes. *Ann. Stat.* **23**, 193–212 (1952).
3. B. W. Turnbull, The empirical distribution function with arbitrarily grouped, censored and truncated data. *J. Roy. Stat. Soc. B* **38**, 290–295 (1976).
4. J. Friedman, Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **29**, 1189–1232 (2001).
5. R. Kohavi, "Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid" in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, E. Simoudis, J. Han, U. Fayyad, Eds. (Assoc. for Computing Machinery, New York, 1996), pp. 202–207.
6. L. Breiman, Random forests. *Mach. Learn.* **45**, 5–32 (2001).
7. B. Efron, R. Tibshirani, *An Introduction to the Bootstrap* (Springer, 1994).
8. K. Fernandes, P. Vinagre, P. Cortez, "A proactive intelligent decision support system for predicting the popularity of online news" in *Progress in Artificial Intelligence*, F. Pereira, P. Machado, E. Costa, A. Cardoso, Eds. (Lecture Notes in Computer Science, Springer, Cham, 2015), vol. 9273, pp. 535–546.
9. J. Friedman, Predicting regression probability distributions with imperfect data through optimal transformations. arXiv:2001.10102v1 (27 January 2020).
10. J. Morgan, J. Sonquist, Problems in the analysis of survey data, and a proposal. *J. Am. Stat. Assoc.* **58**, 415–434 (1963).
11. W. Loh, Fifty years of classification and regression trees. *Int. Stat. Rev.* **82**, 329–348 (2014).
12. G. Valdes *et al.*, MediBoost: A patient stratification tool for interpretable decision making in the era of precision medicine. *Sci. Rep.* **6**, 37854 (2016).
13. J. Luna *et al.*, The additive tree. *Proc. Natl. Acad. Sci. U.S.A.* **116**, 19887–19893 (2019).
14. M. Segal, Y. Xiao, Multivariate random forests. WIREs Data Mining and Knowledge Discovery **1**, 80–87 (2011).
15. J. Friedman, N. Fisher, Bump hunting in high-dimensional data. *Stat. Comput.* **9**, 123–143 (1999).
16. A. Buja, Y. Lee, Data mining criteria for tree-based regression and classification. *Proc. KDD* **2001**, 27–36 (2001).
17. Y. Freund, R. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **55**, 119–139 (1997).
18. A. Mayr, H. Binder, O. Gefeller, M. Schmid, The evolution of boosting algorithms from machine learning to statistical modelling. *Methods Inf. Med.* **53**, 419–427 (2014).
19. M. Meinshausen, Quantile random forests. *J. Mach. Learn. Res.* **7**, 983–999 (2006).
20. S. Athey, J. Tibshirani, S. Wagner, Generalized random forests. *Ann. Stat.* **47**, 1148–1178 (2019).
21. A. Mayr, N. Fenske, B. Hofner, T. Kneib, M. Schmid, GAMLSS for high dimensional data—A flexible approach based on boosting. *J. R. Stat. Soc. Ser. C (Appl. Stat.)* **61**, 403–427 (2012).
22. M. T. Pratola, H. A. Chipman, E. I. George, R. E. McCulloch, Heteroscedastic BART via multiplicative regression trees. *J. Comput. Graph Stat.*, 10.1080/10618600.2019.1677243 (2019).
23. T. Hothorn, Transformation boosting machines. *Stat. Comput.* **30**, 141–152 (2019).
24. S. Mukhopadhyay, K. Wang, On the problem of relevance in statistical inference. *J. Am. Stat. Assoc.* (2019).
25. D. Dua, C. Graff, UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets.php. Accessed 7 August 2020.