مجله‌ی پژوهش‌های آماری ایران
پژوهشکده‌ی آمار
دوره‌ی ۳، شماره‌ی ۱، بهار و تابستان ۱۳۸۵، صص ۱-۲۱

J. Statist. Res. Iran **3** (2006): 1-21

Outlier Detection by Boosting Regression Trees

Nathalie Chèze and Jean-Michel Poggi*

Paris-Sud University

Invited Paper

Abstract. A procedure for detecting outliers in regression problems is proposed. It is based on information provided by boosting regression trees. The key idea is to select the most frequently resampled observation along the boosting iterations and reiterate after removing it. The selection criterion is based on Tchebychev's inequality applied to the maximum over the boosting iterations of the average number of appearances in bootstrap samples. So the procedure is noise distribution free. It allows to select outliers as particularly hard to predict observations. A lot of well-known bench data sets are considered and a comparative study against two well-known competitors allows to show the value of the method.

Keywords. boosting; CART; outlier; regression.

1 Introduction

We address the problem of detecting outliers in a given sample in regression problems. Outliers are generally considered as observations which are not "consistent" with most of the data. Many interesting ideas can be found around a classical way to detect them by considering PCA and related methods dealing with stability, sensitivity and robust estimation of the principal components (see Jolliffe, 2002, ch. 10, for a review).

The book of Rousseeuw and Leroy (1987) contains an overview of outlier detection problems in the regression context and several methods are described

* Corresponding author

and proposed. The underlying model, the estimation method and the number of outliers with respect to the number of observations lead to define various kinds of outliers. For example, one can consider different ways of contamination: outliers in the response space, outliers in the covariate space or outliers in both spaces. Another point of view is to consider as outliers not only atypical observations but also the observations coming from a second population.

Many methods have been developed to cope with such situations. They are essentially supported by robustness ideas and are based on linear modeling (see for example, Rousseeuw and Leroy (1987), Pena and Yohai (1999), or Verboven and Hubert (2005) for a short software-oriented review). Classical methods involve robust estimators of the covariance matrix or of the linear fit like the Minimum Covariance Determinant (MCD) estimator (see Rousseeuw and Van Driessen, 1999) or the Least Trimmed Squares (LTS) estimator (see Rousseeuw and Leroy, 1987) or the Least Median of Squares (LMS) estimator (see Rousseeuw, 1984). Of course, these approaches suffer from the restriction of the outlier definition related to deviations with respect to the linear model. More generally, the outlier definition depends on a given parametric regression design method.

The aim of this paper is to propose a procedure based on boosting and such that:

- the regression design method is nonparametric and able to explore different features of the data by adaptive resampling;
- the detection is entirely automatic and the associated parameters are data-driven;
- it is possible to detect outliers in the response direction as well as in the covariate space.

A classical remark about the boosting procedure AdaBoost (introduced for classification problems by Schapire et al. (1998), and then for regression ones by Drucker (1997)) and its variants, is its sensitivity to outliers. This property is in general identified as a drawback, but it can be used (see Gey and Poggi, 2005) to improve the model estimated by a given estimation method to be better adapted to particularly hard observations. The goal is here to use it to detect outliers. Our procedure is based on the information provided by the adaptive resampling process generated when boosting CART regression trees is used. This adaptive process tells us a lot about the data set and this is one of the most attractive feature of the boosting from the data analytic point of view.

Let us recall that the boosting algorithm generates a sequence of regression function estimates, when the observations are supposed to come from a

nonlinear regression model of the form (1). Each estimator of this sequence fits a bootstrap sample obtained from the original training sample by adaptive resampling, highlighting the observations poorly predicted by its predecessor in the sequence. It turns out that such a resampling leads to focus on hard observations with respect to the chosen estimation method, that is to focus on more often badly predicted observations.

Of course an outlier is such an observation. So the adopted strategy is two stages: the first highlights the hard observations and the second selects among them the outliers. The key idea of the first stage is to retain the most frequently resampled observation along the boosting iterations and reiterate after removing it, while the second stage defines a data-driven confidence region to select outliers.

This paper is organized as follows. In Section 2, the principle of CART regression trees and the boosting for regression problems, are briefly recalled. The outlier detection procedure is introduced and motivated in Section 3. The Section 4 is dedicated to experimental results of the application of the detection procedure to real and artificial data sets and to compare it with some well-known competitors. Finally some concluding remarks are collected in Section 5.

2 CART Regression Trees and Boosting

Let us consider the following regression model:

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

where $(X, Y) \in \mathbb{R}^p \times \mathbb{R}$, f is the unknown regression function, and ξ is an unobservable additive noise centered conditionally to X with unknown variance σ_ξ^2.

Let us consider a sample L of size n composed of realizations of the variable (X, Y) and possibly some outliers.

2.1 CART Regression Trees

We focus on CART regression trees to generate estimators of f, generically denoted by $\hat{f}$ in the sequel. Since we are mainly interested in the sequence of resampling probabilities produced by the boosting sequence, a particularly attractive property of this estimation method is, in this context, its instability. Therefore, the booststrap regression trees do not have the same number of terminal nodes for random perturbations of the sample L and involve different features of the data.

We use here the well-known CART for regression (see Breiman et al., 1984) allowing to construct from L an estimator $\hat{f}$ of f having low generalization error. Since the joint distribution is unknown, the resubstitution error is used to generate the models and the prediction error of $\hat{f}$ is evaluated (and then the final tree is chosen) using a 10-fold cross validation, when the number of observations is sufficient (more precisely when the sample size is greater than 100). Otherwise, the maximal tree is retained.

2.2 Boosting

The aim of boosting is to improve the performance of the chosen estimation method (here CART) by generating a sequence of estimators using a suitably chosen adaptive resampling scheme, and aggregating them. Starting from the uniform distribution, the current sampling distribution is modified in such a way that each estimator copes with a bootstrap sample obtained from the original one by adaptive resampling, highlighting the observations poorly predicted by its predecessor in the sequence.

The AdaBoost algorithm for classification problems, has been proposed by Freund and Schapire (1997) and really impressive results are obtained for prediction purposes. Some papers partially elucidate the surprisingly good behaviour of this algorithm in the classification context (let us mention among others Schapire et al., 1998; Friedman et al., 2000).

Drucker (1997) provides a direct adaptation of AdaBoost to the regression framework, which exhibits interesting performance by boosting CART regression trees (see Borra and Di Ciacco, 2000; Gey and Poggi, 2005). The boosting algorithm used here can be found in Table 1. Let us mention that, since usually boosting is used to design an aggregated estimator, the final output (omitted here) is $\tilde{f}$, the median of $\left(\hat{f}_k\right)_{1\leqslant k\leqslant K}$ weighted by $\left(\log\left(\frac{1}{\beta_k}\right)\right)_{1\leqslant k\leqslant K}$.

In addition, let us mention that $K = 50$, the value selected here for the number of iterations within the boosting loop, is sufficiently large according to experimental results about AdaBoost algorithm stabilization, both for classification and regression cases (see Breiman, 1998; Gey and Poggi, 2005).

3 Outlier Detection Procedure

The proposed outlier detection procedure has two stages. The first step is iterative. At each boosting iteration, we retain the most frequently resampled observation along the iterations and reiterate after removing it. We obtain then

Table 1. Boosting algorithm

	$[M,i_0] = \text{boost}(L, K)$
Input:	L: the sample of size n and K the number of iterations
Initialization:	Set $p_1 = D$ the uniform distribution on $\{1, \ldots, n\}$
Loop:	for $k = 1$ to K do
step 1	• randomly draw from L with replacement, according to p_k, a sample L_k of size n,
step 2	• using CART, construct an estimator $\hat{f_k}$ of f from L_k,
step 3	• set from the original sample L: $i = 1, \ldots, n$ $l_k(i) = \left(Y_i - \hat{f}_k(X_i)\right)^2$ and $\epsilon_{p_k} = \sum_{i=1}^{n} p_k(i) l_k(i)$, $\beta_k = \epsilon_{p_k} / \left(\max_{1 \leqslant i \leqslant n} l_k(i) - \epsilon_{p_k} \right)$ and $d_k(i) = l_k(i) / \max_{1 \leqslant i \leqslant n} l_k(i)$, $p_{k+1}(i) = \beta_k^{1-d_k(i)} p_k(i)$, normalize p_{k+1} to be of sum 1,
step 4	• compute $I_{i,k}$ the number of times the observation i appears in L_k
Output:	$M = \max_{i \in L} S_i, \quad i_0 = \text{argmax}_{i \in L} S_i$, where $S_i = \frac{1}{K} \sum_{k=1}^{K} I_{i,k}$

a set H containing the hard observations with respect to CART (the considered estimation method) and, in addition, sufficiently many typical observations. In the second stage, we select among them the outliers by defining a data-driven confidence region. The detailed outlier detection procedure can be found in Table 2. We emphasize that it is noise distribution free.

For sure, the second stage needs to be motivated. To construct the rejection region $(M_j > C_\alpha)$ of stage 2, let us assume that J is chosen sufficiently large so that it exists an integer j_0 such that $(M_j)_{j=j_0,\ldots,J}$ are not associated with indices corresponding to outliers and then can be assimilated to weakly dependent identically distributed observations of common mean m and variance σ^2. The plot on the top right of Figure 3 shows a simulated example for which $j_0 = 5$ and $J = 30$.

For each $j \in (1, ..., J)$, let us assimilate the outlier detection problem to the individual test of the null hypothesis:

$$H_0 : \textit{The observation } i(j) \textit{ is not an outlier,}$$

against the alternative hypothesis:

$$H_1 : \textit{The observation } i(j) \textit{ is an outlier.}$$

Table 2. Outlier detection algorithm

	Outlier detection
Input:	J: the number of applications of boosting. L: the initial sample, α: the significance level of confidence interval, and K: the number of iterations of each boosting.
Initialization:	Set $L^1 = L$
Stage 1:	for $j = 1$ to J do $[M_j, i(j)] = \text{boost}(L^j, K)$; $L^{j+1} = L^j \setminus i(j)$; $H = L \setminus L^J$
Stage 2:	Outliers are defined as the observations of index $i(j) \in H$ such that $(M_j > C_\alpha)$

Since if $i(j)$ is associated to an outlier then M_j is large, it is consistent to choose the rejection region W of the form $W = (M_j > C_\alpha)$ for a given level of significance α. By applying Tchebychev's inequality to M_j under H_0, we obtain:

$$P_{H_0}\left(\frac{M_j - m}{\sigma} > \sqrt{\frac{1}{\alpha}}\right) \leqslant P_{H_0}\left(\frac{|M_j - m|}{\sigma} > \sqrt{\frac{1}{\alpha}}\right) \leqslant \alpha,$$

and then deduce C_α after estimating unknown parameters leading to (2).

The gap between M_j and $m = E_{H_0}(M_j)$ under H_1 allows to circumvent the usual Tchebychev's inequality conservativeness. Indeed, even if

$$P_{H_0}\left(|M_j - m| > \sigma\alpha^{-1/2}\right) << \alpha,$$

leads to shrink the rejection region, the hypotheses to be tested are sufficiently separated to correctly select the outliers. In addition, let us remark that if one think to α as an individual level in the interval $[0.05, 0.25]$, the key quantity $\sqrt{1/\alpha}$ controlling the bound (see Equation 2) varies between 2 and 4.5, which generates quite small variations with respect to $(M_j - m)$ for $i(j)$ associated to an outlier. In the sequel, we use $\alpha = 5\%$ for all the computations. The unknown parameters m and σ^2 must be estimated under H_0. An entirely data-driven solution is to use robust estimators denoted respectively by $\hat{m}_{rob}$ and $\hat{\sigma}^2_{rob}$. Then, we select the outliers in the set H by taking the elements $i(j)$ for which $(M_j > C_\alpha)$ with

$$C_\alpha = \hat{m}_{rob} + \frac{\hat{\sigma}_{rob}}{\sqrt{\alpha}}. \tag{2}$$

Finally, let us make some additional remarks by answering three questions:

- *Why boosting instead of bagging?*

  The bagging procedure can be defined as a special case of boosting by defining $L_k \equiv L$ and $p_k \equiv D$ the uniform distribution on $\{1, \ldots, n\}$ in Table 1. So the bagging seems to be a good candidate to build a robust estimate of the regression function. In fact, it is true only when the number of outliers is small with respect to the sample size. An idea could be to select observations badly predicted by this robust estimate and threshold the residuals according to the noise distribution. Of course, this strategy requires assumptions about it.

- *Why to reiterate boosting?*

  As a matter of fact, the j_0 most frequently resampled observations along the iterations of a single boosting are different from the first j_0 observations selected stepwise using J boosting reiterations. The reason is that for a given boosting application, the most frequently resampled observation would mask other hard observations.

- *How to choose the number of boosting iterations?*

  Boosting is reiterated until all the outliers have been removed and in addition a sufficient number of observations non-contaminated by outliers are available to estimate the mean and variance under H_0 to plug in the Tchebichev's inequality. When n is not too large, a convenient choice for J is to take the integer part of $0.75n$.

The lack of theoretical results (due to the difficulty to handle analysis of boosting) necessitates an extended experimental study to evaluate the performance of our method.

4 Experimental Results

We examine various well-known bench data sets allowing to study the behavior of the proposed method for various kinds of outliers depending on the way of contamination, for various sample sizes including small ones (which could be critical for nonparametric estimation method) as well as larger ones.

The results are organized following four paragraphs in this section. Many figures illustrate typical situations. Each figure contains four plots: at the top left, the relevant data are displayed (a legend specifies the concerned useful data); at the top right, the plot represents the value of M_j for $1 \leqslant j \leqslant J$

(defined in Table 2) obtained by our method (using $\alpha = 5\%$); at the bottom, two plots give the results obtained by two alternative methods, Least Trimmed Squares (LTS) and Minimum Covariance Determinant (MCD). The estimates $\hat{m}_{rob}$ and $\hat{\sigma}_{rob}$ in (2) are obtained from the MCD estimators applied to $(M_j)_{1 \leqslant j \leqslant J}$. These estimates and the results obtained by these alternative methods have been carried out using the library LIBRA (Verboven and Hubert, 2005) developed using MATLAB® (for a SAS® counterpart see Chen, 2002). We use for each method, the default values for the corresponding parameters.

For our method and the MCD one, outliers are indices associated with points located upside the solid horizontal line while for the LTS method, outliers are located outside the interval delimited by two horizontal lines. In addition, we indicate, for simulated data sets, the indices of outliers and for real data sets, those of some observations chosen to facilitate the interpretation of the plots. Let us remark that in the plot corresponding to our method only J points are drawn while for the two other methods, all the points are present.

4.1 Outliers in Small Size Real Data Sets

Outlier detection is a well-documented topic and, thanks to the book of Rousseeuw and Leroy (1987), a lot of interesting and intensively studied real examples of small sample size, have been examined during twenty years by many authors. All but a few examples extracted from this book are available from the website *http://www.uni-koeln.de/themen/statistik/data/rousseeuw/*. In the sequel, we refer to a specific data set by giving the corresponding page number in the above mentioned book (the same convention is also used in the website).

We apply our method to all the examples (see Table 5 in Appendix). In this section, we have taken the results given by the LTS method as a reference, because it has been considered as a convenient one for such small data sets (see for example, discussions of Rousseeuw and Leroy, 1987). The main conclusion is that in many cases, we obtain results very close to those obtained using MCD and LTS methods in spite of the small sample size (around twenty for most of these data sets) and the parametric model. More precisely, we obtain unsuccessful results for only three examples among eighteen. For the others, we obtain always satisfactory detection with partial or total selection.

Let us be a little more precise by examining some examples of each typical situation.

4.1.1 Why the Method Can Fail?

First of all, let us focus on the three examples for which the method fails. A careful examination of the decision trees leads to easily explain this drawback:

when CART creates a single node containing all the outliers, the method cannot highlight them.

Figure 1 illustrates such a situation: as it can be seen at the top left plot, the four outliers (identified using LTS) of indices 11, 20, 30 and 34, are atypical both in response and covariate directions. We detect only two of them, LTS captures the four and MCD identifies the four same observations plus three others. The explanation is that CART is sufficiently flexible to create a node containing the four outliers which are atypical in a similar way: their X-values are close to each other and far from the other observations, and their Y-values are the four first maxima. Let us observe that, along the iterations of the detection algorithm (see the top right plot), as soon as the observations 34 and 30 are suppressed from the learning sample, outliers of index 20 and 11 are then easily detected.

4.1.2 Examples of Correct Detection

Second, when the percentage of outliers is less than 10%, our method performs correctly except, of course, when the above mentioned drawback occurs. A first example without outliers is given by Figure 2. Our method performs correctly as well as the two other methods. The second example (see Figure 3) exhibits interesting behaviour and highlights an important difference with MCD and LTS methods.

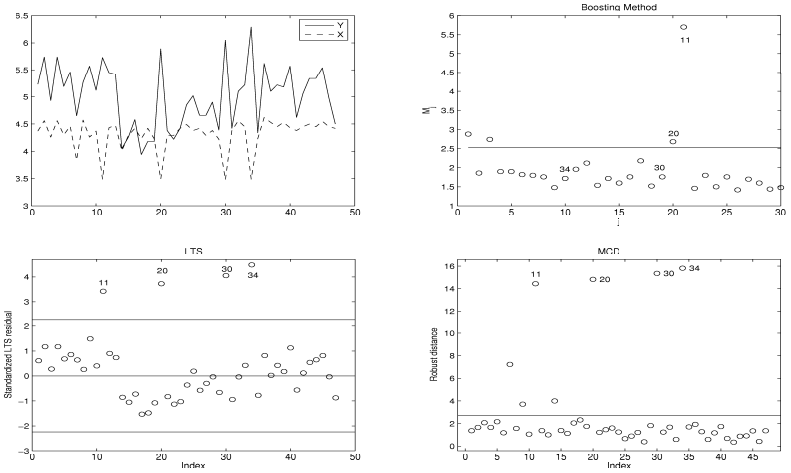

Figure 1. Data set page 27 in Rousseeuw and Leroy (1987), $n = 47$, $p = 1$, $n_{out}^{LTS} = 4$. Our method fails partially and detects only two outliers among four. This comes from the fact that CART creates a single node containing all the outliers.

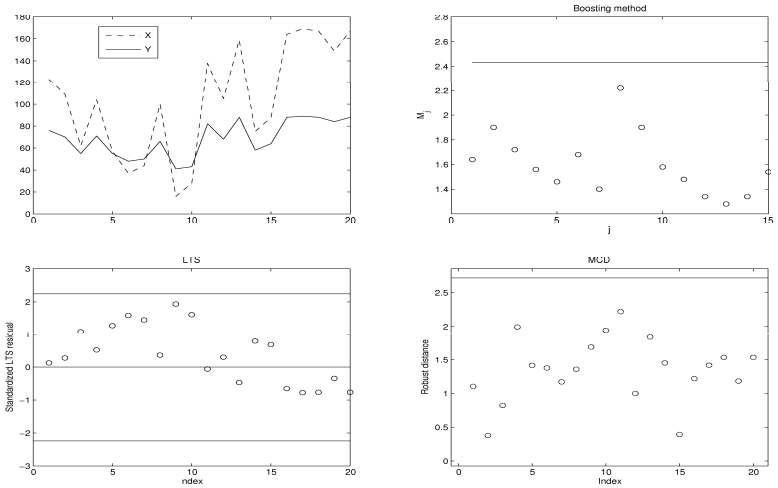

Figure 2. Data set page 22 in Rousseeuw and Leroy (1987), $n = 20$, $p = 1$, $n_{out}^{LTS} = 0$. Our method performs correctly on this data set without outliers.

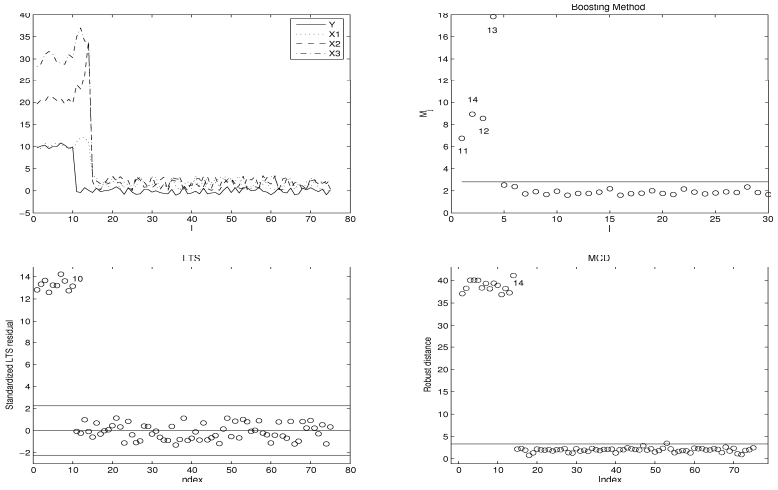

Figure 3. Data set page 94 in Rousseeuw and Leroy (1987), $n = 75$, $p = 3$, $n_{out} = 4$. Our method detects correctly the four outliers without any false detection while the two other methods assimilate the first population to outliers. MCD detects the outliers but LTS fails.

Since it is the only one simulated in Rousseeuw and Leroy (1987), the number of "true" outliers is known and equals to 4. The top left plot shows that the sample can be divided in three parts, two different populations and the outliers: the observations of index from 1 to 10, those of index greater than 15 and the four outliers from 11 to 14. Our method detects correctly the four outliers without any false detection while the two other methods assimilate the first population to outliers. MCD detects the outliers but LTS fails.

4.1.3 Examples of Good Selection but Poor Detection

Third, when the percentage of outliers is greater than 10%, the outliers are brought at the top of the set H but the threshold is too large to automatically select all the outliers. Let us examine two examples.

Figure 4 shows a perfect detection for both MCD and LTS methods, while our method fails to correctly detect the seven outliers which are the observations of index from 15 to 20, as it can be seen in the top left plot showing the sample values of the explained variable Y. Nevertheless, the boosting method selects correctly the outliers: the top eight values of the set H do contain all the outliers but the bound is too large. This comes from the following fact: $n = 24$ and $J - j_0 = 19 - 6$ are too small to have a sufficient number of observations to conveniently estimate the unknown parameters involved in the detection region definition (see (2)).

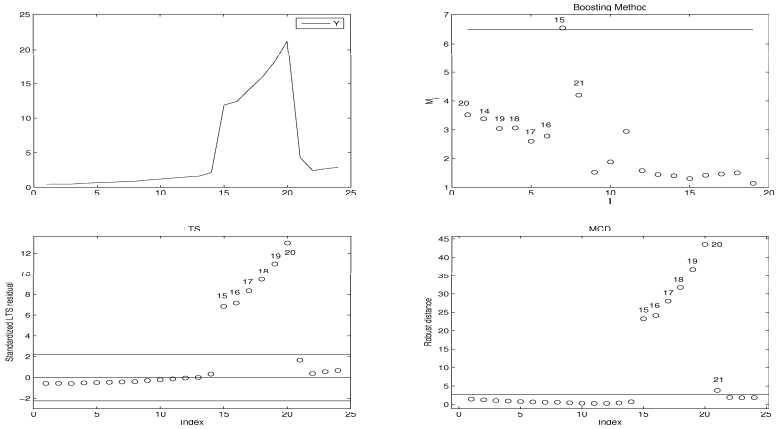

Figure 4. Data set page 26 in Rousseeuw and Leroy (1987), $n = 24$, $p = 1$, $n_{out}^{LTS} = 6$. Perfect detection for both MCD and LTS methods, while our method fails to correctly detect the six outliers which are the observations of index from 15 to 20. Nevertheless, the boosting method selects correctly the outliers: the top eight values do contain all the outliers.

A second example of such a situation is given by the Figure 5. The MCD and LTS methods detect the two outliers corresponding to indices 2 and 18. The observation 19 is detected by MCD. Our method selects first the three outliers but the bound is slightly too large and then only one outlier is detected. Of course at the naked eye, one would obviously select the three outliers.

Let us remark that, by taking 20% instead of 5%, the three outliers are correctly selected and detected.

4.2 Outliers in Simulated Data Sets of Large Size

Table 3 presents three large sample size ($n = 500$) simulated data generated from models FR#1, FR#2, FR#3 used in Gey and Poggi (2006) and first considered by Friedman (1991). They exhibit different difficulties with respect to CART regression trees: model FR#1 is defined using a simple nonlinear function but involves five useless variables, models FR#2 and FR#3 correspond to highly nonlinear functions with strong interactions.

For each model, some outliers have been introduced. We present some selected and typical examples among numerous numerical trials. The main conclusion is that the boosting method performs very well (and much better

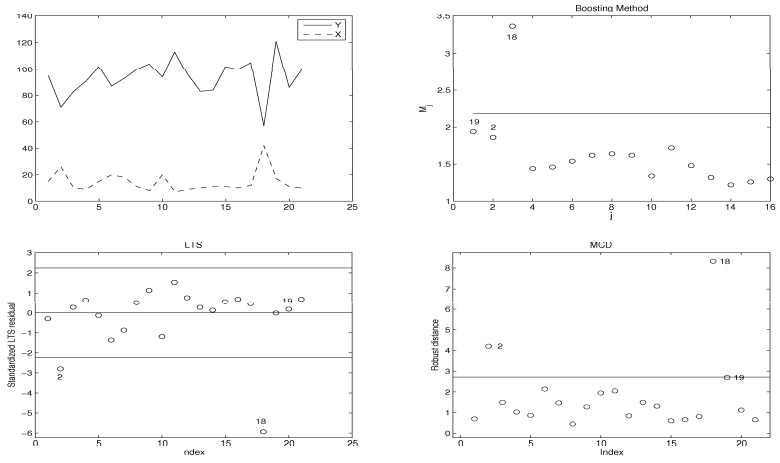

Figure 5. Data set page 47 in Rousseeuw and Leroy (1987), $n = 21$, $p = 1$, $n_{out}^{LTS} = 2$. MCD and LTS methods detect the two outliers 2 and 18. The observation 19 is detected by MCD method. Boosting method selects first the three outliers but the bound is slightly too large and then only one outlier is detected. Of course at the naked eye, one would obviously select the three outliers.

Table 3. Outliers and simulated data sets

Data	Predictors	Regression function f	Noise
FR#1	$X_i \sim \mathcal{U}([0,1])$ $i = 1, \ldots, 10$	$10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2$ $+10x_4 + 5x_5 + 0\sum_{i=6}^{10} x_i$	$N(0,1)$
FR#2	$X_1 \sim \mathcal{U}([0,100])$ $X_2/2\pi \sim \mathcal{U}([20,280])$ $X_3 \sim \mathcal{U}([0,1])$ $X_4 \sim \mathcal{U}([1,11])$	$\sqrt{x_1^2 + \{x_2 x_3 - (1/x_2 x_4)\}^2}$	$N(0,\sigma_\xi^2)$ such that signal-to-noise ratio about 3/1
FR#3	Same as FR#2	$\tan^{-1}\left(\frac{x_2 x_3 - (1/x_2 x_4)}{x_1}\right)$	Same as FR#2

than the two considered competitors which are obviously not well suited to deal with such nonlinear situations; the results provided by these two methods are only given for illustration).

The first example is given by Figure 6. The top left plot shows the Y values containing four outliers located at time instants: $90, 110, 250, 380$. It should be noted that the outliers are not detectable at the naked eye (the same occurs

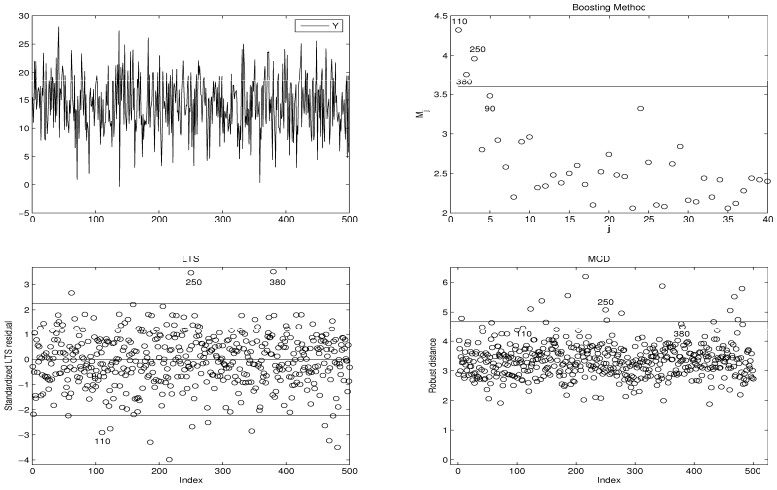

Figure 6. Model FR#1, $n = 500$, $n_{out} = 4$ outliers located at time instants: 90, 110, 250, 380. The LTS and the boosting methods detect correctly three outliers. But LTS generates some false detections and the MCD method fails to identify the outliers and leads to too many false detections. In addition our method selects correctly the last outlier at the fifth position in the set H.

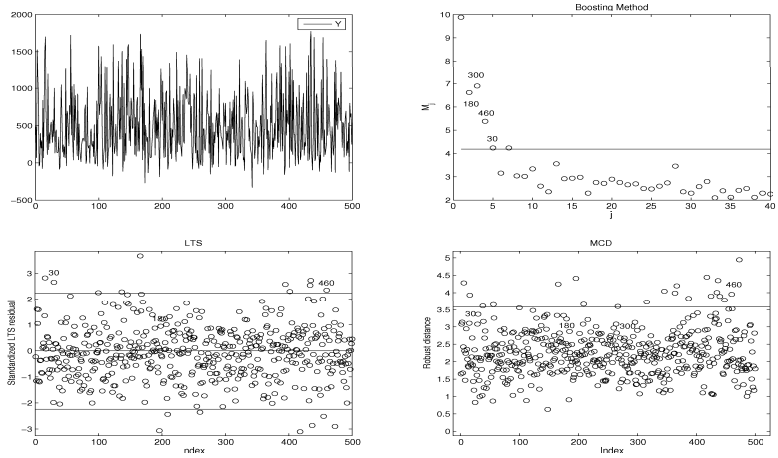

Figure 7. Model FR#2, $n = 500$, $n_{out} = 4$ outliers located at time instants: 30, 180, 300, 460. Our method detects correctly the four outliers and generates only one false detection. MCD and LTS detect only one or two outliers. LTS generates seven false detections and the MCD method leads to too many false detections.

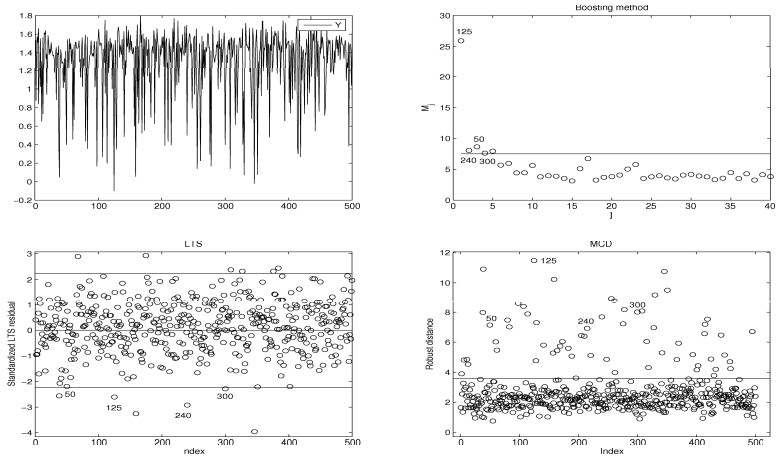

Figure 8. Model FR#3, $n = 500$, $n_{out} = 4$ outliers located at time instants: 50, 125, 240, 300. MCD method generates too many false alarms. LTS method detects correctly the outliers but the price to pay is to diagnose erroneously atypical observations. The boosting method highlights the four outliers with only one false detection.

for the two other examples, see Figures 7 and 8). The LTS and the boosting methods detect correctly three outliers. But LTS generates some false detections and the MCD method fails to identify the outliers and leads to too many false detections. In addition our method selects correctly the last outlier bringing it at the fifth position in the set H of hard observations.

The second example is considered in Figure 7. Our method detects correctly the four outliers and generates only one false detection. MCD and LTS detect only one or two outliers. LTS generates seven false detections and the MCD method again leads to too many false detections. A similar example is given by the plots of Figure 8 confirming that in the presence of highly nonlinear components in the regression function, the MCD method is useless and the LTS method can detect correctly the outliers but the price to pay is to diagnose erroneously atypical observations. Again the boosting method highlights the four outliers with only one false detection.

4.3 A Small Size AR Example with Outliers

In this section, we try to apply our detection algorithm to the time series framework for which, of course, it is not designed. Let us consider a simple time series model assuming that the order is known. An $AR(3)$ model is considered by Justel et al. (2001) and two time series are involved: a single small size realization denoted by $(Z_t)_{1\leqslant t\leqslant 50}$ and a contaminated version which exhibits two kinds of outliers (these two signals are available from the website *http:/www.uam.es/personal_pdi/ciencias/ajustel/jpt.html*). The first one is classical and consists of an additive perturbation at instant 24 while the second one is innovative (a constant is added to the innovation term in the AR equation, see Catoni and Karioti (2004) for a specific development of a time series oriented approach) and located at instant 35 which may generate atypical events up to instant 38. As usual in the time series situation, we set $Y_t = Z_t$ and $X_t = (Z_{t-1}, Z_{t-2}, Z_{t-3})$, since the order is known.

To illustrate the behavior with respect to false detections, let us first consider the situation under H_0. Figure 9 shows that the boosting method works correctly and detects none. At the contrary, two observations are borderline for the LTS detection limit and MCD generates six false detections.

We examine in Figure 10 a corresponding alternative situation (under H_1). The LTS method detects correctly the isolated outlier ($t = 24$) and instants 35, 36 and 38 associated with the innovative outlier plus three false detections. The MCD method detects correctly all the outliers but generates six false detections. The boosting method fails to select the isolated outlier but selects the beginning ($t = 35$) and the end ($t = 38$) of the events generated by the innovative outlier without generating any false detection.

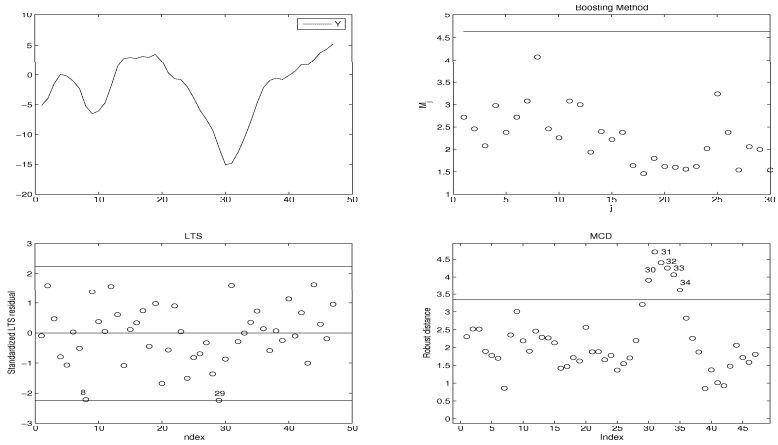

Figure 9. An $AR(3)$ time series without outliers, $n = 47$, $n_{out} = 0$. The boosting method works correctly and detects none. At the contrary, two observations are borderline for the LTS detection limit and MCD generates six false detections.

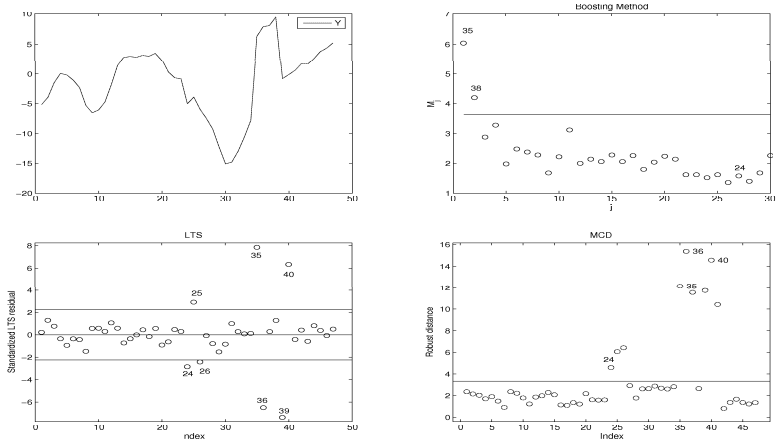

Figure 10. An $AR(3)$ time series with an additive and an innovative outliers, $n = 47$, $n_{out} = 5$ (24 and the interval 35 to 38). LTS method detects correctly the isolated outlier and instants 35, 36 and 38 associated with the innovative outlier plus three false detections. MCD method detects correctly all the outliers but generates six false detections. The boosting method fails to select the isolated outlier but selects the beginning ($t = 35$) and the end ($t = 38$) of the events generated by the innovative outlier without generating any false detection.

Table 4. Hard real data sets without outliers

Data	**Response**	**Predictors**	**Number of Obs.**
Boston Housing	Median housing price in the tract.	13 predictors. Fully described in Breiman et al. (1984).	506
Paris Pollution	Daily maximum ozone concentration.	3 predictors. Fully described in Chèze et al. (2002).	1200

4.4 Two Real Data Sets of Large Size, without Outliers

To end, we examine two hard real data sets (described in Table 4) which do not contain any outliers, in order to test the method on real data to illustrate the good behavior with respect to false detections. The first real data set, called Boston Housing, is fully described in Breiman et al. (1984, pp. 217-220) and extensively used in regression literature. The Paris Pollution data are used to deal with the analysis and prediction of ozone concentration in Paris area (see Bel et al., 1999). Highly polluted days are often hard to predict: usual estimation methods need to be suitably post-processed to improve the performance on these observations. In Chèze et al. (2003), it is shown that starting from a CART regression tree, boosting performs automatically this improvement. The conclusion is that it seems that the highly nonlinear nature of the data lead to over estimations by the two considered alternative methods.

The results for Boston Housing are given in Figure 11. All the methods generates false detections but only six for the boosting one to be compared to very large numbers for the two other ones. A similar situation occurs for the Paris Pollution data (see Figure 12). The LTS and MCD methods lead to very large numbers of false detections while the boosting one highlights only one day.

A deeper examination of this day selected by the boosting-based algorithm, shows that it corresponds to a day where the temperature is high (about 28°C), the day before is polluted (about 126 $\mu g/m^3$) and there is no wind, so the ozone concentration should be about 120 $\mu g/m^3$ but only 15 $\mu g/m^3$ is observed, which is particularly hard to predict and atypical with respect to the small set of explanatory variables considered in this model.

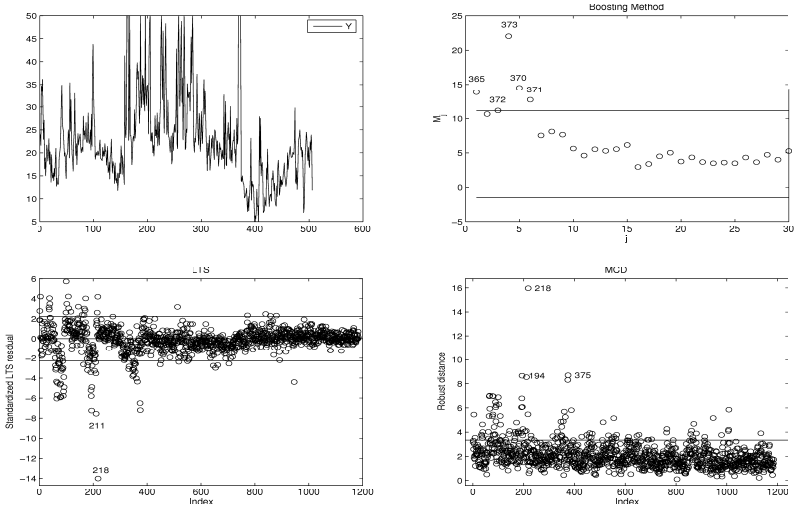

Figure 11. Boston Housing real data set. All the methods generates false detections: six for the boosting one and very large numbers for the two other ones.

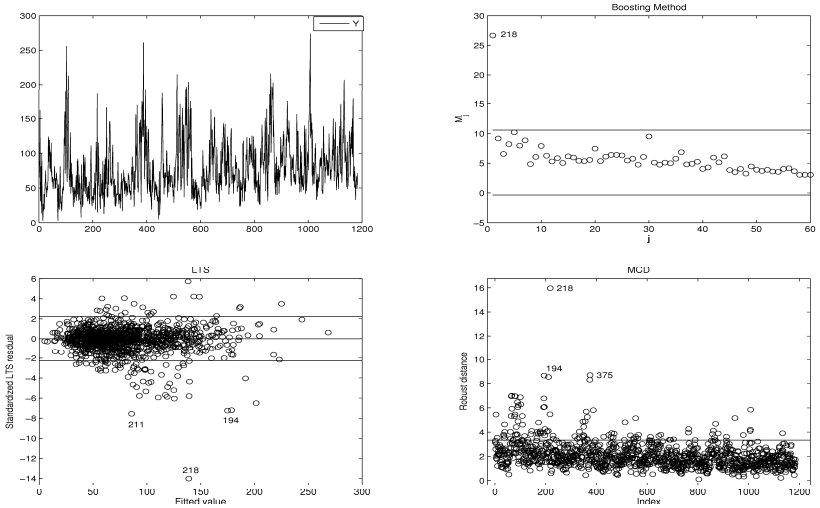

Figure 12. Pollution real data set. LTS and MCD methods lead to very large numbers of false detections while the boosting one highlights only one day which is particularly hard to predict.

5 Conclusion

We have proposed in this paper, an outlier detection algorithm for regression problems based on the use of a flexible nonparametric estimation method and the information provided by the adaptive resampling process generated by boosting. An entirely data-driven procedure can be defined without any specific assumption about the noise distribution.

Due to the difficulty to handle theoretical analysis of boosting, an extended experimental study is provided. A lot of well-known bench data sets are considered: examples of small sample size as well as larger ones, real data sets as well as simulated ones, slightly nonlinear underlying regression function as well as highly nonlinear ones. A comparative study with two well-known competitors (which are taken here to provide reference results) allows to show the value of the method despite the computational effort needed to perform outlier detection especially when both the sample size and the number of outliers are large.

Finally, let us remark that, since CART allows to construct classification trees and since AdaBoost algorithm was originally designed to cope with classification problems, it would be straightforward to extend this kind of algorithm to the classification case.

References

Bel, L.; Bellanger, L.; Bonneau, V.; Ciuperca, G.; Dacunha-Castelle, D.; Deniau, C.; Ghattas, B.; Misiti, M.; Misiti, Y.; Oppenheim, G.; Poggi, J.M.; Tomassone. R. (1999). Éléments de comparaison de prévisions statistiques des pics d'ozone. *Revue de Statistique Appliquée* **47**, 7-25 (in French).

Borra, S.; Di Ciacco. A. (2002). Improving nonparametric regression methods by bagging and boosting. *Comput. Statist. Data Anal.* **38**, 407-420.

Breiman, L. (1998). Arcing classifiers (with discussion). *Ann. Statist.* **26**, 801-849.

Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. (1984). *Classification and Regression Trees.* Chapman, New York.

Catoni, C.; Karioti, V. (2004). Detecting an innovative outlier in a set of time series. *Comput. Statist. Data Anal.* **46**, 561-570.

Chen, C. (2002). Robust regression and outlier detection with the ROBUSTREG procedure. In *Proc. 27th Annual SAS User Group Int. Conf.*, SAS Institute, Cary, NC, Paper 265-27.

Chèze, N.; Poggi, J-M.; Portier, B. (2003). Partial and recombined estimators for nonlinear additive models. *Stat. Inference Stoch. Process.* **6**, 155-197.

Drucker, H. (1997). Improving regressors using boosting techniques. In *Proc. 14th Int. Conf. Machine Learning*, D.H. Fisher, ed., Morgan Kaufmann, San Francisco, CA. pp. 107-115.

Freund, Y.; Schapire, R.E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. System Sci.* **55**, 119-139.

Friedman, J. (1991). Multivariate adaptive regression splines (with discussion). *Ann. Statist.* **19**, 1-141.

Friedman, J.; Hastie, T.; Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting. *Ann. Statist.* **28**, 337-407.

Gey, S.; Poggi, J.-M. (2006). Boosting and instability for regression trees. *Comput. Statist. Data Anal.* **50**, 533-550.

Jolliffe, I.T. (2002). *Principal Component Analysis.* Springer, New York.

Justel, A.; Pena, D.; Tsay, S.T. (2001). Detection of outlier patches in autoregressive time series. *Statist. Sinica* **11**, 651-673.

Pena, D.; Yohai, V. (1999). A fast procedure for outlier diagnostics in large regression problems. *J. Amer. Statist. Assoc.* **94**, 434-445.

Rousseeuw, P.J. (1984). Least median of squares regression. *J. Amer. Statist. Assoc.* **79**, 871-880.

Rousseeuw, P.J.; Van Driessen, K. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics* **41**, 212-223.

Rousseeuw, P.J.; Leroy, A. (1987). *Robust Regression and Outlier Detection.* Wiley, New York.

Schapire, R.E.; Freund, Y.; Bartlett, P.; Sun Lee, W. (1998). Boosting the margin: a new explanation for the effectiveness of voting methods. *Ann. Statist.* **26**, 1651-1686.

Verboven, S.; Hubert, M. (2005). LIBRA: a MATLAB library for robust analysis. *Chemometrics and Intelligent Laboratory Systems* **75**, 127-136.

Appendix

Table 5. Synthetic results of our method for the small size data sets from Rousseeuw and Leroy (1987). The table gives the book page number, the number of observations, the number of explanatory variables and, for the two methods LTS and MCD, the supposed numbers of outliers (the one given using LTS is taken as reference value in the sequel) and the corresponding percentages. Finally, the last column classifies the data sets in three categories: 1 stands for successful results, 2 for satisfactory detection but poor selection and 3 for unsuccessful results.

Page number in Rousseeuw and Leroy (1987)	n	p	n_{out}^{LTS}	%	n_{out}^{MCD}	%	**Category**
22	20	1	0	0	0	0	1
26	24	1	6	25%	7	26%	2
27	47	1	4	8%	7	15%	3
47	21	1	2	9%	3	14%	2
57	28	1	4	14%	6	21%	2
62	20	1	0	0	2	10%	1
73	18	1	2	11%	4	22%	2
76	21	3	0	0	4	19%	2
79	21	5	2	9%	4	18%	3
82	28	3	1	3%	4	12%	3
86	31	3	1	3%	5	12%	1
94	75	3	10	13%	14	18%	1
96	19	1	0	0	4	21%	2
103	12	2	0	0	2	16%	2
110	50	3	1	2%	5	10%	1
154	23	4	1	4%	5	20%	2
155	25	2	2	8%	5	20%	2
156	18	2	0	0	4	22%	2

Nathalie Chèze
Laboratoire de Mathématique – U.M.R.
C 8628,
"Probabilités, Statistique et Modélisation",
Université Paris-Sud,
Bât. 425, 91405 Orsay cedex,
France

Jean-Michel Poggi
Laboratoire de Mathématique – U.M.R.
C 8628,
"Probabilités, Statistique et Modélisation",
Université Paris-Sud,
Bât. 425, 91405 Orsay cedex,
France
e-mail: *Jean-Michel.Poggi@math.u-psud.fr*