Extensions to Quantile Regression Forests for Very High-Dimensional Data

Nguyen Thanh Tung[1], Joshua Zhexue Huang[2], Imran Khan[1], Mark Junjie Li[2], and Graham Williams[1]

[1] Shenzhen Key Laboratory of High Performance Data Mining. Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
[2] College of Computer Science and Software Engineering, Shenzhen University
tungnt@wru.vn, {zx.huang,jj.li}@szu.edu.cn, imran.khan@siat.ac.cn, Graham.Williams@togaware.com

Abstract. This paper describes new extensions to the state-of-the-art regression random forests *Quantile Regression Forests* (QRF) for applications to high-dimensional data with thousands of features. We propose a new subspace sampling method that randomly samples a subset of features from two separate feature sets, one containing important features and the other one containing less important features. The two feature sets partition the input data based on the importance measures of features. The partition is generated by using feature permutation to produce raw importance feature scores first and then applying p-value assessment to separate important features from the less important ones. The new subspace sampling method enables to generate trees from bagged sample data with smaller regression errors. For point regression, we choose the prediction value of Y from the range between two quantiles $Q_{0.05}$ and $Q_{0.95}$ instead of the conditional mean used in regression random forests. Our experiment results have shown that random forests with these extensions outperformed regression random forests and quantile regression forests in reduction of root mean square residuals.

Keywords: Regression Random Forests, Quantile Regression Forests, Data Mining, High-dimensional Data.

1 Introduction

Regression is a task of learning a function $f(\mathbf{X}) = E(Y|\mathbf{X})$ from a training data $\mathbb{L} = \{(\mathbf{X}, Y) = (\mathbf{X_1}, Y_1), ..., (\mathbf{X_N}, Y_N)\}$, where N is the number of objects in $\mathbb{L}$, $X \in \mathbb{R}^M$ are predictor variables or features and $Y \in \mathbb{R}^1$ is a response variable or feature. The regression model has the form

$$Y = E(Y|\mathbf{X}) + \epsilon \tag{1}$$

where error $\epsilon \sim N(0, \sigma^2)$.

A parametric method assumes that a formula for conditional mean $E(Y|\mathbf{X})$ is known, for instance, linear equation $Y = \beta_0 + \beta_1 X_1, \ldots, \beta_M X_M$. The linear

V.S. Tseng et al. (Eds.): PAKDD 2014, Part II, LNAI 8444, pp. 247–258, 2014.

regression model is solved by estimating parameters $\beta_0, \beta_1, \ldots, \beta_M$ from $\mathbb{L}$ with least squares method to minimize the sum of square residuals. A nonparametric method does not require that a model form be known. Instead, a model structure is specified, such as a neural network and $\mathbb{L}$ is used to learn the model. Linear regression models do not perform on nonlinear domains and suffer the problem of curse of dimensionality. Neural networks are not scalable to big data.

Decision tree is a nonparametric regression model that works on nonlinear situations. A decision tree model partitions the training data $\mathbb{L}$ into subsets of leaf nodes and the prediction value in each leaf node is taken as the mean of Y values of the objects in that leaf node. Decision tree model is unstable in high-dimensional data because of the large prediction variance. This problem can be remedied by using an ensemble of decision trees or random forests [3] built from the bagged samples of $\mathbb{L}$ [2]. Regression random forests takes the average of multiple decision tree predictions to reduce the prediction variance and increase the accuracy of prediction.

Quantile regression forests (QRF) represents the state-of-the-art technique for nonparametric regression [7]. Instead of modeling $Y = E(Y|\mathbf{X})$, QRF models $F(y|X = x) = P(Y < y|X = x)$, i.e., the conditional distribution function. Given a continuous distribution function and a probability α, the α-quantile $Q_\alpha(x)$ can be computed as

$$P(Y < Q_\alpha(x)|X = x) = \alpha \tag{2}$$

where $0 < \alpha < 1$. Given two quantile probabilities α_l and α_h, QRF enables to predict 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 the range prediction, quantile regression forests can perform well in situations where the conditional distribution function is not in normal distribution.

Both regression random forests and quantile regression forests suffer performance problems in high-dimensional data with thousands of features. The main cause is that in the process of growing a tree from the bagged sample data, the subspace of features randomly sampled from the thousands of features in $\mathbb{L}$ to split a node of the tree is often dominated by less important features, and the tree grown from such randomly sampled subspace features will have a low accuracy in prediction which affects the final prediction of the random forests.

In this paper, we propose a new subspace feature sampling method to grow trees for regression random forests. Given a training data set $\mathbb{L}$, we first use 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 $\mathbb{L}$ 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 the subspace features 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 sampling method enables to generate trees from bagged sample data with smaller regression errors.

For point regression, we choose the prediction value of Y from the range between two quantiles $Q_{0.05}$ and $Q_{0.95}$ instead of the conditional mean used in regression random forests. Our experiment results have shown that random forests with these extensions outperformed regression random forests and quantile regression forests in reduction of root mean square residuals (RMSR).

2 Random Forests for Regression

2.1 Regression Random Forests

Given a training data $\mathbb{L} = \{(\mathbf{X_1}, Y_1), ..., (\mathbf{X_N}, Y_N)\}$, where N is the number of objects in $\mathbb{L}$, a regression random forests model is built as follows.

- Step 1: Draw a bagged sample $\mathbb{L}_k$ from $\mathbb{L}$.
- Step 2: Grow a regression tree T_k from $\mathbb{L}_k$. At each node t, the split is determined by the decrease in impurity that is defined as $\sum_{x_i \in t}(Y_i - \bar{Y}_t)/N(t)$, 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 sample, it is grown with only two-thirds of objects in $\mathbb{L}$. 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.

2.2 Quantile Regression Forests

Quantile Regression Forests (QRF) uses the same method as described above to grow trees [7]. However, at each leaf node, it retains all Y values instead of only the mean of Y values. Therefore, QRF keeps the raw distribution of Y values at leaf node.

To describe QRF with notation by Breiman [3], we compute a positive weight $w_i(x, \theta_k)$ by each tree for each case $X_i \in \mathbb{L}$, where θ_k indicates the kth tree for a new given x. Let $l(x, \theta_k)$ be a leaf node t. All $X_i \in l(x, \theta_k)$ are assigned to an equal weight $w_i(x, \theta_k) = 1/N(t)$ and $X_i \notin l(x, \theta_k)$ are assigned to 0 otherwise, where $N(t)$ is the number of objects in $l(x, \theta_k)$. For single tree prediction, given $X = x$, the prediction value is

$$\hat{Y}^k = \sum_{i=1}^{N} w_i(x, \theta_k)Y_i = \sum_{x, X_i \in l(x,\theta_k)} w_i(x, \theta_k)Y_i = \frac{1}{N(t)}\sum_{x, X_i \in l(x,\theta_k)} Y_i \quad (3)$$

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

$$w_i(x) = \frac{1}{K} \sum_{k=1}^{K} w_i(x, \theta_k) \tag{4}$$

The prediction of regression random forests is

$$\hat{Y} = \sum_{i=1}^{N} w_i(x) Y_i \tag{5}$$

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

Given an input X, we can find the leaf node $l_k(x, \theta_k)$ from all trees 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

$$\hat{F}(y|\mathbf{X} = x) = \sum_{i=1}^{N} w_i(x) \mathcal{I}(Y_i \leq y) \tag{6}$$

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

$$\hat{Q}_\alpha(\mathbf{X} = x) = inf\{y : \hat{F}(y|\mathbf{X} = x) \geq \alpha\}. \tag{7}$$

For range prediction, we have

$$[Q_{\alpha_l}(X), Q_{\alpha_h}(X)] = [inf\{y : \hat{F}(y|\mathbf{X} = x) \geq \alpha_l\}, inf\{y : \hat{F}(y|\mathbf{X} = x) \geq \alpha_h\}] \tag{8}$$

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 choose a value in a range such as the mean or median of Y_i values. The median surpasses the mean in robustness towards extreme values/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 Subspace Selection

3.1 Importance Measure of Features by Permutation

Given a training data set $\mathbb{L}$ and a regression random forests model RF, Breiman [3] described a permutation method to measure the importance of features in the prediction. The procedure for computing the importance scores of features consists of the following steps.

1. Let $\mathbb{L}_k^{oob}$ be the *out-of-bag* samples of the kth tree. Given $X_i \in \mathbb{L}_k^{oob}$, use the tree T_k to predict $\hat{Y}_i^k$, denoted as $\hat{f}_i^k(\mathbf{X_i})$.

2. Choose a predictor feature j and randomly permute the value of feature j in X_i with another case in $\mathbb{L}_k^{oob}$. Use tree T_k to obtain the new prediction on the permuted X_i as $\hat{f}_i^{k,p,j}(\mathbf{X_i})$. Repeat the permutation process P times.
3. For M_i trees grown without X_i, compute the out-of-bag prediction by RF in the pth permutation of the jth predictor feature as

$$\hat{f}_i^{p,j}(\mathbf{X_i}) = \frac{1}{M_i} \sum_{X_i \in \mathbb{L}_k^{oob}} \hat{f}_i^{k,p,j}(\mathbf{X_i})$$

.
4. Compute the two *mean square residuals* (MSR) with and without permutations of predictor feature j on X_i as $MSR_i = \frac{1}{M_i} \sum_{k \in M_i} (\hat{f}_i^k(\mathbf{X_i}) - Y_i)^2$ and $MSR_i^j = \frac{1}{P} \sum_{p=1}^{P} (\hat{f}_i^{p,j}(\mathbf{X_i}) - Y_i)^2$, respectively.
5. Let $\Delta MSR_i^j = max(0, MSR_i^j - MSR_i)$. The importance of feature j is $IMP_j = \frac{1}{N} \sum_{i \in \mathbb{L}} \Delta MSR_i^j$. To normalize the importance measures, we have the raw importance score as

$$VI_j = \frac{IMP_j}{\sum_l IMP_l} \tag{9}$$

With the raw importance scores by (9) we can rank the features on the importance.

3.2 *p*-Value Feature Assessment

Permutation method only gives the importance ranking of features. We need to identify important features from less important ones. To do so, we use Welch's two-sample t-test that compares the importance score of a feature with the maximum importance scores of 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 noisy features, it is less important. Otherwise, it is considered as important. This idea was introduced by Stoppiglia et al. [10], and were further developed in [5], [11].

Table 1. The importance scores matrix of all real features and shadows with R replicates

Iteration	VI_{X_1}	VI_{X_2}	...	VI_{X_M}	$VI_{A_{M+1}}$	$VI_{A_{M+2}}$	...	$VI_{A_{2M}}$
1	$VI_{x_{1,1}}$	$VI_{x_{1,2}}$	...	$VI_{x_{1,M}}$	$VI_{a_{1,(M+1)}}$	$VI_{a_{1,(M+2)}}$	...	$VI_{a_{1,2M}}$
2	$VI_{x_{2,1}}$	$VI_{x_{2,2}}$	...	$VI_{x_{2,M}}$	$VI_{a_{2,(M+1)}}$	$VI_{a_{2,(M+2)}}$	...	$VI_{a_{2,2M}}$
⋮	⋮							⋮
R	$VI_{x_{R,1}}$	$VI_{x_{R,2}}$	...	$VI_{x_{R,M}}$	$VI_{a_{R,(M+1)}}$	$VI_{a_{R,(M+2)}}$	...	$VI_{a_{R,2M}}$

We build a random forests model RF from this extended data set. Following the importance measure by 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 shows the importance measure of M features in input data and M shadow features generated by permutating the values of the corresponding feature in data.

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

$$t_i = \frac{\overline{X}_i - \overline{V}^*}{\sqrt{(s_1^2 + s_2^2)/R}} \tag{10}$$

where s_1^2 and s_2^2 are the unbiased estimators of the variances of the two samples. For significance test, the distribution of t_i in (10) is approximated as an ordinary Student's distribution with the degrees of freedom df calculated as

$$df = \frac{(s_1^2/n_1 + s_2^2/n_2)^2}{(s_1^2/n_1)^2/(n_1 - 1) + (s^2{}_2/n_2)^2/(n_2 - 1)} \tag{11}$$

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 $\overline{X}_i > \overline{V}^*$. 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 it is added to the less important feature subset X_{low} otherwise. The two subsets partitions 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 A New Quantile Regression Forests Algorithm

Now we can extend the quantile regression forests with the new feature 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 a high

probability. The new quantile regression forests algorithm eQRF is summarized as follows.

1. Given $\mathbb{L}$, generate the extended data set $\mathbb{L}^e$ in $2M$ dimensions by permutating the corresponding predictor feature values for shadow features.
2. Build a regression random forests model RF^e from $\mathbb{L}^e$ 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 as (10).
4. Compute the degree of freedom df as (11).
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 features in two feature subsets X_{low} and X_{high}.
7. Sample the training set $\mathbb{L}$ with replacement to generate bagged samples $\mathbb{L}_1, \mathbb{L}_2, .., \mathbb{L}_K$.
8. For each sample $\mathbb{L}_k$, grow a regression tree T_k as follows:
   (a) At each node, select a subspace of $m = \lfloor \sqrt{M} \rfloor (m > 1)$ features randomly and separately from X_{low} and X_{high} and use the 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 (8) (We set default values $[\alpha_l = 0.05, \alpha_h = 0.95]$ and $\tau = 0.9$).
10. Given a $\mathbf{X}$, estimate the prediction value from a value in the quantile range of Q_{α_l} and Q_{α_h} such as mean or median.

5 Simulation Analysis

5.1 Simulation Data

We used three models as listed in Table 2 to generate synthetic data for simulation analysis. Each model has 5 predictor variables or features. With each model, we first created 200 objects in 5 dimensions plus a response feature. After this, we expanded the data set with different numbers of noisy features and obtained 5 data sets named as {𝕃M5, 𝕃M50, 𝕃M500, 𝕃M2000, 𝕃M5000} where the number in the data name indicates dimensions of the data set. Similarly, we generated extra 5 data sets with 1000 objects from each model as test data sets named {𝕀M5, 𝕀M50, 𝕀M500, 𝕀M2000, 𝕀M5000}.

Table 2. Three simulation models for synthetic data generation. Each model uses 5 iid predictor features from $U(0,1)$ and ϵ from *Exp(1)* (exponential mean 1) distribution.

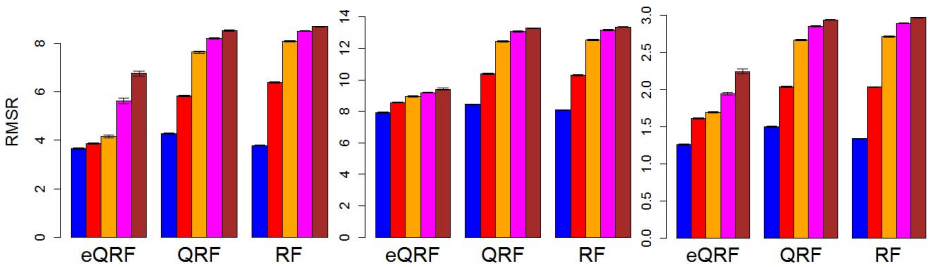
Model	Error Distribution	Simulation models
1	*Exp(1)*	$Y = 10(X_1 + X_2 + X_3 + X_4 + X_5 - 2.5)^2 + \epsilon$
2	*Exp(1)*	$Y = 10sin(\pi X_1 X_2) + 20(X_3 - 0.5)^2 + 10X_4 + 5X_5 + \epsilon$
3	*Exp(1)*	$Y = 0.1e^{4X_1} + 4/[1 + e^{-20(X_2 - 0.5)}] + 3X_3 + 2X_4 + X_5 + \epsilon$

5.2 Evaluation Measure

The performance of a model was evaluated on test data with the *root mean of square residuals* ($RMSR$) computed as

$$RMSR = \sqrt{\frac{1}{\|\mathbb{H}\|} \sum_{\mathbf{X}_i \in \mathbb{H}} [\hat{f}_{\mathbb{H}}(\mathbf{X}_i) - Y_i]^2}. \tag{12}$$

where $\hat{f}_{\mathbb{H}}(\mathbf{X}_i)$ is the prediction given X_i, $\mathbb{H}$ is a test data set and $\|\mathbb{H}\|$ is the number of objects in test data $\mathbb{H}$.



(a) Model 1 (b) Model 2 (c) Model 3

Fig. 1. Comparisons of three regression forests algorithms on 5 test data sets generated with the simulation models in Table 2

5.3 Evaluation Results

We used regression random forests RF, quantile regression forests QRF and our algorithm eQRF to build regression models from the training data sets and used evaluation measure (12) to evaluate the models with the test data sets. We used the latest RF and QRF packages *randomForest, quantregForest* in R in these experiments [6], [8]. For each training data set, we built 100 regression models, each with 500 trees and tested the 100 models with the corresponding test data. Then, the result was evaluated with (12) and the average of 100 models was computed.

Figure 1 shows the evaluation results of three random forests regression methods in RMSR measures. Each random forests method produced 5 test results on 5 simulation data sets from the left to right as {HM5, HM50, HM500, HM2000, HM5000}. We can see that the more noisy features in the data, the lower accuracy in the prediction model. Clearly, in all simulated data generated by the three models in Table 2, eQRF performed the best and its RMSR was significantly lower than those of QRF and RF.

6 Experiments on Real Datasets

6.1 Real-World Data

Five real-world data sets were used to evaluate the performance of our new regression random forests algorithm. The general characteristics of these data sets are presented in Table 3.

The *computed tomography (CT)* data was taken from the UCI[1] which was 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 in the image and the second represents the location of air inclusions inside of the body. Both histograms are concatenated to form the feature vector.

TFIDF-2006 [2] is a text data set containing financial reports. Each document is associated with an empirical measure of financial risk. These measures are log transformed volatilities of stock returns.

The *Microarray data "Diffuse Large B-cell Lymphoma"* (DLBCL) was collected from Rosenwald et al. [9]. The DLBCL data consisted of measurements of 7399 genes from 240 patients with diffuse large B-cell lymphoma. The outcome was survival time, which was either observed or censored. We used observed survival time as the response feature because censored data only indicates two states, dead or alive. A detailed description can be found in [9].

"Leukemia" and *"Lung cancer"* are two gene data sets taken from NCBI [3]. Each of those data sets contains two classes. We changed one class label to 1 and another label to 0. We treat 0 and 1 as continuous values and consider this problem as a regression problem. We built a regression random forests model to estimate the outcome and used a defined threshold to divide the outcomes into two classes.

6.2 Experiments and Results

For each real-world data set, we used two-third of data for training and one-third for testing. We generated 10 models from each training data and each model

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

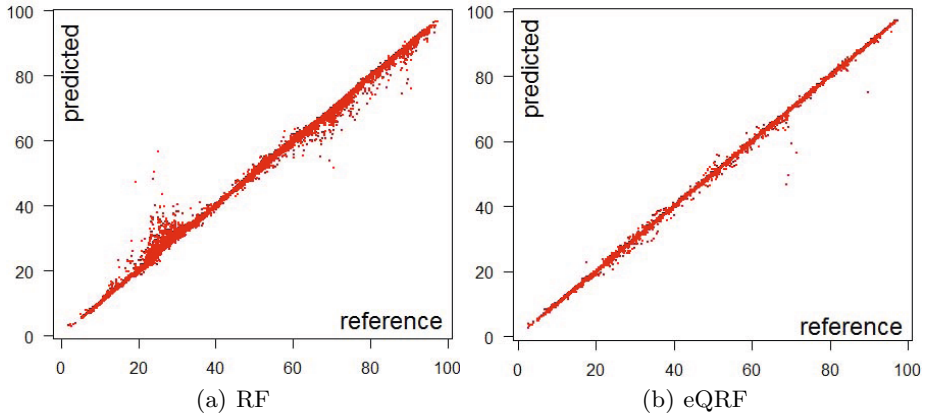
[2] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets`

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

Table 3. Description of the real data sets sorted by the number of features and RMSR performance of three regression algorithms

Dataset	Name	#training	#testing	#features	eQRF	RF	QRF
1	CT Data	35,700	17,800	385	**0.29**	1.33	2.09
2	Leukemia	48	24	7,129	**0.17**	0.22	0.24
3	DLBCL Data	160	80	7,399	**3.77**	4.28	4.55
4	Lung cancer	114	58	54,675	**0.21**	0.32	0.36
5	TFIDF-2006	16,087	3,308	150,361	**0.41**	0.68	0.69

contained 200 trees. We computed the average of RMSRs of the 10 models with (12). The average RMSRs of three regression random forests models on five real-world data sets are shown in Table 3 on the right. We can see that eQRF had the lowest average RMSR. RF performed better than QRF.

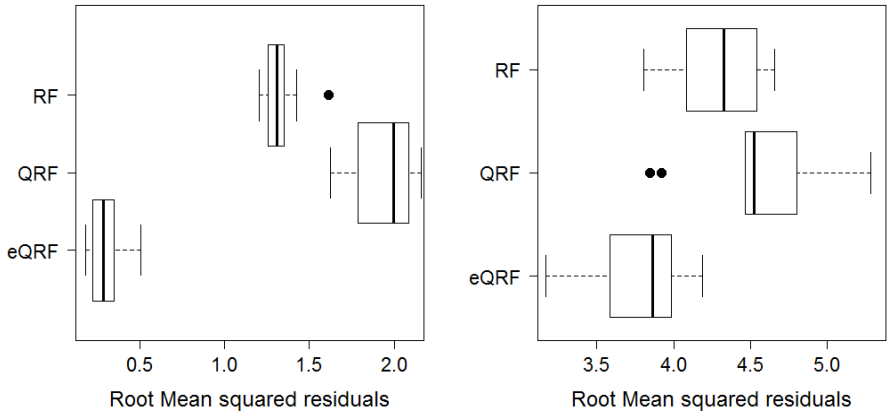


(a) RF (b) eQRF

Fig. 2. Plots of predicted response values against the true values of CT test data. (a) Result of RF. (b) Result of eQRF.

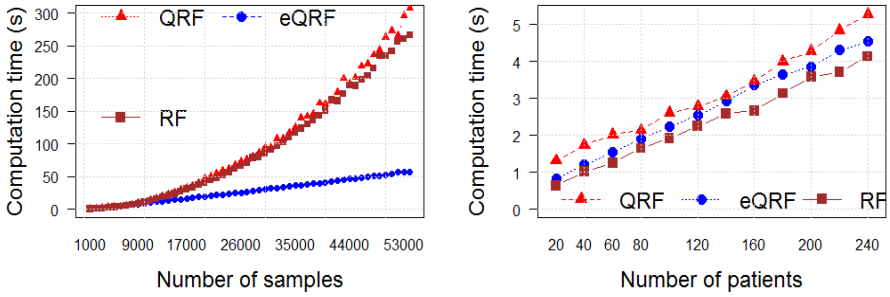
Figure 2 plots the predicted values by RF and eQRF against the true values of the response feature in CT test data. We can see from Figure 2 (a) that there are some regions that RF predicted higher than the true value, for instance [25 cm, 35 cm], and some regions that RF predicted lower than the true value, for instance > 70 cm. These are the prediction error regions including shoulder [20-30cm] and abdomen [60-75cm]. On the contrast, the predicted values of eQRF were more close to the true values as shown in Figure 2 (b) and the prediction results are consistent and more stable in all regions of human body.

Figure 3 shows the average RMSR box plots of three regression models from the real-world data sets. Figure 3 (a) is the result of CT data and Figure 3 (b) is the result of DLBCL data. We can see that eQRF produced less RMSR than QRF and RF and the variance is also small.



(a) CT Data (b) DLBCL Data

Fig. 3. Boxplots of RMSR of three models RF, QRF and eQRF. (a) Result of CT test data. (b) Result of DLBCL test data.



(a) CT Data (b) DLBCL Data

Fig. 4. .Plots of computational time of three algorithms against the number of objects in data. The experiments were conducted on a computer with 2.13 Ghz Intel Core 2 Quad processor and 24GB RAM. (a) Result of CT data. (b) Result of DLBCL data.

Figure 4 shows the computational time of three regression models on the two data sets. We can see that the computational time of the three models linearly increases as the number of objects increases if the size is small, such as DLBCL data. However, for data set with a large number of objects as CT data, the computational times of RF and QRF increase exponentially as shown in Figure 4 (a) but eQRF still maintains a linear increase as shown in Figure 4 (b).

7 Conclusions

We have presented a new regression random forests algorithm for high-dimensional data with thousands of features. In this algorithm, we have made two extensions to the quantile regression forests. One is the subspace sampling method to select the set of features for splitting a node in growing trees. The other is to use the median of Y values in the range of two quantile as the prediction of Y given an input X. The first extension increases the prediction accuracy of decision trees. The second extension reduces the effect of outliers and reduces the variance of random forests regression. Experiment results have demonstrated the improvement in reduction of RMSR in comparison with regression random forests and quantile regression forests.

Acknowledgment. This research is supported in part by NSFC under Grant No.61203294, Shenzhen New Industry Development Fund under Grant No.JC201 005270342A, No.JCYJ20120617120716224, the National High-tech Research and Development Program(No. 2012AA040912), and Guangdong-CAS project(No. 2011B090300025).

References

1. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.: Classification and Regression Trees. Wadsworth International, Belmont (1984)
2. Breiman, L.: Bagging Predictors. Machine Learning 24(2), 123–140 (1996)
3. Breiman, L.: Random Forests. Machine Learning 45(1), 5–32 (2001)
4. Ho, T.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(8), 832–844 (1998)
5. Kursa, M.B., Rudnicki, W.R.: Feature Selection with the Boruta Package. Journal of Statistical Software 36(11) (2010)
6. Liaw, A., Wiener, M.: randomForest 4.6-7. R package (2012), `http://cran.r-project.org`
7. Meinshausen, N.: Quantile Random Forests. Journal Machine Learning Research, 983–999 (2006)
8. Meinshausen, N.: quantregForest 0.2-3. R package (2012), `http://cran.r-project.org`
9. Rosenwald, A., et al.: The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. N. Engl. J. Med. 346, 1937–1947 (2002)
10. Stoppiglia, H., Dreyfus, G.: Ranking a random feature for variable and feature selection. The Journal of Machine Learning Research 3, 1399–1414 (2003)
11. Tuv, E., Borisov, A., Runger, G., Torkkola, K.: Feature selection with ensembles, artificial variables, and redundancy elimination. The Journal of Machine Learning Research 10, 1341–1366 (2009)