Les Cahiers du GERAD ISSN: 0711–2440

Robustness of Random Forests for Regression

M.-H. Roy
D. Larocque

G–2010–56

Octobre 2010

Robustness of Random Forests for Regression

Marie-Hélène Roy
Denis Larocque

Department of Management Sciences
HEC Montréal
3000 chemin de la Côte-Sainte-Catherine
Montréal (Québec), Canada, H3T 2A7
{marie-helene.3.roy; denis.larocque}@hec.ca

Octobre 2010

Les Cahiers du GERAD
G–2010–56

Abstract

In this paper, we empirically investigate the robustness of random forests for regression problems. We also investigate the performance of five variations of the original random forest method, all aimed at improving robustness. All the proposed variations can be easily implemented using the R package randomForest. The first main idea behind these variations is the use of the median, instead of the mean, to combine the predictions from the individual trees. The second idea is to build the trees using the ranks of the response instead of the original values. The competing methods are compared via a simulation study and ten real data sets obtained from the UCI Machine Learning Repository. Our results show that the median–based random forests (using either the ranks or the original responses) offer good and stable performances for the simulated and real data sets considered and, as such, should be considered as serious alternatives to the original random forest method.

Key Words: Random Forests; Robustness; Median; Ranks; UCI Machine Learning Repository.

Résumé

Dans cet article, nous explorons de manière empirique la robustesse des forêts aléatoires dans le contexte de la régression. De plus, nous étudions la performance de cinq variations, visant toutes l'amélioration de sa robustesse, au modèle des forêts aléatoires original. Ces méthodes offrent, entre autres, l'avantage d'être facile à implémenter en utilisant le "package" "randomForest" du logiciel R. La première idée à l'origine de ces variations est l'utilisation de la médiane, en remplacement de la moyenne, pour combiner les prédictions générées par les arbres. La deuxième idée est de construire les arbres en utilisant les rangs de la réponse en remplacement des valeurs originales. Les méthodes à l'étude sont comparées à travers une simulation et des ensembles de données réels provenant du "UCI Machine Learning Repository". Les résultats montrent que les forêts aléatoires construites avec la médiane (en utilisant soit les réponses originales ou les rangs) offrent une performance stable et compétitive sur les ensembles de données simulées et réelles considérées. Ainsi, ces méthodes devraient être considérées comme de sérieuses alternatives à la méthode des forêts aléatoires originale.

Mots clés : Forêts aléatoires; Robustesse; Médiane; Rangs; *UCI Machine Learning Repository*.

1 Introduction

The random forest algorithm (Breiman, 2001) is one of the most interesting and powerful statistical learning method. Among other advantages, it is fast, versatile and has the ability to work with very large data sets. It has been tested and tried in a wide array of domains with real and simulated data sets and has proven to yield very accurate results. Siroky (2009) provides a recent survey on random forests. Moreover, random forests are part of the vast family of ensemble methods for which a survey appears in Rokach (2008).

In the classification context, Hamza and Larocque (2005) empirically showed that random forests are more robust against noise in the outcome compared to many other methods. Folleco et al. (2008) reported similar findings. However, it is not clear if, and to what extent, random forests are sensitive to outlying values in the regression context. If they are sensitive, then our goal is to investigate whether or not simple variations of random forests could improve robustness. The two main ideas that will be studied are i) the use of the median, instead of the mean, for aggregation, and ii) using the ranks of the outcome, instead of the original values, for constructing the trees in the forest. The principal advantage of these ideas is that their implementations require only minimal modifications of existing codes. Our motivation for using this approach comes from the facts that the median is a well known robust location estimator and that the appropriate use of ranks, which is well developed and established in classical nonparametric statistics, usually entails good robustness properties.

This paper presents the results from an empirical study comparing the predictive accuracy and robustness of six methods. These are, the original random forest, the original method using the median, instead of the mean, to combine the predictions, and four variations using forests constructed with the ranks of the outcome instead of the original values. Section 2 presents the description of the six methods under investigation and the three criteria used to assess their performances. Section 3 presents the results from applying the methods to simulated data sets and investigates the impact of three different types of contamination. Section 4 presents the results from applying the methods to real data sets obtained from the UCI Machine Learning Repository (Frank and Asuncion, 2010). Concluding remarks are given in Section 5.

2 Description of the Methods and Evaluation Criteria

We focus on the regression problem with a continuous outcome Y and a vector of p predictors X. The underlying model is

$$Y = h(X) + \epsilon$$

where h is an unknown function and ϵ is random noise. The goal is to obtain a prediction of Y for a new observation with $X = x$ based on a training sample (Y_i, X_i), $i = 1, \ldots, n$. We will denote this prediction by $\hat{h}(x)$.

Six methods are evaluated using three performance criteria. The first method is the original random forest (RF) algorithm as described below.

2.1 The Basic Random Forest Algorithm

1. Draw K bootstrap samples from the original data.
2. For each bootstrap sample, grow an unpruned regression tree, with the following modification: at each node, rather than choosing the best split among all predictors, randomly sample p_0 ($0 < p_0 \leq p$) of the p predictors and choose the best split from among those variables.
3. Predict new data by averaging the predictions of the K trees.

The algorithm has the particularity of exploiting two layers of randomness, random inputs (bootstrap) and random features (random selection of a subset of predictors), which greatly enhances its speed and accuracy.

2.2 Variations on the Basic Algorithm

In order to predict an observation with predictor values x, let $\hat{h}_k(x)$ denote the prediction from the k^{th} tree in the forest, $k = 1, \ldots, K$.

The original RF method uses the mean of the individual predictions to establish the final predictions:

$$\hat{h}_{RF} = \frac{1}{K} \sum_{k=1}^{K} \hat{h}_k(x) = \text{mean}(\hat{h}_k(x)). \tag{1}$$

The first variation, RFM, uses the original RF algorithm with the exception of using the median, instead of the mean, to aggregate the predictions over the trees:

$$\hat{h}_{RFM} = \text{median}(\hat{h}_k(x)). \tag{2}$$

This modification is meant as a first potential improvement of the random forests robustness.

The next four methods are based on random forests grown using the ranks of the outcome Y instead of the original values. Thus, each tree returns a predicted rank denoted by $\hat{R}_k(x)$. Given a rank $R(x)$, we will denote by $\hat{h}^R(R(x))$ the predicted value of Y obtained by rounding $R(x)$ to the nearest integer and matching it with the corresponding observation in the training sample. For example, if $R(x) = 34.6$, then $\hat{h}^R(R(x)) = 35^{\text{th}}$ order statistic of Y in the training sample.

The first rank–based method RFR1 consists of first taking the mean of the predicted ranks and then matching it:

$$\hat{h}_{RFR1} = \hat{h}^R(\bar{R}(x)) \tag{3}$$

where $\bar{R}(x) = (1/K) \sum_{k=1}^{K} \hat{R}_k(x) = \text{mean}(\hat{R}_k(x))$. The second rank–based method, RFRM1 uses the same process as RFR1 but uses the median, instead of the mean, to aggregate the predicted ranks:

$$\hat{h}_{RFRM1} = \hat{h}^R(\tilde{R}(x)) \tag{4}$$

where $\tilde{R}(x) = \text{median}(\hat{R}_k(x))$. The third rank–based method, RFR2, first matches the predicted ranks with the corresponding observations in the training sample for all of the individual trees and then averages them:

$$\hat{h}_{RFR2} = \frac{1}{K} \sum_{k=1}^{K} \hat{h}^R(\hat{R}_k(x)) = \text{mean}(\hat{h}^R(\hat{R}_k(x))). \tag{5}$$

Finally, the fourth and last rank–based method, RFRM2, uses the same process as RFR2 but with the median instead:

$$\hat{h}_{RFRM2} = \text{median}(\hat{h}^R(\hat{R}_k(x))). \tag{6}$$

The first variation, RFM, passes the original data to the basic algorithm but tries to improve robustness by changing the aggregation method (using the median instead of the mean). Consequently, the same trees will be built. The other four variations (RFR1, RFR2, RFRM1 and RFRM2) are using transformed response values and thus the tree building will be affected. The idea is that some types of data contamination might adversely affect the individual trees themselves. Using transformed responses may then alleviate the contamination effect. It is clear that many other transformations are possible. The goal of the present paper is not to investigate that aspect but rather to see if the use of one reasonable transformation (i.e. ranks) can help to improve robustness. Note that ranking also the predictors, or taking any order preserving transformation of them, would be useless since a tree is invariant under monotone transformations of the predictors.

In the simulations and analysis of the UCI data sets reported in Sections 3 and 4, it turns out that RFRM1 and RFRM2 produced almost identical results. Consequently, only the results of RFRM2 are reported and discussed.

2.3 Evaluation Criteria

To assess the accuracy of the six methods under investigation, three criteria are used. These are the Predictive Mean Squared Error (PMSE), the Mean Absolute Prediction Error (MAPE) and the Median Absolute Prediction Error (MedAPE) which are respectively defined as:

$$PMSE = \frac{1}{n}\sum_{i=1}^{n}(h_i - \hat{h}_i)^2,$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|h_i - \hat{h}_i| = \text{mean}(|h_i - \hat{h}_i|)$$

and

$$MedAPE = \text{median}(|h_i - \hat{h}_i|).$$

These criteria are evaluated on independent test sets for the simulation study and through a cross–validation scheme for the real data sets.

2.4 Implementation in R

All simulations are performed in R 2.11 (R Development Team, 2007) using the package randomForest (Liaw and Wiener, 2002). The default settings of the function randomForest are used. In particular, each forest is built with 500 trees (ntree=50), the number of predictors chosen at random in each node is $p/3$ (mtry=$p/3$), and the minimum node size is 5 (nodesize=5). Sample code to compute predictions for the five proposed variations is given in the Appendix.

3 Simulation Study

In this section we compare the predictive performance of the original RF method and its different variations aiming at improving its robustness on simulated data sets. To assess their respective performance and robustness in a variety of situations, simulations are carried out in two parts. The first part is limited to normal error (no–contamination) scenarios in order investigate the methods' behaviors with "clean" data. The second part introduces three different types of contamination to investigate the robustness aspects. A brief description of the process used to generate these scenarios is outlined below. This is followed by the presentation of the results. For every scenario, the simulation is repeated 1000 times and the results are averaged. The training (test) sample size is set to 500 (10000) for each repetition.

3.1 Simulation with Clean Data

The 24 scenarios are issued from a full factorial plan with four factors. For all scenarios, let the vector of predictors $X = (X_1, \ldots, X_6)$ follow a multivariate normal distribution with mean vector 0 and a covariance matrix Σ with 1's on its diagonal and ρ elsewhere and let e be a standard normal error term. Let also m be a parameter used to control the signal to noise ratio.

The first factor is the Data Generating Process (DGP). The first DGP is a binary tree with 7 leaves defined by:

$$\begin{aligned} Y &= m\{I((X_1 \le 0), (X_2 \le 0)) \\ &+ 2I((X_1 \le 0), (X_2 > 0), (X_4 \le 0)) \\ &+ 3I((X_1 \le 0), (X_2 > 0), (X_4 > 0), (X_6 \le 0)) \\ &+ 4I((X_1 \le 0), (X_2 > 0), (X_4 > 0), (X_6 > 0)) \\ &+ 5I((X_1 > 0), (X_3 \le 0)) \\ &+ 6I((X_1 > 0), (X_3 > 0), (X_5 \le 0)) \end{aligned}$$

$$
\begin{aligned}
&+ \quad 7I((X_1 > 0), (X_3 > 0), (X_5 > 0))\} \\
&+ \quad e
\end{aligned}
$$

where I is the indicator function.

The second DGP is a "non–tree" DGP defined by

$$
\begin{aligned}
Y &= m\{X_1 + 0.707X_2^2 + 2I(X_3 > 0) + 0.873\log(|X_1|))X_3 \\
&+ 0.894X_2X_4 + 2I(X_5 > 0) + 0.464\exp(X_6)\} + e
\end{aligned}
$$

where the coefficients are chosen such that each summing term as approximatively a variance of 1 when the predictors are uncorrelated.

The second factor introduced is the variation of the number of predictors. In the first case, only the six relevant predictors are provided. In the second case, 10 unrelated predictors are added to the data. These additional predictors are generated as independent standard normal variables and thus are independent of Y.

The third factor is a variation of the correlation between the six relevant predictors. In the first case, the six relevant predictors are uncorrelated (i.e. $\rho = 0$ and Σ is the identity matrix). In the second case, the predictors are equicorrelated with a correlation of $\rho = 0.5$.

The fourth and last factor consists of three effect sizes, small, medium and large, obtained by varying the parameter m. As the parameter ρ, this parameter modulates the size of the signal with respect to the error. The selected values of m are 0.8 (large), 0.2 (medium), 0.05 (small) for DGP 1 and 0.6 (large), 0.15 (medium), 0.05 (small) for DGP 2. With these values, the signal to noise ratio vary between 1.94 and 3.91 for the large effect scenarios, between 0.12 and 0.24 for the medium effect scenarios and between 0.01 and 0.02 for the small effect scenarios.

A summary containing the main findings is given in Table 1 and Figure 1. For each scenario, Table 1 reports the efficiency of each method with respect to the original RF and the % increase for a given criterion with respect to the best performer in the same scenario. For example, if PMSE_{is} is the PMSE of method i ($i \in \{\text{RF,RFM,RFR1,RFR2,RFRM2}\}$) for scenario s ($s = 1, \ldots, 24$) obtained as the average PMSE over the 1000 simulation runs, then the PMSE efficiency of method k in scenario s with respect to RF is

$$\frac{\text{PMSE}_{ks}}{\text{PMSE}_{RFs}},$$

and the % increase in PMSE of method k in scenario s with respect to the best peformer is

$$100\frac{(\text{PMSE}_{ks} - \min_i\{\text{PMSE}_{is}\})}{\min_i\{\text{PMSE}_{is}\}}.$$

Table 1 presents the mean, median and maximum, over the 24 scenarios, of these values for each criterion. Figure 1 show boxplots of the % increase values over the 24 scenarios. Since all mean efficiencies are greater than 1 (left part of Table 1), the original RF methods performs the best but its advantage is rather small being at most 1.035 for RFR1. The % increase for a given criterion with respect to the best performer proves to be more useful for the comparisons to follow. The original RF method also performs the best with regards to the mean increase with respect to the best performer for all three criteria. However, the two median–based methods, RFM and RFRM2, are very close. For example, on average, RFM has a PMSE which is 2.377% higher than the best performer compared to 0.986% for RF. The performance of RFR1 is highly variable. On one hand, it is the method with smallest median increase for all criteria. On the other hand, it is sometimes very far from the best performer as shown in the boxplots and as indicated by the maximum increase. Consequently, RFR1 has the largest mean increase for all criteria. The RFR2 method is also very close to RF. The main message here is that, except for RFR1 which is more erratic, the other variations of RF offer a good and stable performance over the 24 clean data scenarios.

3.2 Simulation with Contaminated Data

To evaluate the robustness of the suggested variations of RF, a second simulation, where we introduce contaminated data, is carried out. Four of the original 24 scenarios are retained for this part. They are

Table 1: Results for the 24 clean data scenarios. The mean, median and maximum are computed over the 24 scenarios for the efficiency of each method compared to the original RF and for the % increase in the value of the criterion of each method when compared to the best performer for the same scenario.

	Efficiency w/r RF				% increase w/r best performer				
	PMSE				PMSE				
	RFM	RFR1	RFR2	RFRM2	RF	RFM	RFR1	RFR2	RFRM2
mean	1.014	1.035	1.008	1.023	0.986	2.377	4.512	1.778	3.246
median	1.012	1.000	0.999	1.017	0.413	2.660	0.164	1.034	2.471
max	1.033	1.345	1.081	1.117	2.959	5.287	34.51	8.057	11.68
	MAPE				MAPE				
	RFM	RFR1	RFR2	RFRM2	RF	RFM	RFR1	RFR2	RFRM2
mean	1.007	1.013	1.002	1.009	0.500	1.153	1.751	0.720	1.409
median	1.006	1.000	0.999	1.008	0.210	1.302	0.082	0.532	1.230
max	1.016	1.119	1.024	1.038	1.561	2.593	11.91	2.410	3.831
	MedAPE				MedAPE				
	RFM	RFR1	RFR2	RFRM2	RF	RFM	RFR1	RFR2	RFRM2
mean	1.006	1.009	1.001	1.007	0.508	1.127	1.376	0.584	1.227
median	1.006	0.999	0.999	1.008	0.242	1.310	0.036	0.533	1.202
max	1.014	1.083	1.012	1.025	1.658	2.587	8.285	2.480	2.506

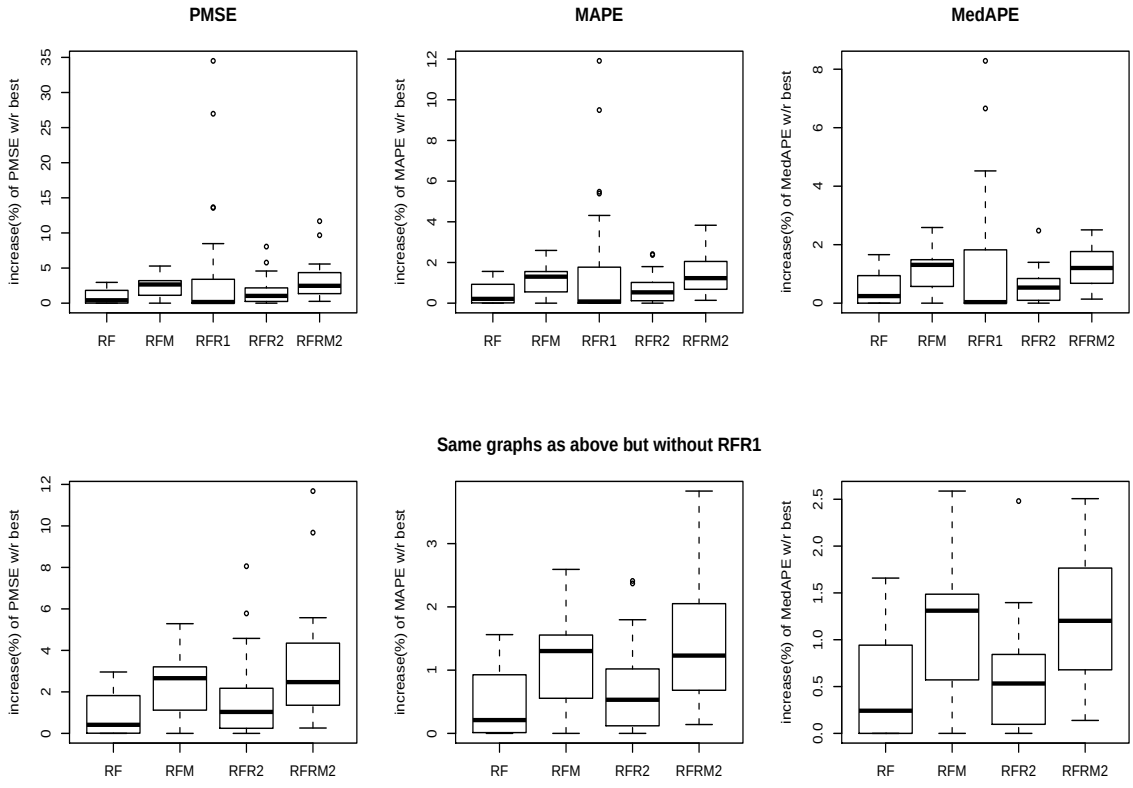

Figure 1: Results for the 24 clean data scenarios. Distribution of the % increase in the value of the criterion of each method when compared to the best performer for the same scenario. The same boxplots are reproduced below without RFR1 to facilitate the comparisons.

the two DGP's, with either large or medium effects, using only relevant and uncorrelated predictors. This is explained by the fact that the first part of the simulation shows that the impact of adding unrelated predictors and/or having correlation between the relevant predictors is rather small. Moreover, the small effect size is left out as the introduction of contamination already reduces the effect size. Three different types

of contamination are introduced in the data. In all cases, p represents the probability of contamination and is set, successively, to 0%, 5%, 10%, 15%, 20% and 25%. The uncontaminated cases of the last subsection correspond to $p = 0$.

The first type is a "variance" contamination where the error term is generated in the following way: with probability $1 - p$, e is a standard normal variate and with probability p, it is a normal variate with mean 0 and variance 25. Both the training and test data sets are generated with this definition of e as this produces a heavier–tailed error distribution in the population.

The second type is a "shuffle" contamination where the responses Y of a randomly selected portion p of the sample in the training data set are permuted. This is only performed in the training data set and not the test set. This contamination imitates a common process used to assess the robustness of methods with a binary (or categorical) response. In this situation, "noise" is added by replacing the value of the categorical response by a value chosen uniformly at random among its other possible values. The shuffle contamination used here is a simple adaptation of this idea.

Finally, the third type is a "shift" contamination where, with probability p, the value of Y is augmented by 10. As for the shuffle contamination, this is done only for the training set.

By crossing the two factors (two GDP and two effect sizes, large and medium), three contamination types and six contamination levels, we obtain a total of 72 scenarios. Only the results for the PMSE are presented and discussed as similar conclusions are reached with the other criteria. The results for the PMSE criterion are presented in Figures 2, 3 and 4. As in the last subsection, the methods are compared using the % increase in a criterion with respect to the best performer.

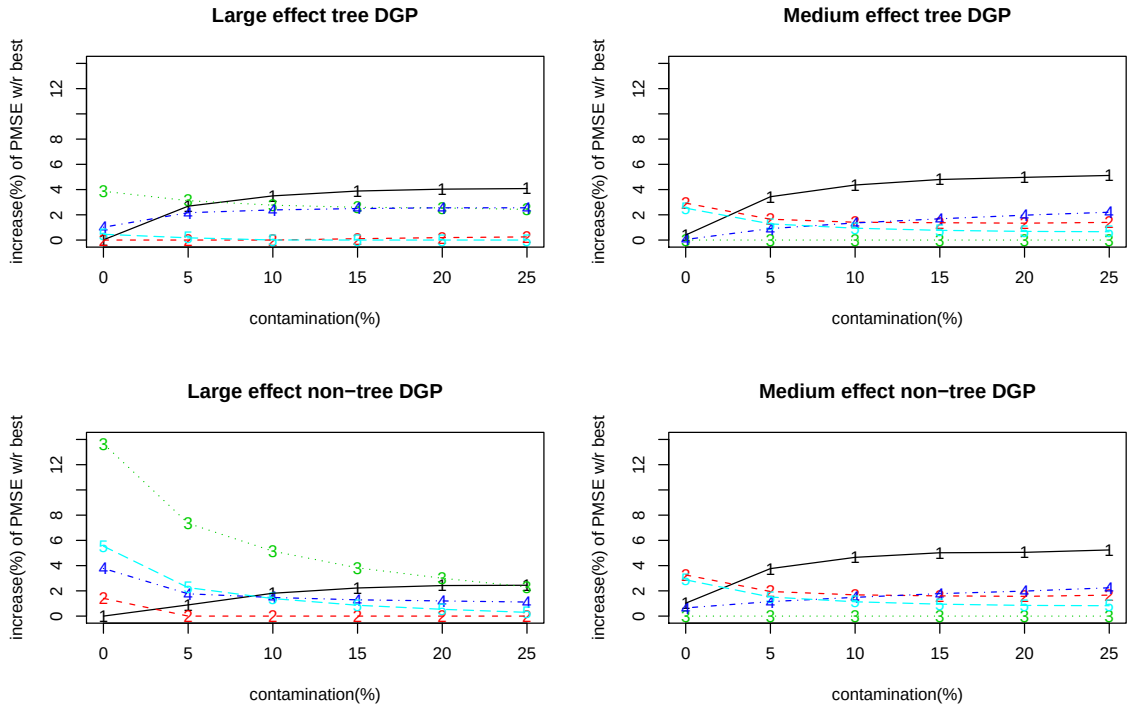

Figure 2: Results for the variance contamination. Behavior of the % increase in PMSE of each method when compared to the best performer for the same scenario as a function of the proportion of contamination. The lines are indexed in the following manner: RF=1, RFM=2, RFR1=3, RFR2=4, RFRM2=5.

The three types of contamination clearly have different impacts, the shift contamination being the one most affecting the methods. Starting with the variance contamination (Figure 2), we see that the original RF seems to be most affected but its % increase in PMSE is not that high, being at 5.24% in the worst case (medium effect non–tree GDP with 25% contamination). RFR1 again has a variable performance, being

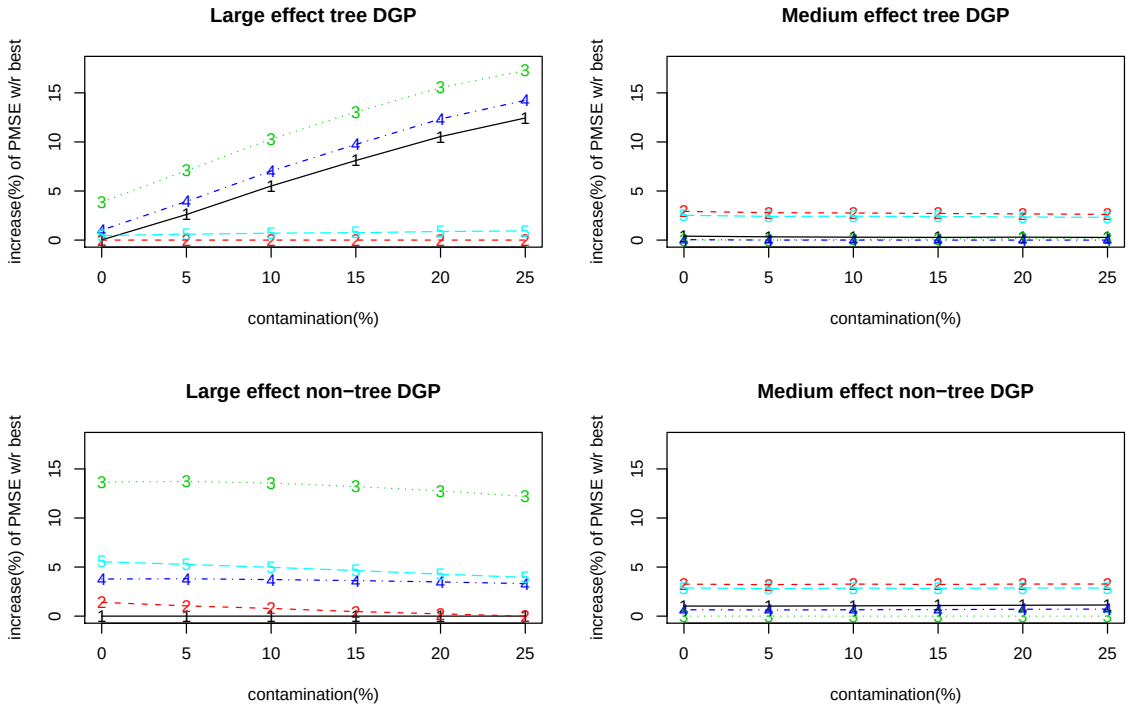

Figure 3: Results for the shuffle contamination. Behavior of the % increase in PMSE of each method when compared to the best performer for the same scenario as a function of the proportion of contamination. The lines are indexed in the following manner: RF=1, RFM=2, RFR1=3, RFR2=4, RFRM2=5.

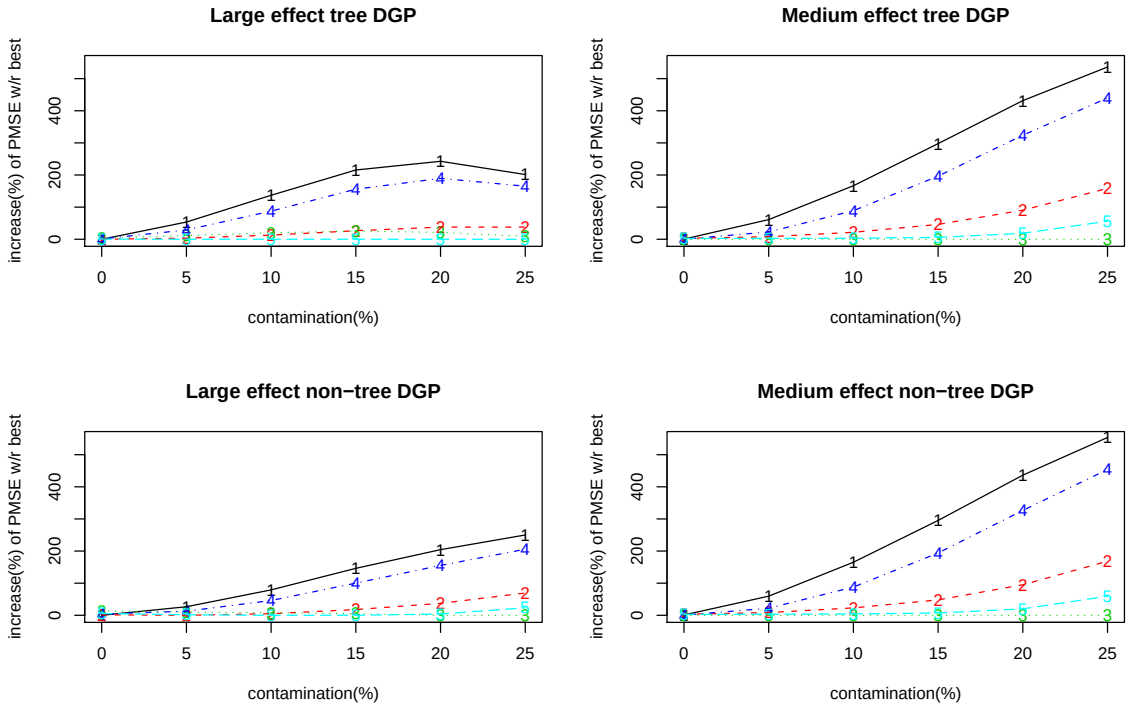

Figure 4: Results for the shift contamination. Behavior of the % increase in PMSE of each method when compared to the best performer for the same scenario as a function of the proportion of contamination. The lines are indexed in the following manner: RF=1, RFM=2, RFR1=3, RFR2=4, RFRM2=5.

good in the medium effect cases (the two right plots) but not for the large effect cases (the two left plots). The median–based methods, RFM and RFRM2, offer stable and good performances.

The median–based methods are again stable and competitive for the shuffle contamination (Figure 3). More precisely, RFM has at worst an increase of 3.26% while the worst increase of RFM2 is 5.54%. All other three methods have an increase of more than 10% in at least two cases.

The shift contamination has a big impact on RF and RFR2 (Figure 4). The increase in PMSE of the original RF reaches 554% in the medium effect non–tree DGP case. The median–based methods and RFR1 are a lot less affected.

From all these simulations, we can conclude that the two median–based methods, RFM and RFRM2, are quite stable and competitive for both clean and contaminated data. RFR1 offers a highly variable performance, being the best performer in some situations and the worst in others. The original RF is very good with clean data but is more sensitive to contamination than the other methods. RFR2 is also good with clean data but is affected by shift contaminations.

4 Real Data Sets

To compare the methods by means of real data, we use ten data sets from the UCI Machine Learning Repository for which regression is the default task. The selected data sets are Auto MPG (AUTO), Breast Cancer Wisconsin (BCWI), Communities and Crime (CCRI), Computer Hardware (COMP), Concrete Compressive Strength (CONC), Concrete Slump Test (CSLU), Housing (HOUS), Parkinsons Telemonitoring (PARK), Servo (SERV) and Wine Quality (WINE). Their characteristics are presented in Table 2. In the data set AUTO, the "car name" predictor was excluded since it contained too many different categorical values. Six out of the 398 observations which contained missing data were also excluded. In the data set CCRI, 25 out of the 128 predictors which contained at least 58% of missing data each, were also excluded. Consequently, there are no missing values in any of the data sets used.

Table 2: Characteristics of the real data sets.

Data set	# data points (n)	# predictors	# of unique response values
AUTO	392	7	127
BCWI	569	32	94
CCRI	1994	103	97
COMP	209	9	116
CONC	1030	9	39
CSLU	103	10	90
HOUS	506	14	229
PARK	5875	26	1129
SERV	167	4	51
WINE	4898	12	7

All the methods are evaluated with these data sets. To assess the performance of RF and its variations, 10–fold cross validation is repeated five times and the results averaged. The results are reported in Table 3 and Figure 5.

Looking at the mean or median % increase with respect to the best performer or at the rankings of the methods, we see that the original RF and RFM are the two best methods if PMSE is the criterion and that the two median–based methods, RFM and RFRM2, are the two top ones if MAPE or MedAPE are the criteria. RFM (RFM2) is even the best performer for eight (seven) of the ten data sets according to MAPE (MedAPE). The boxplots also show that the performances of RFM and RFRM2 are more stable across the data sets.

Table 3: Results with the real data sets. The upper part presents the raw criteria. The mean rank is the average rank when the five methods are ranked among themselves separately for each data set and criterion. The lower part presents the % increase in the value of the criterion of each method when compared to the best performer for the same data set.

	PMSE					MAPE					MedAPE				
Data set	RF	RFM	RFR1	RFR2	RFRM2	RF	RFM	RFR1	RFR2	RFRM2	RF	RFM	RFR1	RFR2	RFRM2
AUTO	7.441	7.363	8.268	7.563	7.582	1.901	1.870	1.943	1.904	1.875	1.301	1.213	1.190	1.289	1.085
BCWI	1114	1210	1142	1107	1201	28.38	29.17	28.26	28.12	28.69	26.12	25.83	25.20	25.42	25.15
CCRI	0.018	0.019	0.021	0.018	0.019	0.093	0.088	0.091	0.091	0.088	0.298	0.050	0.050	0.057	0.050
COMP	3295	3149	7453	4250	6051	25.97	25.49	33.08	27.08	30.04	10.97	10.12	10.00	10.44	10.00
CONC	27.55	25.80	36.92	28.75	26.87	3.841	3.428	4.421	3.918	3.506	3.046	2.375	3.372	3.033	2.429
CSLU	2000	1993	2134	2040	2049	35.81	33.91	36.44	35.99	34.30	32.22	27.62	31.98	30.78	27.63
HOUS	10.79	11.33	16.73	11.51	12.24	2.163	2.105	2.493	2.197	2.162	1.502	1.486	1.560	1.483	1.465
PARK	15.48	7.484	21.56	14.92	7.126	2.875	1.429	3.329	2.842	1.401	2.124	0.625	2.344	2.107	0.608
SERV	0.709	0.600	1.739	0.812	0.863	0.558	0.430	0.676	0.527	0.463	0.358	0.164	0.198	0.227	0.193
WINE	0.349	0.366	0.466	0.359	0.409	0.424	0.333	0.383	0.432	0.342	0.300	0.000	0.000	0.305	0.000
mean rank	2.0	2.0	4.8	2.7	3.5	3.4	1.5	4.3	3.5	2.3	4.6	2.2	3.2	3.5	1.6
# of times best	3	5	0	1	1	0	8	0	1	1	0	5	3	0	7

	% increase of PMSE w/r best					% increase of MAPE w/r best					% increase of MedAPE w/r best				
Data set	RF	RFM	RFR1	RFR2	RFRM2	RF	RFM	RFR1	RFR2	RFRM2	RF	RFM	RFR1	RFR2	RFRM2
AUTO	1.061	0.000	12.30	2.722	2.974	1.620	0.000	3.882	1.802	0.246	19.92	11.81	9.677	18.76	0.000
BCWI	0.711	9.390	3.200	0.000	8.540	0.955	3.736	0.518	0.000	2.033	3.853	2.717	0.199	1.086	0.000
CCRI	0.000	1.093	15.30	0.279	4.918	5.568	0.000	3.409	3.636	0.227	495.2	0.000	0.000	13.60	0.000
COMP	4.635	0.000	136.6	34.96	92.13	1.888	0.000	29.80	6.249	17.85	9.736	1.233	0.000	4.352	0.000
CONC	6.806	0.000	43.14	11.47	4.163	12.06	0.000	28.98	14.29	2.276	28.27	0.000	42.01	27.74	2.295
CSLU	0.306	0.000	7.066	2.343	2.774	5.578	0.000	7.440	6.105	1.146	16.66	0.000	15.79	11.43	0.036
HOUS	0.000	4.989	55.01	6.610	13.39	2.741	0.000	18.40	4.332	2.674	2.491	1.420	6.485	1.222	0.000
PARK	117.3	5.018	202.5	109.3	0.000	105.3	2.042	137.7	102.9	0.000	249.3	2.696	285.3	246.5	0.000
SERV	18.28	0.000	190.0	35.45	43.95	29.92	0.000	57.31	22.50	7.701	118.3	0.000	20.35	38.27	17.31
WINE	0.000	4.875	33.55	2.896	17.35	27.34	0.000	14.92	29.56	2.491	∞	0.000	0.000	∞	0.000
mean**	14.91	2.537	69.88	20.58	19.02	19.29	0.578	30.23	19.14	3.664	104.9	2.208	42.20	40.33	2.182
median	0.886	0.546	38.35	4.753	6.729	5.573	0.000	16.66	6.177	2.154	24.09	0.617	8.081	16.18	0.000

** mean increase of MedAPE w/r best is computed without the WINE data set since the best has an error of 0 and this gives an infinite increase for the methods with a greater than 0 error.

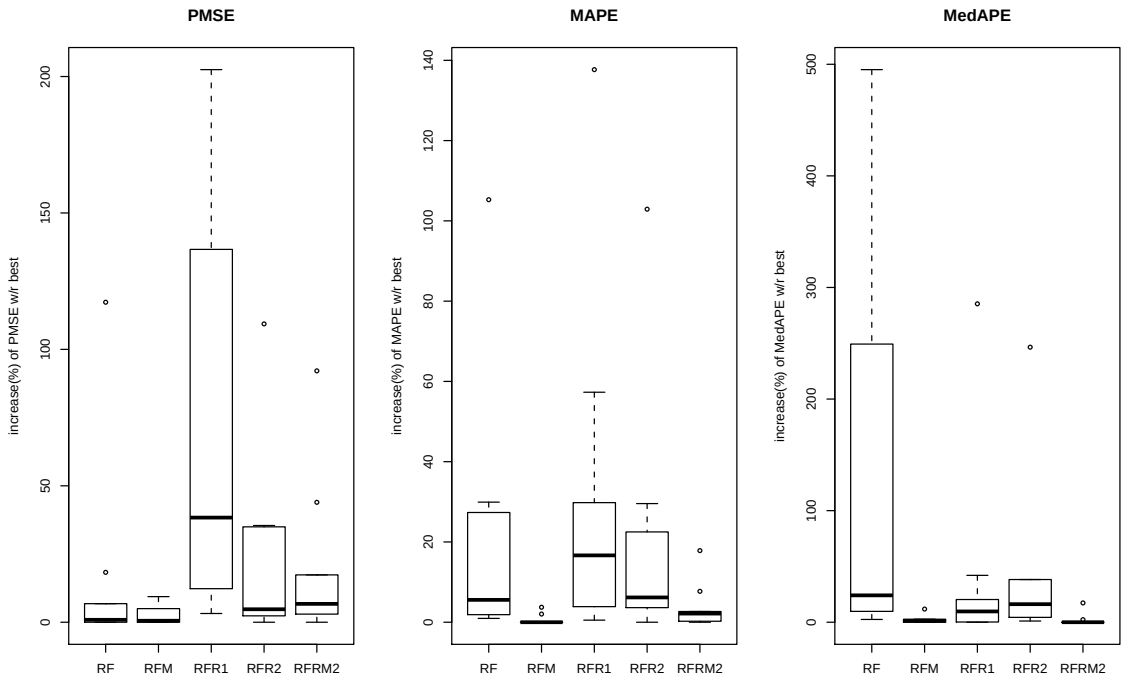

Figure 5: Results for the ten UCI data sets. Distribution of the % increase in the value of the criterion of each method when compared to the best performer for the same data set.

5 Concluding Remarks

The goal of this study was to investigate the robustness of the random forests method for regression problems. Five variations of the original RF method are proposed and all of them can be easily implemented. In the light of the simulation study and the evaluation of the methods with ten real data sets, we can conclude that the original RF is very good with clean data but is generally more sensitive to contamination than the proposed variations. The three median–based methods stand out as good alternatives. The first one, RFM, takes the median of the individual tree predictions instead of the mean. The second and third ones, RFRM1 and RFRM2, which produced nearly equivalent results, also use the median but use the ranks of the response to construct the trees. These three methods are stable and competitive for both clean and contaminated data in the simulation study and they also performed very well with real data sets. This, combined with their very low implementation cost, should make them routinely computed along with the original RF method in order to obtain a certain amount of protection against contamination.

The methods proposed in this paper are simple ways to improve robustness. Here we transformed the responses by taking their ranks. Future studies could investigate the benefits of other transformations and maybe imagine a way for the data to select the best transformation adaptively. Another possibility to improve robustness would be to robustify the tree building process itself. All the methods proposed in this paper use the original regression tree least–squares splitting rule. The mean absolute deviation could be used instead but this would require more work to implement as the randomForest function of R could not be used directly.

6 Appendix

6.1 Computations of the six types of predictions with R

In the following, **xtrain** and **ytrain** are the matrix of predictors and vector of responses, respectively, for the training data set and **xtest** is the matrix of predictors of **ntest** new data for which predictions are needed.

```
# forest with the original responses
rforiginal=randomForest(xtrain,ytrain)
# extracts the individual tree predictions
predoriginalall=predict(rforiginal,newdata=xtest,predict.all=TRUE)$individual

# RF predictions
rf=apply(predoriginalall,1,mean)

# RFM predictions
rfm=apply(predoriginalall,1,median)

# computes the ranks of the responses and sorts the responses for the matching
rytrain=rank(ytrain)
ytrainsort=sort(ytrain)
# forest with the ranks of the responses
rfrank=randomForest(xtrain,rytrain)
# extraction of the predicted ranks of the individual trees
predrankall=predict(rfrank,newdata=xtest,predict.all=TRUE)$individual

# RFR1 predictions
rfr1=apply(predrankall,1,mean)
rfr1=round(rfr1)
rfr1=ytrainsort[rfr1]

# RFRM1 predictions
rfrm1=apply(predrankall,1,median)
rfrm1=round(rfrm1)
rfrm1=ytrainsort[rfrm1]

# matching of the individual predicted ranks with the original responses
predrankall2=round(predrankall)
predrankall2=matrix(ytrainsort[predrankall2],ntest,500,byrow=FALSE) # 500 is the number of trees in the forest

# RFR2 predictions
RFR2=apply(predrankall2,1,mean)

# RFRM2 predictions
RFRM2=apply(predrankall2,1,median)
```

References

[1] Breiman, L. (2001). Random Forests. *Machine Learning* **45**, 5–32.

[2] Folleco, A., Khoshgoftaar, T.M., Van Hulse, J. and Bullard, L. (2008). Software Quality Modeling: The Impact of Class Noise on the Random Forest Classifier. *IEEE Congress on Evolutionary Computation*, Volumes 1-8, 3853–3859.

[3] Frank, A. and Asuncion, A. (2010). *UCI Machine Learning Repository*. Irvine, CA: University of California, School of Information and Computer Science. [http://archive.ics.uci.edu/ml]

[4] Hamza, M. and Larocque, D. (2005). An Empirical Comparison of Ensemble Methods Based on Classification Trees. *Journal of Statistical Computation and Simulation* **75**, 629–643.

[5] Liaw, A. and Wiener, M. (2002). Classification and Regression by randomForest. *R News* **2**, 18–22.

[6] R Development Team (2007). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing: www.R-project.org.

[7] Rokach, L. (2008). Taxonomy for Characterizing Ensemble Methods in Classification Tasks: A Review and Annotated Bibliography. *Computational Statistics and Data Analysis* **53**, 4046–4072.

[8] Siroky, D.S. (2009). Navigating Random Forests and Related Advances in Algorithmic Modeling. *Statistics Surveys* **3**, 147–163.