Mach Learn (2015) 101:325–343
DOI 10.1007/s10994-014-5452-1

Two-level quantile regression forests for bias correction in range prediction

Thanh-Tung Nguyen · Joshua Z. Huang · Thuy Thi Nguyen

Received: 26 December 2013 / Accepted: 28 May 2014 / Published online: 19 July 2014

Abstract Quantile regression forests (QRF), a tree-based ensemble method for estimation of conditional quantiles, has been proven to perform well in terms of prediction accuracy, especially for range prediction. However, the model may have bias and suffer from working with high dimensional data (thousands of features). In this paper, we propose a new bias correction method, called bcQRF that uses bias correction in QRF for range prediction. In bcQRF, a new feature weighting subspace sampling method is used to build the first level QRF model. The residual term of the first level QRF model is then used as the response feature to train the second level QRF model for bias correction. The two-level models are used to compute bias-corrected predictions. Extensive experiments on both synthetic and real world data sets have demonstrated that the bcQRF method significantly reduced prediction errors and outperformed most existing regression random forests. The new method performed especially well on high dimensional data.

Editors: Vadim Strijov, Richard Weber, Gerhard-Wilhelm Weber, and Süreyya Ozogur Akyüz .

T.-T. Nguyen · J. Z. Huang
Shenzhen Key Laboratory of High Performance Data Mining, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
e-mail: tungnt@wru.vn; tungnt@siat.ac.cn

T.-T. Nguyen
School of Computer Science and Engineering, Water Resources University, Hanoi, Vietnam

T.-T. Nguyen
University of Chinese Academy of Sciences, Beijing 100049, China

J. Z. Huang (✉)
College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
e-mail: zx.huang@szu.edu.cn

T. T. Nguyen
Faculty of Information Technology, Vietnam National University of Agriculture, Hanoi, Vietnam
e-mail: ntthuy@vnua.edu.vn

Keywords Bias correction · Random forests · Quantile regression forests · High dimensional data · Data mining

1 Introduction

Random forests (RF) (Breiman 2001) is a non-parametric regression method that builds an ensemble model of regression trees from random subsets of features and bagged samples of the training data. Given a training data set:

$$\mathcal{L} = \left\{ (X_i, Y_i)_{i=1}^{N} \mid X_i \in \mathbb{R}^M, Y \in \mathbb{R}^1 \right\},$$

where N is the number of training samples (also called objects) and M is the number of features, a regression RF independently and uniformly samples with replacement the training data $\mathcal{L}$ to draw a bootstrap data set $\mathcal{L}_k^*$ from which a regression tree T_k^* is grown. Repeating this process for K replicates produces K bootstrap data sets and K corresponding regression trees $T_1^*, T_2^*, \ldots, T_K^*$ which form a regression RF.

Given an input $X = x$, a regression RF is used as a function $f : \mathbb{R}^M \to \mathbb{R}^1$ to estimate the unknown value y of input $x \in \mathbb{R}^M$, denoted as $\hat{f}(x)$. Write the regression RF in the common regression form

$$Y = f(X) + \varepsilon, \tag{1}$$

where $E(\varepsilon) = 0$, $Var(\varepsilon) = \sigma_\varepsilon^2$. The function $f(\cdot)$ is estimated from $\mathcal{L}$ and the prediction $\hat{f}(x)$ is obtained from an independent test case x.

For point regression with a regression RF, each tree T_k gives a prediction $\hat{f}_k(x)$ and the predictions of all trees are averaged to produce the final RF prediction

$$\hat{f}(x) = \frac{1}{K} \sum_{k=1}^{K} \hat{f}_k(x). \tag{2}$$

This is the estimation of $f(x) = E(Y|X = x)$. The mean-squared error of the prediction measures the effectiveness of $\hat{f}$, defined as (Hastie et al. 2009)

$$\begin{aligned} Err(x) &= E\left[(Y - \hat{f}(x))^2 | X = x\right] \\ &= \sigma_\varepsilon^2 + \left[E\hat{f}(x) - f(x)\right]^2 + E\left[\hat{f}(x) - E\hat{f}(x)\right]^2 \\ &= \sigma_\varepsilon^2 + Bias^2(\hat{f}(x)) + Var(\hat{f}(x)) \\ &= Irreducible\ Error + Bias^2 + Variance. \end{aligned} \tag{3}$$

The first term is the variance of the target around its true mean $f(x)$. This cannot be avoided no matter how well $\hat{f}(x)$ is estimated, unless $\sigma_\varepsilon^2 = 0$. The second term is the squared bias and the last term is the variance. The last two terms need to be addressed for a good performance of the prediction model.

Given an input object x, a regression RF predicts a value in each leaf node which is the mean of Y values of the objects in that leaf node. This value can be biased because large and small values in the objects of the leaf node are often underestimated or overestimated. The prediction accuracy can be improved if the median is used (instead of the mean) as the prediction and the median surpasses the mean in robustness towards extreme values/outliers.

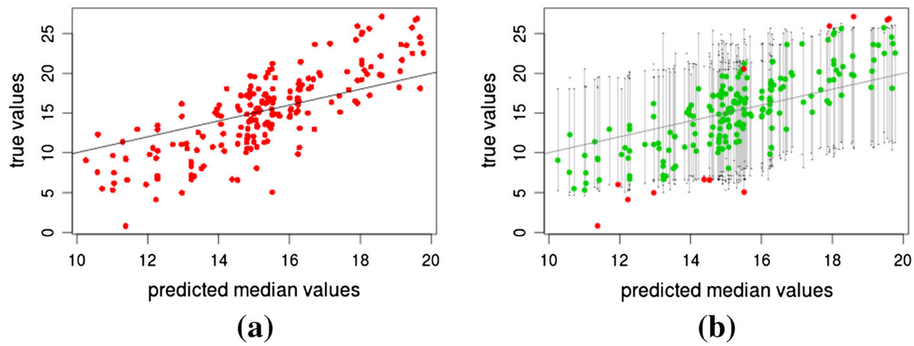


Fig. 1 The predicted median values from the synthetic data set generated by the model of Eq. (4) show the biases of Y values. The *solid line* connects the points where the predicted values and the true values are equal. A large number of points escape from the *solid line*. (**a**) Bias in point prediction, (**b**) The 90 % range prediction

Meinshausen (2006) proposed quantile regression forests (QRF) for both point and range prediction. QRF uses the median in point regression. For range prediction, QRF requires the estimated distribution of $F(y|X=x) = P(Y < y|X=x)$ at each leaf node, not only the mean. Given two quantile probabilities α_l and α_h, QRF predicts the range $[Q_{\alpha_l}(x), Q_{\alpha_h}(x)]$ of Y with a given probability τ that

$$P(Q_{\alpha_l}(x) < Y < Q_{\alpha_h}(x)|X=x) = \tau.$$

Besides range prediction, QRF also performs well in situations where the conditional distributions are not Gaussian. However, similar to regression RF, QRF can still be biased in point prediction even though the median is used instead of the mean in prediction.

To illustrate this kind of bias, we generated 200 objects as a training data set and 1000 objects as the testing data set using the following model:

$$Y = 10\sin(\pi X_1 X_2) + 20(X_3 - 0.5)^2 + 10X_4 + 5X_5 + \epsilon, \tag{4}$$

where X_1, X_2, X_3, X_4, X_5 and ϵ are from $U(0, 1)$.

We ran the QRF program in R package (Meinshausen 2012) on the generated data with the default settings. Figure 1 shows the predicted median values against the true values for point regression and range prediction. The bias in the point estimates is large when the true values are small or big. In case of range prediction, we can see that the predicted values are unevenly distributed in the range area of quantiles represented in the grey bars.

It is known that the performance of both regression random forests and quantile regression forests suffers when applied to high dimensional data, i.e., data with thousands of features. The main cause is that in the process of growing a tree from the bagged samples, the subset of features randomly sampled from thousands of features in $\mathcal{L}$ to split a node of the tree is often dominated by less important features. The tree grown from such a subspace features has a low accuracy in prediction, which affects the final prediction of the random forests.

Breiman (1996) introduced bagging in RF as a way to reduce the prediction variance and increase the accuracy of the prediction. However, the bias problem remained. In his later work (Breiman 2001), an iterative bagging algorithm was developed to reduce both variance and bias in general prediction problems. However, this iterative bagging approach was not well understood in applications to improve RF predictions (Xu 2013). Recently, Zhang and Yan (2012) proposed five techniques for using RFs to estimate the regression functions. They considered that the bias of the model is related to both the predictor features and the response feature. A simple non-iterative approach was introduced to use a regular RF to correct the bias in regression models. The results were compared favorably to other bias-correction approaches. However, their approach can only be applied to point prediction. Moreover, the mean values were used in predictions at leaf nodes, which, as mentioned before, could suffer from extreme values in data. Besides, the techniques were tested only on small low dimensional data sets with the number of features less than or equal to 13. Xu (2013) proposed a bias correction method in random forests which corrects the bias of RFs using a second RF (Liaw and Wiener 2002). They demonstrated that the new approach performed better in de-biasing and improving RF predictions than a standard de-biasing technique in the R-package *randomForest*. They also proposed a generalized version of iterative bias correction in RFs by applying a similar bias correction when predicting the out-of-bag bias estimates from RF, and showed that predictions on some data sets may be improved by more iterations of bias correction.

In this paper, we propose a new bias correction method called bcQRF to correct the bias in QRF models. The bcQRF method is based on the QRF model to correct the bias in regression models instead of the adaptive bagging proposed by Breiman (1999). bcQRF consists of two levels of QRF models. In the first level model, a new subspace feature weighting sampling method is used to grow trees for regression random forests. Given a training data set $\mathcal{L}$, we first use a feature permutation method to measure the importance of features and produce raw feature importance scores. Then, we apply p-value assessment to separate important features from the less important ones and partition the set of features in $\mathcal{L}$ into two subsets, one containing important features and one containing less important features. We independently sample features from the two subsets and put them together as a new feature subspace for splitting the data at a node. Since the subspace always contains important features which can guarantee a better split at the node, this subspace feature weighting sampling method enables generating trees from the bagged sample data with smaller regression errors.

After the first QRF model is built, the residual value is used to replace the response feature of the original training data set and the second level QRF model is built to estimate the bias values of the first level QRF model. The bias-corrected values are computed based on the difference between the values predicted by the first level QRF model and the second level QRF model. With bcQRF, both point regression bias and range prediction bias can be corrected. Our experimental results on both synthetic and real world data sets have shown that the proposed algorithm with these bias-correction techniques dramatically reduced the prediction errors and outperformed existing regression random forests models.

2 Random forests for regression

2.1 Regression random forests

Given a training data $\mathcal{L}$, a regression random forests model is built as follows.

- Step 1: Draw a subset of samples $\mathcal{L}_k$ from $\mathcal{L}$ using bagging (Breiman 1996, 1999), i.e., sampling with replacement.
- Step 2: Grow a regression tree T_k from $\mathcal{L}_k$. At each node t, the split is determined by the decrease in impurity that is defined as $\sum_{x_i \in t}(Y_i - \bar{Y}_t)^2/N(t)$, where $N(t)$ is the number of objects and $\bar{Y}_t$ is the mean value of all Y_i at node t. At each leaf node, $\bar{Y}_t$ is assigned as the prediction value of the node.

– Step 3: Let $\hat{Y}^k$ be the prediction of tree T_k given input X. The prediction of regression random forests with K trees is

$$\hat{Y} = \frac{1}{K}\sum_{k=1}^{K}\hat{Y}^k.$$

Since each tree is grown from a bagged subset of samples, it is grown with only two-third of objects in $\mathcal{L}$. About one-third of objects are left out and these objects are called *out-of-bag (OOB)* samples which are used to estimate the prediction errors (Breiman 1996, 2001; Breiman et al. 1984).

2.2 Quantile regression forests

Quantile regression forests (QRF) uses the same steps as used in regression random forests to grow trees (Meinshausen 2006). However, at each leaf node, it retains all Y values instead of only the mean of Y values. Therefore, QRF keeps a raw distribution of Y values at each leaf node.

Using the notations by Breiman (2001), let θ_k be the random parameter vector that determines the growth of the kth tree and $\Theta = \{\theta_k\}_1^K$ be the set of random parameter vectors for the forests generated from $\mathcal{L}$. In each regression tree T_k from $\mathcal{L}_k$, we compute a positive weight $w_i(x_i, \theta_k)$ for each case $x_i \in \mathcal{L}$. Let $l(x, \theta_k, t)$ be a leaf node t in T_k. The cases $x_i \in l(x, \theta_k, t)$ are assigned to the same weight $w_i(x, \theta_k) = 1/N(t)$, where $N(t)$ is the number of cases in $l(x, \theta_k, t)$. In this way, all cases in $\mathcal{L}_k$ are assigned positive weights and the cases not in $\mathcal{L}_k$ are assigned weight zero.

For a single tree prediction, given $X = x$, the prediction value is

$$\hat{Y}^k = \sum_{i=1}^{N} w_i(x, \theta_k)Y_i = \sum_{x, X_i \in l(x,\theta_k,t)} w_i(x, \theta_k)Y_i. \tag{5}$$

The weight $w_i(x)$ assigned by random forests is the average of weights by all trees, that is

$$w_i(x) = \frac{1}{K}\sum_{k=1}^{K} w_i(x, \theta_k). \tag{6}$$

The prediction of regression random forests is

$$\hat{Y} = \sum_{i=1}^{N} w_i(x)Y_i. \tag{7}$$

We note that $\hat{Y}$ is the average of conditional mean values of all trees in the regression random forests.

Given an input $X = x$, we can find the leaf nodes $l_k(x, \theta_k)$ from all trees where X falls and the set of Y_i in these leaf nodes. Given all Y_i and the corresponding weights $w(i)$, we can estimate the conditional distribution function of Y given X as

$$\hat{F}(y|X = x) = \sum_{i=1}^{N} w_i(x)\mathcal{I}(Y_i \leq y), \tag{8}$$

where $\mathcal{I}(\cdot)$ is the indicator function that is equal to 1 if $Y_i \leq y$ and 0 otherwise. Given a probability α, we can estimate the quantile $Q_\alpha(X)$ as

$$\hat{Q}_\alpha(X = x) = \inf \left\{ y : \hat{F}(y|X = x) \geq \alpha \right\}. \tag{9}$$

For range prediction, we have

$$[Q_{\alpha_l}(X), Q_{\alpha_h}(X)] = \left[\inf \left\{ y : \hat{F}(y|X = x) \geq \alpha_l \right\}, \inf \left\{ y : \hat{F}(y|X = x) \geq \alpha_h \right\} \right], \tag{10}$$

where $\alpha_l < \alpha_h$ and $(\alpha_h - \alpha_l) = \tau$. Here, τ is the probability that prediction Y will fall in the range of $[Q_{\alpha_l}(X), Q_{\alpha_h}(X)]$.

For point regression, the prediction can be a value in the range, such as the mean or median of Y_i values. The median surpasses the mean in robustness towards outliers. We use the median of Y values in the range of two quantiles as the prediction of Y given input $X = x$.

3 Feature weighting for subspace selection

3.1 Importance measure of features by permutation

Given a training data set $\mathcal{L}$ and a regression random forests model RF, Breiman (2001) described a permutation method to measure the importance of features in the prediction. The procedure for computing the importance scores of features consists of following steps.

1. Let $\mathcal{L}_k^{oob}$ be the *out-of-bag* samples of the kth tree T_k. Given $X_i \in \mathcal{L}_k^{oob}$, use T_k to predict $\hat{Y}_i^k$, denoted as $\hat{f}_i^k(X_i)$.
2. Choose a predictor feature j and randomly permute the value of feature j of case X_i with the value of another case in $\mathcal{L}_k^{oob}$. Use tree T_k to obtain a new prediction denoted as $\hat{f}_i^{k,p,j}(X_i)$ on the permuted X_i where p is the index of permutations. Repeat the permutation process P times.
3. For K_i trees grown without X_i, compute the out-of-bag prediction by RF in the pth permutation of the jth predictor feature as

$$\hat{f}_i^{p,j}(X_i) = \frac{1}{K_i} \sum_{X_i \in \mathcal{L}_k^{oob}} \hat{f}_i^{k,p,j}(X_i).$$

4. Compute the two *mean square residuals* (MSR) with and without permutations of predictor feature j on X_i as

$$MSR_i = \frac{1}{K_i} \sum_{k \in K_i} \left[\hat{f}_i^k(X_i) - Y_i \right]^2$$

and

$$MSR_i^j = \frac{1}{P} \sum_{p=1}^{P} \left[\hat{f}_i^{p,j}(X_i) - Y_i \right] \text{respectively.}$$

5. Let $\Delta MSR_i^j = max\left(0, MSR_i^j - MSR_i\right)$. The importance of feature j is

$$IMP_j = \frac{1}{N} \sum_{i \in \mathcal{L}} \Delta MSR_i^j.$$

Table 1 The importance scores matrix of all predictor features and shadow features with R replicates

Iter.	VI_{X_1}	VI_{X_2}	$\dots$	VI_{X_M}	$VI_{A_{M+1}}$	$VI_{A_{M+2}}$	$\dots$	$VI_{A_{2M}}$
1	$VI_{x_{1,1}}$	$VI_{x_{1,2}}$	$\dots$	$VI_{x_{1,M}}$	$VI_{a_{1,(M+1)}}$	$VI_{a_{1,(M+2)}}$	$\dots$	$VI_{a_{1,2M}}$
2	$VI_{x_{2,1}}$	$VI_{x_{2,2}}$	$\dots$	$VI_{x_{2,M}}$	$VI_{a_{2,(M+1)}}$	$VI_{a_{2,(M+2)}}$	$\dots$	$VI_{a_{2,2M}}$
$\vdots$	$\vdots$							$\vdots$
R	$VI_{x_{R,1}}$	$VI_{x_{R,2}}$	$\dots$	$VI_{x_{R,M}}$	$VI_{a_{R,(M+1)}}$	$VI_{a_{R,(M+2)}}$	$\dots$	$VI_{a_{R,2M}}$

To normalize the importance measures, we have the raw importance score as

$$VI_j = \frac{IMP_j}{\sum_{l=1}^{l=M} IMP_l}, \tag{11}$$

where M is the total number of features in $\mathcal{L}$. We can rank the features on the raw importance scores according to Eq. (11).

3.2 p-value feature assessment

The permutation method only gives the importance ranking of features. However, for better feature selection at each node of a tree, we need to separate important features from less important ones. This can be done with Welch's two-sample t-test (Welch 1947) that compares the importance score of a feature with the maximum importance score of the generated noisy features called shadows. The shadow features do not have prediction power to the response feature. Therefore, any feature whose importance score is smaller than the maximum importance score of the noisy features is considered as less important. Otherwise, it is considered as important. This idea was introduced by Stoppiglia et al. (2003) and further developed in Kursa and Rudnicki (2010), Tuv et al. (2006, 2009), Tung et al. (2014), Sandri and Zuccolotto (2008, 2010).

We build a random forests model RF from this extended data set with shadow features. Following the importance measure by the permutation procedure, we use RF to compute $2M$ importance scores for $2M$ features. We repeat the same process R times to compute R replicates. Table 1 illustrates the importance measures of M input features and M shadow features by permutating the values of the corresponding features in the data.

From the replicates of shadow features, we extract the maximum value from each row and put it into the comparison sample $V^* = max\{A_{rj}\}$, $(r = 1, ..R; j = M+1, ..2M)$. For each input feature X_j, we compute t-statistic as:

$$t_j = \frac{\overline{VI}_{X_j} - \overline{V}^*}{\sqrt{s_1^2/n_1 + s_2^2/n_2}}, \tag{12}$$

where s_1^2 and s_2^2 are the unbiased estimators of the variances of the two samples, $\overline{VI}_{X_j}$ is the average of R importance scores on the jth input feature and $\overline{V}^*$ is the average of R comparison values in V^*. For significance test, the distribution of t_j in Eq. (12) is approximated as an ordinary Student's distribution with the degree of freedom df calculated as

$$df = \frac{\left(s_1^2/n_1 + s_2^2/n_2\right)^2}{\left(s_1^2/n_1\right)^2/(n_1-1) + \left(s{^2}_2/n_2\right)^2/(n_2-1)} \tag{13}$$

where $n_1 = n_2 = R$.

Having computed the t statistic and df, we can compute the p-value for the feature and perform hypothesis test on $\overline{VI}_{X_j} > \overline{V}^*$. Given a statistical significance level, we can identify important features. This test confirms that if a feature is important, it consistently scores higher than the shadow over multiple permutations.

3.3 Feature partition and subspace selection

The p-value of a feature indicates the importance of the feature in prediction. The smaller the p-value of a feature, the more correlated the predictor feature to the response feature, and the more powerful the feature in prediction.

Given all p values for all features, we set a significance level as the threshold λ, for instance $\lambda = 0.05$. Any feature whose p-value is smaller than λ is added to the important feature subset X_{high}, and otherwise it is added to the less important feature subset X_{low}. The two subsets partition the set of features in data. Given X_{high} and X_{low} at each node, we randomly select some features from X_{high} and some from X_{low} to form the feature subspace for splitting the node. Given a subspace size, we can form the subspace with 80 % of features sampled from X_{high} and 20 % sampled from X_{low}.

4 Bias correction algorithm

4.1 A new quantile regression forests algorithm

Now we can extend the quantile regression forests with the new feature weighting subspace sampling method to generate splits at the nodes of decision trees and select prediction value of Y from the range of low and high quantiles with high probability. The new quantile regression forests algorithm eQRF is summarized as follows.

1. Given $\mathcal{L}$, generate the extended data set $\mathcal{L}^e$ in $2M$ dimensions by permutating the corresponding predictor feature values to generate shadow features.
2. Build a regression random forests model RF^e from $\mathcal{L}^e$ and compute R replicates of raw importance scores of all predictor features and shadows with RF^e. Extract the maximum importance score of each replicate to form the comparison sample V^* of R elements.
3. For each predictor feature, take R importance scores and compute t statistic according to Eq. (12).
4. Compute the degree of freedom df according to Eq. (13).
5. Given t statistic and df, compute all p-values for all predictor features.
6. Given a significance level threshold λ, separate important features from less important ones in two subsets X_{low} and X_{high}.
7. Sample the training set $\mathcal{L}$ with replacement to generate bagged samples $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_K$.
8. For each sample set $\mathcal{L}_k$, grow a regression tree T_k as follows:

   (a) At each node, select a subspace of $mtry$ ($mtry > 1$) features randomly and separately from X_{low} and X_{high} and use this subspace features as candidates for splitting the node.

   (b) Each tree is grown nondeterministically, without pruning until the minimum node size n_{min} is reached. At each leaf node, all Y values of the objects in the leaf node are kept.
   (c) Compute the weights of each X_i by individual trees and the forests with out-of-bag samples.

9. Given a probability τ, α_l and α_h for $\alpha_h - \alpha_l = \tau$, compute the corresponding quantile Q_{α_l} and Q_{α_h} with Eq. (10) (we set default values $[\alpha_l = 0.05, \alpha_h = 0.95]$ and $\tau = 0.9$).
10. Given an input X, estimate the prediction value from a value in the quantile range of Q_{α_l} and Q_{α_h} such as mean or median.

4.2 Two-level bias-correction algorithm

Breiman described an adaptive bagging method (Breiman 1999) as a stage-wise iterative process. Consider Y values in the first stage and denote $\hat{Y}$ as the predicted values which are calculated by subtracting the predictors, the second stage of bagging is carried out using $\hat{Y}$ values. He suggested that the iteration should stop if the mean squared errors for new cases from the next stage are 1.1 times of the minimal errors calculated so far. Consequently, the residuals $Y - \hat{Y}$ at the second stage will bring extra variance. This means that adding more iterative stages will lead to bias which tends to zero, while the variance will keep increasing. Thus, addressing more than two stages is not necessary.

We propose a two-level bias-correction algorithm bcQRF to correct the prediction bias, instead of Breiman's approach. The first level quantile regression forests model is built from the training data. The prediction errors from the first level QRF model replace the values of the response feature in the original training data. The new training data with the prediction errors as the response feature is used to build the second level quantile regression forests model. The final bias-corrected values are calculated as the prediction value of the first level model minus the prediction value of the second level model.

The bcQRF algorithm in range prediction is summarized as follows.

- Step 1: Grow the first level QRF model from the training data $\mathcal{L}$ with response feature Y.
- Step 2: Obtain the predicted quantile values $\hat{Q}_\alpha(X = x)$ of x from the training data. Estimate the bias as the median of the predicted values in the quantiles minus the true response value of input data, defined as

$$\widetilde{E} = \hat{Q}_{0.5}(X = x) - Y. \tag{14}$$

- Step 3: Given $X = x_{new}$, use the first level QRF model to produce the quantile values and the range $[Q_{\alpha_l}(X = x_{new}), Q_{\alpha_h}(X = x_{new})]$.
- Step 4: Extend the training data set $\mathcal{L}$ with the bias errors $\widetilde{E}$ as a new response feature to generate an extended data set $\mathcal{L}^e = \{\mathcal{L}, \widetilde{E}\}$. Grow the second level QRF model from $\mathcal{L}^e$ with the response feature $\widetilde{E}$. Use the second level QRF model to predict the training data and obtain a new set of errors $\widetilde{E}_{new}$.
- Step 5: The bias-corrected quantiles are computed as

$$\left[\hat{Q}\alpha_{l new}, \hat{Q}\alpha_{h new}\right] = \left[Q_{\alpha_l}(X = x_{new}) - \widetilde{E}_{new}, Q_{\alpha_h}(X = x_{new}) - \widetilde{E}_{new}\right]. \tag{15}$$

For point prediction, the predicted values are chosen as $\hat{Q}_{0.5}$.

5 Experiments and evaluations

5.1 Data sets

5.1.1 Synthetic data sets

We have defined Model 1 in Eq. (4) for synthetic data generation. Here, we define Model 2 as follows.

$$Y = 0.1e^{4X_1} + \frac{4}{1 + e^{-20(X_2 - 0.5)}} + 3X_3 + 2X_4 + X_5 + \epsilon, \tag{16}$$

where $\epsilon \sim N(0, 1.5^2)$ and 5 *iid* predictor features were from $U(0, 1)$. The two models were used in Friedman (1991) to generate data sets with multiple non-linear interactions between predictor features. Each model has 5 predictor features. In generating a synthetic data set, we first used a model to create 200 objects in 5 dimensions plus a response feature and then expanded the 200 objects with five noisy features. Two data sets were generated with the two models and saved in files $\mathcal{L}M10_1$ and $\mathcal{L}M10_2$ where the subscripts indicate the models used to generate the data, $M10$ indicates the number of dimensions and $\mathcal{L}$ is the training data. In the same way, we also generated two test data sets, $\mathcal{H}M10_1$ and $\mathcal{H}M10_2$ where $\mathcal{H}$ indicates test data. Each test data set contained 1000 objects.

To investigate the effect of irrelevant or noisy features on prediction errors, we used Model 1 of Eq. (4) to generate two groups of data sets with three different dimensions $\{M5, M20, M50\}$ and three noise levels $\sigma = 0.1, 1$ and 5. Totally, we created 9 synthetic training data sets as $\{\mathcal{L}M5S0.1$, $\mathcal{L}M5S1$, $\mathcal{L}M5S5$, $\mathcal{L}M20S0.1$, $\mathcal{L}M20S1$, $\mathcal{L}M20S5$, $\mathcal{L}M50S0.1$, $\mathcal{L}M50S1$, $\mathcal{L}M50S5\}$, where S indicates a noise level. Each data set contained 200 objects. In the same way, we created 9 test sets $\{\mathcal{H}M5S0.1$, $\mathcal{H}M5S1$, $\mathcal{H}M5S5$, $\mathcal{H}M20S0.1$, $\mathcal{H}M20S1$, $\mathcal{H}M20S5$, $\mathcal{H}M50S0.1$, $\mathcal{H}M50S1$, $\mathcal{H}M50S5\}$. Finally, we used Model 1 to generate 3 pairs of high dimensional data sets $\{\mathcal{L}M200S5$, $\mathcal{H}M200S5$, $\mathcal{L}M500S5$, $\mathcal{H}M500S5$, $\mathcal{L}M1000S5$, $\mathcal{H}M1000S5\}$ to evaluate the new algorithm on high dimensional noise data. Again, each training data set had 200 objects and each test data had 1000 objects.

5.1.2 Real-world data sets

Table 2 lists the real-world data sets used to evaluate the performance of regression forests models. The table is divided into two sections. The top section contains 10 real world data sets in low dimensions. Seven of them were taken from UCI.[1] We removed the object records with missing values and feature "car name" from data set *Auto MPG* because the feature has too many categorical values. Twenty-five predictor features were removed from data set *Communities and Crime*. Three data sets *Childhood, Horse Racing* and *Pulse Rates* were obtained from the DASL[2] database.

The bottom section of Table 2 lists 5 high-dimensional data sets. The *computed tomography (CT)* data was taken from UCI and used to build a regression model to calculate the relative locations of CT slices on the axial axis. The data set was generated from 53,500 images taken from 74 patients (43 males and 31 females). Each CT slice was described by two histograms in a polar space. The first histogram describes the location of bone structures

[1] The data are available at http://archive.ics.uci.edu/.

[2] http://lib.stat.cmu.edu/DASL/DataArchive.html.

Table 2 Characteristics of real-world data sets sorted by the number of features

	Dataset name	#Objects	#Features	#Unique of Y
1	Servo	167	4	51
2	Childhood	654	4	575
3	Computer hardware	209	6	116
4	Auto MPG	392	7	127
5	Concrete slump test	103	7	90
6	Concrete compressive strength	1,030	8	845
7	Pulse rates	110	10	55
8	Horse racing	102	12	19
9	Boston housing	506	13	229
10	Communities and crime	1,994	102	98
11	Computed tomography	53,500	385	53,347
12	Embryonal tumours C	60	7,129	2
13	DLBCL	240	7,399	104
14	Prostate tumor	102	10, 509	2
15	TFIDF-2006	19,395	150,360	15,848

in the image and the second represents the location of air inclusions inside the body. Both histograms are concatenated to form the feature vector.

The microarray data *Diffuse Large B-cell Lymphoma* (DLBCL) was collected from Rosenwald et al. (2002). The DLBCL data consisted of measurements of 7399 genes from 240 patients with diffuse large B-cell lymphoma. The outcome was the survival time, which was either observed or censored. We used observed survival time as a response feature because censored data only had two states, dead or alive. A detailed description can be found in Rosenwald et al. (2002).

Embryonal Tumours C and *Prostate Tumor* are two gene data sets taken from NCBI.[3] Each of those data sets contained two classes. We changed one class label to 1 and the other label to 0, and treat them as continuous values. The classification problem was converted to a regression one. We built a regression random forests model to estimate the outcome and used a given threshold to divide the outcomes into two classes.

The *TFIDF-2006*[4] data set represents a set of financial reports. Each document is associated with an empirical measure of financial risk. These measures were the log transformed volatilities of stock returns. All records from 2001 to 2005 were used to train regression random forests models and the last year records (2006) were used as test data.

5.2 Experimental settings

Two evaluation metrics are used to measure the performance of a model in prediction. They are the *mean of square residuals* (MSR) and the *mean absolute prediction error* ($MAPE$) calculated as follows, respectively.

[3] http://www.ncbi.nlm.nih.gov.

[4] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets.

$$MSR = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{f}(x_i)\right)^2 \tag{17}$$

and

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{f}(x_i)|, \tag{18}$$

where $\hat{f}(x_i)$ is the prediction on $X = x_i$, N is the number of objects in the test data and y_i is the true value.

The $MAPE$ evaluation is used to measure how close the predictions are to Y values. A low MSR and $MAPE$ of a model corresponds to a better prediction performance.

In the experiments, four random forests methods were used to build regression models from the training data sets. They were the regression random forests RF, unbiased conditional random forests cRF, quantile regression forests QRF and the proposed bcQRF. We used the latest R-packages of RF, cRF and QRF, *randomForest, cForest, quantregForest* in these experiments (Liaw and Wiener 2002; Hothorn et al. 2011; Meinshausen 2012). We implemented a new feature weighting subspace sampling method and a new bias-correction method in bcQRF. In the experiment environment, we ran R code to call the corresponding C/C++ functions. The experiment results were evaluated using two measures MSR and MAPE as defined in Eqs. (17) and (18).

For the two large data sets *computed tomography (CT)* and *TFIDF-2006*, we only experimented one model with 500 trees in each random forests method. In CT data, two-third was used for training, and one-third for testing. In TFIDF-2006 data, the given training data was used to learn the models and the test data was used to evaluate the models.

For other remaining synthetic and real-world data sets, we used 10-fold cross-validation to evaluate the prediction performance of regression random forests algorithms. In each fold, we built 30 regression models, each with 500 trees and tested 30 models with the corresponding test data sets. The number of features used to split a node are $\lfloor M/3 \rfloor$ for low dimensional data sets and $\lfloor \sqrt{M} \rfloor$ for high dimensional data, respectively. The parameter *nodesize* in a regression forests model was five. The performance of each algorithm was measured by the average of MSR and MAPE.

5.3 Results and evaluations

5.3.1 Comparison of results on synthetic data sets

The result comparisons of the four regression random forests RF, cRF, QRF and bcQRF from 11 pairs of synthetic data sets evaluated in MAPE and MSR are listed in Table 3. The second pair of the data sets $\mathcal{L}M10_2$ and $\mathcal{H}M10_2$ were generated with Model 2 of Eq. (16) and the rest 10 pairs of data sets were generated from Model 1 of Eq. (4). The upper part of the table shows the MAPE evaluations of the four algorithms on the test data and the lower part is the MSR evaluations of the four algorithms. We can clearly see that bcQRF significantly reduced MAPE and MSR errors, especially on the data sets with the large noise level ($\sigma = 5$). These results demonstrated that the bias correction algorithm bcQRF produced more accurate prediction results and was robust to noisy (irrelevant) features.

Figure 2 shows the MAPE and MSR evaluations of the four regression forests algorithms on three high dimensional test data sets. Thirty regression models were created with each algorithm from each training data set. The MAPE and MSR values of each algorithm on each data set are the averages of MAPE and MSR values of 30 models. At the top of each bar

Table 3 Comparisons of four regression random forests algorithms on synthetic data sets

Training data	Test data	RF	cRF	QRF	bcQRF
MAPE					
$\mathcal{L}M10_1$	$\mathcal{H}M10_1$	2.03	2.37	2.13	**1.67**
$\mathcal{L}M10_2$	$\mathcal{H}M10_2$	2.18	2.83	2.02	**1.68**
$\mathcal{L}M5S0.1$	$\mathcal{H}M5S0.1$	0.151	0.209	0.156	**0.123**
$\mathcal{L}M20S0.1$	$\mathcal{H}M20S0.1$	0.201	0.241	0.196	**0.139**
$\mathcal{L}M50S0.1$	$\mathcal{H}M50S0.1$	0.226	0.253	0.216	**0.149**
$\mathcal{L}M5S1$	$\mathcal{H}M5S1$	1.83	2.31	1.97	**1.55**
$\mathcal{L}M20S1$	$\mathcal{H}M20S1$	2.27	2.510	2.30	**1.80**
$\mathcal{L}M50S1$	$\mathcal{H}M50S1$	2.47	2.61	2.50	**1.93**
$\mathcal{L}M5S5$	$\mathcal{H}M5S5$	18.2	31.7	14.0	**11.1**
$\mathcal{L}M20S5$	$\mathcal{H}M20S5$	25.4	36.8	17.9	**12.3**
$\mathcal{L}M50S5$	$\mathcal{H}M50S5$	28.7	38.7	19.9	**13.1**
MSR					
$\mathcal{L}M10_1$	$\mathcal{H}M10_1$	6.45	8.71	7.09	**4.54**
$\mathcal{L}M10_2$	$\mathcal{H}M10_2$	8.81	16.83	7.43	**4.70**
$\mathcal{L}M5S0.1$	$\mathcal{H}M5S0.1$	0.037	0.070	0.039	**0.024**
$\mathcal{L}M20S0.1$	$\mathcal{H}M20S0.1$	0.064	0.092	0.061	**0.031**
$\mathcal{L}M50S0.1$	$\mathcal{H}M50S0.1$	0.080	0.101	0.074	**0.035**
$\mathcal{L}M5S1$	$\mathcal{H}M5S1$	5.27	8.23	6.07	**3.92**
$\mathcal{L}M20S1$	$\mathcal{H}M20S1$	7.98	9.74	8.23	**5.20**
$\mathcal{L}M50S1$	$\mathcal{H}M50S1$	9.39	10.43	9.68	**5.94**
$\mathcal{L}M5S5$	$\mathcal{H}M5S5$	604.8	1747.1	356.3	**203.7**
$\mathcal{L}M20S5$	$\mathcal{H}M20S5$	1117.3	2343.4	588.8	**250.5**
$\mathcal{L}M50S5$	$\mathcal{H}M50S5$	1371.0	2506.1	727.1	**283.5**

The value of bold in each row indicates the best result from the corresponding data among the algorithms

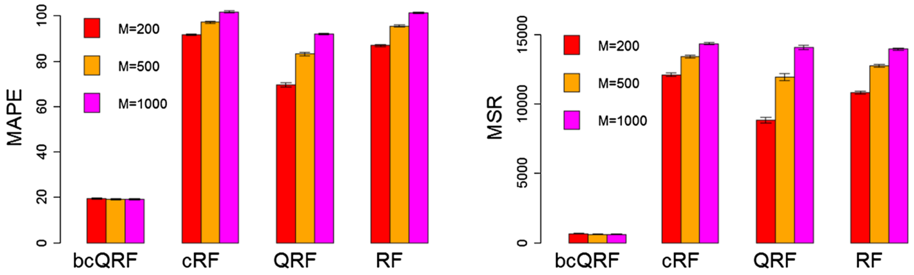
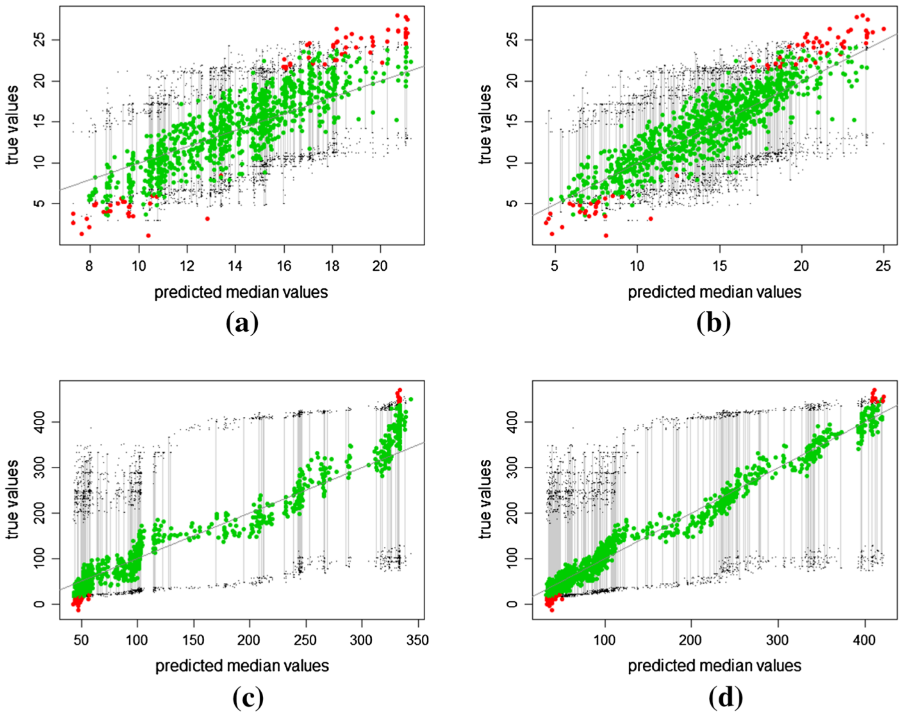


Fig. 2 Comparisons of four regression forests algorithms on three high dimensional test data sets $\{\mathcal{H}M200S5, \mathcal{H}M500S5, \mathcal{H}M1000S5\}$. The results were the averages of 30 models by each algorithm

is the variance of the 30 results. We can see that bcQRF performed much better than other algorithms on these high dimensional synthetic data. This was because the feature weighting subspace sampling method was used in generating trees in bcQRF.

Figure 3 shows prediction results of two test data sets $\mathcal{H}M10_1$ and $\mathcal{H}M1000S5$ by QRF without bias correction in Fig. 3a, c and bcQRF with bias correction in Fig. 3b, d. $\mathcal{H}M10_1$ is a small synthetic data set generated with Model 1. The biased predictions can be clearly



(a) **(b)**

(c) **(d)**

Fig. 3 (**a**) Point and range predictions of data set $\mathcal{HM}10_1$ by QRF. The *solid line* represents the set of points where the predicted values equal to the true values. The predicted values on the *left* are smaller than the true values whereas the predicted values are greater than the true values on the *right*. Both *right* and *left* predictions by QRF are biased. (**b**) Point and range predictions of data set $\mathcal{HM}10_1$ by bcQRF. The *left* and *right* biases are corrected. (**c**) and (**d**) Point and range predictions of data set $\mathcal{HM}1000S5$. The predicted points and ranges in (**d**) are more evenly distributed along the *solid line*, which demonstrates the effect of bias correction by bcQRF

observed from Fig. 3a. The biased predictions were corrected by bcQRF as shown in Fig. 3b. $\mathcal{HM}1000S5$ is a high dimensional data set. We can see that predictions with bias correction by bcQRF were clearly improved as shown in Fig. 3d in comparison with the predictions by QRF in Fig. 3c.

5.3.2 Comparison of prediction performance on real-world data sets

Table 4 lists the MAPE and MSR evaluations of the four regression random forests algorithms RF, cRF, QRF and bcQRF on 10 small real world data sets listed in Table 2. The MAPE and MSR values were the averages of 30 evaluation values from 30 models by each algorithm on each data set. The RFM/RFLADWM column lists the best prediction results of the corresponding data sets obtained by Roy and Larocque (2012). We can see that bcQRF outperformed other three algorithms and the best results by Roy and Larocque (2012) on most data sets. In the data sets where bcQRF did not obtain the best results, the differences from the best results were minor. These results indicate that bcQRF is able to produce the-state-of-the-art prediction results.

Figure 4 shows comparisons of the point and 90 % range predictions of the 10 small real world data sets by QRF and bcQRF. The point predictions were the median values of Y.

Table 4 Comparisons of regression random forests algorithms on 10 small real world data sets listed in Table 2. The best results in the RFM/RFLADWM column lists the best prediction results of the corresponding data sets given in Roy and Larocque (2012)

	Data set	RF	cRF	QRF	RFM/ RFLADWM	bcQRF
MAPE						
1	Servo	0.313	0.505	0.303	0.345	**0.254**
2	Childhood	0.327	0.318	0.320	0.332	**0.315**
3	Computer hardware	26.8	40.5	29.9	**25.8**	25.9
4	Auto MPG	1.89	2.07	1.93	1.87	**1.86**
5	Concrete slump	38.9	47.0	39.9	35.1	**32.8**
6	Concrete com. strength	3.33	5.06	3.35	3.37	**2.89**
7	Pulse rates	9.87	13.76	9.54	9.67	**8.54**
8	Horse racing	1.58	2.06	1.62	1.46	**1.38**
9	Boston housing	2.14	2.54	2.03	2.07	**1.90**
10	Communities and crime	0.092	0.091	0.092	**0.088**	0.089
MSR						
1	Servo	**0.391**	0.745	0.551	0.524	0.420
2	Childhood	0.188	0.173	0.175	0.190	**0.171**
3	Computer hardware	3470	9324	5500	3405	**3370**
4	Auto MPG	**7.15**	8.64	7.96	7.41	7.43
5	Concrete slump test	2357	3218	2747	2060	**1988**
6	Concrete com. strength	22.8	43.5	25.2	25.5	**20.3**
7	Pulse rates	224	337	217	255	**183**
8	Horse racing	4.04	6.93	4.32	3.48	**3.18**
9	Boston housing	10.3	16.0	9.1	10.3	**8.0**
10	Communities and crime	0.019	0.019	0.023	**0.018**	0.019

The value of bold in each row indicates the best result from the corresponding data among the algorithms

The left figure on each data set shows the predictions by QRF and the right figure shows the predictions by bcQRF. Clear improvements in predictions can be observed in the right figures of most data sets from the facts that the predicted points are closer to the diagonal lines which indicates that the predicted values were close to the true values in data, and there are less red points in the right figures which indicates that a large number of predictions were within the predicted ranges. These results clearly demonstrate the advantages of bcQRF over QRF.

Table 5 lists the MAPE and MSR evaluations of three regression random forests algorithms RF, QRF and bcQRF on 5 high dimensional real world data sets. Since cRF could not run on the two large data sets *CT* and *TFIDF-2006*, we exclude it from this experiment. Except for the MSR result of data set *Embryonal Tumours C*, bcQRF outperformed other two algorithms in both MAPE and MSR evaluations. Error reductions on most data sets are significant. Even with data set *Embryonal Tumours C*, the difference of MSR evaluation of bcQRF from the best MSR evaluation of RF is minor. These results demonstrate that bcQRF has advantages over other algorithms on high dimensional data.

Figure 5 shows the point and 90 % range prediction results of two large high dimensional data sets *CT* and *TFIDF-2006* by QRF and bcQRF. It can be seen that prediction errors of *CT*

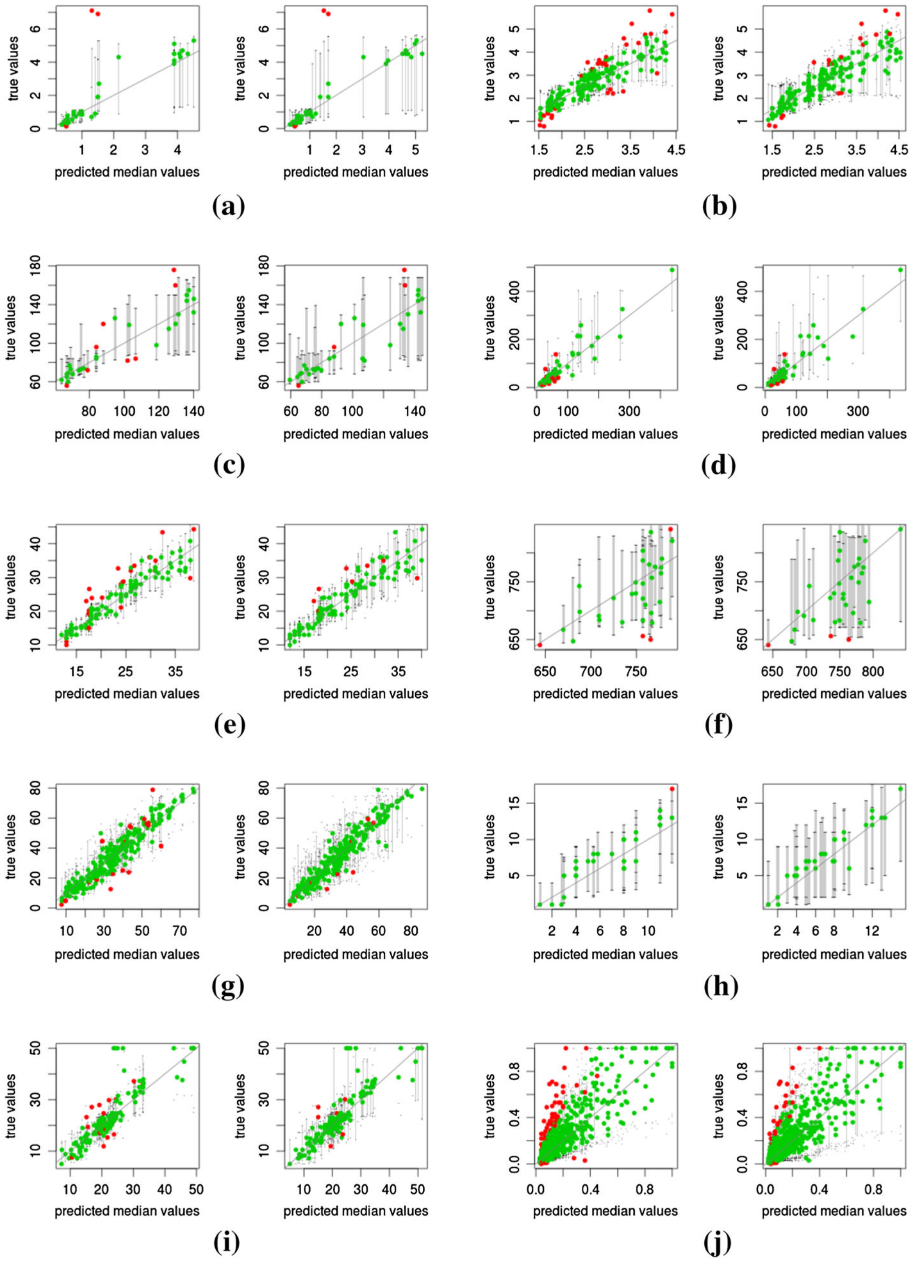


Fig. 4 Comparisons of prediction results on 10 small real world data sets. The *left* figure of each data set shows the point and the 90 % range predictions of QRF without bias correction. The *vertical axis* shows the true values in data and the *horizontal axis* shows the predicted values. The *vertical bars* indicate the predicted ranges. The *green points* show the predictions within the predicted ranges and the red points are the predictions outside the predicted ranges. The *right* figure of each data set shows the predictions of bcQRF with bias correction. (**a**) Servo, (**b**) childhood, (**c**) pulse rate, (**d**) computer hardware, (**e**) auto MPG, (**f**) concrete slump test, (**g**) concrete compressive strength, (**h**) horse racing, (**i**) boston housing, (**j**) communities and crime

Table 5 Comparisons of three regression random forests algorithms on high-dimensional data sets

	Data set	RF	QRF	bcQRF
MAPE				
11	Computed tomography (CT)	0.798	0.392	**0.141**
12	Embryonal tumours C	0.463	0.356	**0.331**
13	DLBCL	3.43	3.10	**2.73**
14	Prostate tumor	0.304	0.117	**0.063**
15	TFIDF-2006	0.450	0.455	**0.232**
MSR				
11	Computed tomography (CT)	1.785	1.280	**0.261**
12	Embryonal tumours C	**0.242**	0.355	0.330
13	DLBCL	16.5	17.9	**13.8**
14	Prostate tumor	0.118	0.115	**0.063**
15	TFIDF-2006	0.337	0.351	**0.118**

The value of bold in each row indicates the best result from the corresponding data among the algorithms

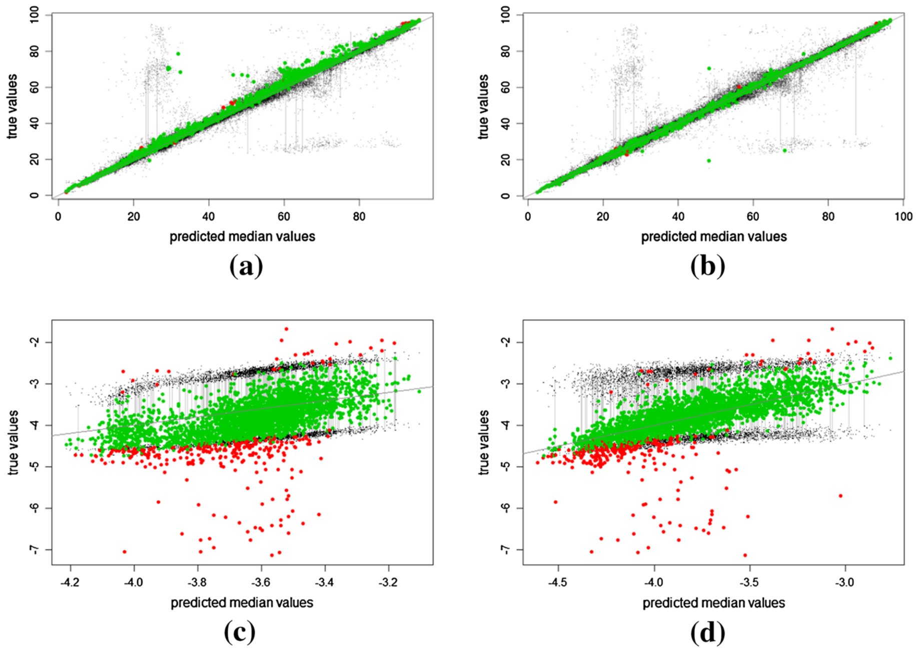


Fig. 5 Comparisons of range predictions by QRF and bcQRF on high-dimensional data sets *CT* and *TFIDF-2006*. (**a**) Range predictions of CT data by QRF, (**b**) Range predictions of CT data by bcQRF, (**c**) Range predictions of TFIDF-2006 by QRF, (**d**) Range predictions of TFIDF-2006 by bc-QRF

data set by QRF appeared at values in the ranges of [20-30 cm] (shoulder part) and [60–75 cm] (abdomen part) as shown in Fig. 5a. The prediction errors in the same regions reduced in the predictions by bcQRF as shown in Fig. 5b. For the predictions of data set *TFIDF-2006*, a clear bias in point prediction can be observed in the results of QRF as shown in Fig. 5c. The bias was corrected in the results of bcQRF as shown in Fig. 5d. The effect of bias correction is clearly demonstrated in this result.

6 Conclusions

We have presented a new bias-correction quantile regression forests algorithm bcQRF that uses two-level models to correct bias in point and range predictions and improve the performance of prediction models. The first level model is an extended quantile regression forests model that uses a feature weighting subspace sampling method in tree growth to improve prediction accuracy of the tree models. The second level regression forests model is built with the residuals of the first level model as the response feature in the training data so it can predict the bias of the first level model. With these two techniques, bcQRF can effectively correct the prediction bias that often occurs in predictions of other regression random forests models. We have presented a series of experiment results on both synthetic and real world data sets to demonstrate the capability of the bcQRF models in bias correction and advantages of bcQRF over other commonly used regression random forests algorithms.

In our future work, we will increase the scalability of the bcQRF algorithm by parallelizing it on the cloud platform to deal with big data.

Acknowledgments This work is supported by the Shenzhen New Industry Development Fund under Grant NO.JC201005270342A and the project "Some Advanced Statistical Learning Techniques for Computer Vision" funded by the National Foundation of Science and Technology Development, Vietnam under grant number 102.01-2011.17.

References

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Breiman, L. (1999). Using adaptive bagging to debias regressions. Technical report, Technical Report 547, Statistics Dept. UCB.

Breiman, Leo. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. Boca Raton: CRC Press.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics*, *19*, 1–67.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning* (Vol. 2). New York: Springer.

Hothorn, T., Hornik, K., & Zeileis, A. (2011) party: A laboratory for recursive part (y) itioning. r package version 0.9-9999. URL: http://cran.r-project.org/package=party. Accessed 28 Nov 2013.

Kursa, M. B., & Rudnicki, W. R. (2010). Feature selection with the boruta package. *Journal of Statistical Software*, *36*, 1–13.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomforest. *R news*, *2*(3), 18–22.

Meinshausen, N. (2006). Quantile regression forests. *The Journal of Machine Learning Research*, *7*, 983–999.

Meinshausen, N. (2012). Quantregforest: quantile regression forests. *R package version 0.2-3*.

Rosenwald, A., Wright, G., Chan, W. C., Connors, J. M., Campo, E., Fisher, R. I., et al. (2002). The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. *New England Journal of Medicine*, *346*(25), 1937–1947.

Roy, M. H., & Larocque, D. (2012). Robustness of random forests for regression. *Journal of Nonparametric Statistics*, *24*(4), 993–1006.

Sandri, M., & Zuccolotto, P. (2008). A bias correction algorithm for the gini variable importance measure in classification trees. *Journal of Computational and Graphical Statistics*, *17*(3), 27.

Sandri, M., & Zuccolotto, P. (2010). Analysis and correction of bias in total decrease in node impurity measures for tree-based algorithms. *Statistics and Computing*, *20*(4), 393–407.

Stoppiglia, H., Dreyfus, G., Dubois, R., & Oussar, Y. (2003). Ranking a random feature for variable and feature selection. *The Journal of Machine Learning Research*, *3*, 1399–1414.

Tung, N. T., Huang, J. Z., Imran, K., Li, M. J., & Williams, G. (2014). Extensions to quantile regression forests for very high dimensional data. In *Advances in knowledge discovery and data mining*, vol. 8444, (pp. 247–258). Springer.

Tuv, E., Borisov, A., & Torkkola, K. (2006). Feature selection using ensemble based ranking against artificial contrasts. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, (pp. 2181–2186). IEEE.

Tuv, E., Borisov, A., Runger, G., & Torkkola, K. (2009). Feature selection with ensembles, artificial variables, and redundancy elimination. *The Journal of Machine Learning Research*, *10*, 1341–1366.

Welch, B. L. (1947). The generalization ofstudent's' problem when several different population variances are involved. *Biometrika*, *84*, 28–35.

Xu, R. (2013). Improvements to random forest methodology. PhD thesis, Iowa State University.

Zhang, G., & Yan, L. (2012). Bias-corrected random forests in regression. *Journal of Applied Statistics*, *39*(1), 151–160.