

Robust regression with a categorical covariable

Mia Hubert and Peter J. Rousseeuw

*Department of Mathematics and Computer Science, U.I.A.,
Universiteitsplein 1, B-2610 Antwerp, Belgium*

Abstract: A fast algorithm is presented for robust estimation of a linear model with a distributed intercept. This is a regression model in which the data set contains groups with the same slopes but different intercepts, a situation which often occurs in economics. In each group, the algorithm first looks for outliers in (x, y) -space by means of a robust projection method. Then a modified version of the resampling technique is applied to the whole data set, in order to find an approximation to least median of squares or other regression methods with a positive breakdown point. Because of the preliminary projections, the number of subsets may be drastically reduced. Simulations and examples show that the overall computation time is substantially lower than that of the straightforward algorithm. The method is illustrated with a real data set.

Keywords: Algorithms; Computation time; Distributed intercept; Outlier detection; Positive-breakdown methods.

1 Introduction

We consider the regression model

$$y = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + f(x_{p+1}) + \varepsilon, \quad (1.1)$$

where the regressors x_1, \dots, x_p take on real values and ε is normally distributed with mean 0 and variance σ^2 . In the usual linear model one puts $x_{p+1} = 1$, hence $f(x_{p+1}) = \beta_{p+1}$ is

the intercept term. In this paper, we will consider the more general situation where x_{p+1} is a categorical (or nominal) variable which can take on q different values. Therefore $f(x_{p+1})$ can also take on q values, which we will denote by $\delta_1, \dots, \delta_q$.

Equivalently, we can say that each of the n observations belongs to one of q groups, according to its value of x_{p+1} . This relation is easily expressed by the introduction of dummy variables. We can therefore rewrite (1.1) as the following model:

$$y_i = \sum_{j=1}^p \beta_j x_{ij} + \sum_{k=1}^q \delta_k I_{ik} + \varepsilon_i, \quad i = 1, \dots, n, \quad (1.2)$$

where

$$\begin{aligned} I_{ik} &= 1 && \text{if case } i \text{ belongs to group } k \\ &= 0 && \text{otherwise.} \end{aligned}$$

Thus all groups obey a linear structure, with the same slope parameters β_1, \dots, β_p . On the other hand, the intercept term δ_k (for $k = 1, \dots, q$) differs between the groups. Therefore we call (1.2) a linear model with a distributed intercept. In simple regression it describes q parallel regression lines. In multiple regression, we will fit q parallel hyperplanes. Note that the usual linear model with a single intercept is still a particular case, with $q = 1$. One could also consider (1.2) as a linear model with $p + q$ explanatory variables, but then the continuous regressors and the dummies would be treated in the same way, which would increase the computational effort for the robust algorithms considered in this paper.

The least squares method (LS) fits the model (1.2) in a nonrobust way. For instance, it is possible to apply the standard calculations by processing the dummy variables in the same manner as the continuous ones, as described by Draper and Smith (1981), Montgomery and Peck (1982), and Chatterjee and Price (1977). We can also carry out an analysis of covariance, which directly assumes x_{p+1} to be a categorical variable. We refer to Montgomery (1991) and Edwards (1985) for more details.

Unfortunately, the least squares method is very sensitive to outliers. Even a small fraction of contamination can influence the regression estimates. In our model, both outliers in the y -direction and in the x -direction can occur. Our aim will be to estimate the $p + q$ coefficients β_1, \dots, β_p and $\delta_1, \dots, \delta_q$ in a robust way. Armstrong and Frome (1977) developed a linear programming algorithm to obtain the least absolute value (L_1) estimate. This approach is already more robust against outliers in the y -direction, but not against outliers

in the x -direction which can still tilt the estimated hyperplanes. Rousseeuw and Wagner (1994) proposed an algorithm to approximate the least median of squares (LMS) solution. This algorithm can deal with vertical outliers as well as bad leverage points, but it consumes a large amount of computation time. Here, we will focus on obtaining a faster method.

In the next section we will describe the proposed algorithm. Section 3 studies its behaviour on generated data, whereas Section 4 analyzes a real data set.

2 Description of the algorithm

The algorithm essentially consists of two parts. For each group, we first detect outliers in (\mathbf{x}, y) -space by means of a robust projection method. Then we apply an adapted version of the resampling algorithm. The latter was originally devised for the single-intercept linear model ($q = 1$), where it is used for computing the least median of squares estimator (Rousseeuw 1984) given by

$$\underset{T}{\text{minimize}} \quad \text{median}_{i=1, \dots, n} \quad r_i^2(T), \quad (2.1)$$

where r_i stands for the regression residual.

Let us now consider both parts of the algorithm in turn.

2.1 Projection method

We will denote the different groups by G_k (for $k = 1, \dots, q$) with $n_k = |G_k|$. The index i always runs through the whole data set, whereas j indexes the objects of one group. An observation $z_j = (x_{1j}, x_{2j}, \dots, x_{p+1j}, y_j)$ is written as (\mathbf{x}_j, y_j) . Suppose now we are investigating G_k .

Analogous to Donoho and Gasko (1992), we define the *outlyingness* of an observation $z_j = (\mathbf{x}_j, y_j)$ as:

$$u_j = \sup_{\mathbf{v} \neq \mathbf{0}} \frac{|z_j \mathbf{v}^t - L(z_1 \mathbf{v}^t, \dots, z_{n_k} \mathbf{v}^t)|}{S(z_1 \mathbf{v}^t, \dots, z_{n_k} \mathbf{v}^t)} \quad (2.2)$$

where $L(z_1 \mathbf{v}^t, \dots, z_{n_k} \mathbf{v}^t)$ and $S(z_1 \mathbf{v}^t, \dots, z_{n_k} \mathbf{v}^t)$ are the univariate least median of squares (LMS) location and scale estimators applied to the projections of the data on the direction \mathbf{v} . These robust estimators correspond with the midpoint and the length of the shortest half (see Rousseeuw and Leroy 1987). Note that (2.2) measures the outlyingness of the points z_j relative to the group G_k . From a computational point of view, is it necessary to restrict

ourselves to a finite set of directions \mathbf{v} . Rousseeuw and van Zomeren (1990) propose to take all

$$\mathbf{v}_l = \mathbf{x}_l - M \quad \text{for } l = 1, \dots, n_k,$$

where M is the coordinatewise median of the data (here, of the group G_k). The corresponding u_j are then easy to compute. Note that these u_j are no longer affine invariant (because we do not consider all vectors \mathbf{v}), but they are still invariant for permutations of the data, as well as translations and scalar factors.

We calculate the u_j in the $(p + 1)$ -dimensional space (\mathbf{x}, y) and not only in the space of the explanatory variables. The latter is done by Rousseeuw and van Zomeren (1991) before they apply L_1 regression, which works well because the L_1 method is rather resistant against vertical outliers. But our purpose here is to indicate all possible outliers in order to speed up the second stage of our algorithm, which is based on subsets consisting of clean data.

Having the collection $\{u_j, j = 1, \dots, n_k\}$, we will select a set P_k of points with relatively small u_j . There exists a tradeoff between the dangers of selecting too many points (some of which may be outlying) or too few points. Here, our strategy will be to select approximately $\frac{2n_k}{3}$ points, as described in the next subsection.

2.2 Resampling method

In each group G_k we will select points with small u_j . The complete algorithm is organized as follows.

For $k = 1, \dots, q$ do:

Step 1. Consider the k -th group G_k and check whether it contains at least $\lceil \frac{3}{2}(p + 1) \rceil$ data points. (Otherwise skip this value of k).

Step 2. Carry out the projection method on G_k as described in Section 2.1. Then compute the order statistic $u_{(h)}$ of $\{u_1, \dots, u_{n_k}\}$ where $h = \lceil \frac{2n_k}{3} \rceil$. Select the set P_k consisting of all points (\mathbf{x}_j, y_j) in G_k for which

$$u_j \leq u_{(h)}.$$

Step 3. Repeat $nrep(k)$ times:

- Draw a subset of $p + 1$ points from P_k . (This is possible by Step 1.)
- Compute the hyperplane formed by this $(p + 1)$ -subset. A linear system has to be solved, which results in the slopes $\hat{\beta}_1, \dots, \hat{\beta}_p$ and an intercept term.
- For each group a separate intercept is obtained. First calculate the values $d_i = y_i - \hat{\beta}_1 x_{i1} - \dots - \hat{\beta}_p x_{ip}$ for $i = 1, \dots, n$. For each group G_l (where $l = 1, \dots, q$) then apply a robust location estimator to those d_j belonging to G_l , yielding an intercept denoted by $\hat{\delta}_l$.
- For each point i , compute the residual $r_i = y_i - \hat{\beta}_1 x_{i1} - \dots - \hat{\beta}_p x_{ip} - \hat{\delta}_l$ where $\hat{\delta}_l$ is the intercept of its group. From the entire collection of residuals $\{r_i, i = 1, \dots, n\}$ we then obtain the objective function

$$\underset{i=1, \dots, n}{\text{median}} r_i^2 \tag{2.3}$$

Enddo $k = 1, \dots, q$.

Step 4. Report the estimate $(\hat{\beta}_1, \dots, \hat{\beta}_p, \hat{\delta}_1, \dots, \hat{\delta}_q)$ with the smallest objective function (2.3).

This resampling algorithm is related to that in Rousseeuw and Wagner (1994), but now the $(p + 1)$ -subsets are drawn from P_k and not from the whole group G_k . The advantage is that $nrep(k)$, the number of replications, may be taken quite small. On the other hand, the initial projections do not need much computation time.

Once the above algorithm has been carried out, it remains possible to assign weights to the observations based on their LMS residuals, and then to perform a reweighted least squares analysis.

In the next section we will give more information about $nrep(k)$, illustrate the method on a generated data set, and compare it with the earlier algorithm.

Table 1: Number m of subsamples used in PROGROUPE for method 1 and 2

p	Method 1		Method 2
	m	for	m
1	1000	$n > 50$	250
2	1500	$n > 22$	375
3	2000	$n > 17$	500
4	2500	$n > 15$	625
5	3000	$n > 14$	750
6	3500	all n	875
7	4000	all n	1000
8	4500	all n	1125
9	5000	all n	1250
10	5500	all n	1375
≥ 11	6000	all n	1500

3 Simulation results

We have written a Fortran program named PROGROUPE, an abbreviation which stands for Program for RObust reGression on gROups Using Projections. It is a revised and extended version of the program PROGRESS (Rousseeuw and Leroy 1987) for robust multiple regression. PROGROUPE fits a distributed intercept model (1.2) by means of the algorithm of Rousseeuw and Wagner (method 1), and by means of our new algorithm with the preliminary projections (method 2). (For completeness, the program also provides a third method in which the slopes as well as the intercepts may vary between the groups. This method corresponds to carrying out a robust regression in each group, but its computation is much faster than with PROGRESS.)

The number of subsamples used by method 1 is the same as in PROGRESS. It was chosen such that the probability of drawing at least one good subsample is near 1. (A "good" subsample consists only of uncontaminated points.) The first column of Table 1 shows this number m for the different values of p . For smaller values of n we draw *all* subsets, hence m equals C_n^{p+1} .

The next column of Table 1 lists the number used in the second method. Our simulations

have indicated that it is sufficient to consider about one fourth of the original number, since then the objective function value is close to that of method 1.

For each group G_k we compute $nrep(k)$ accordingly, by

$$nrep(k) = \left\lceil m \frac{n_k}{n} \right\rceil.$$

For $p \leq 5$ and small n_k we also compute the number of all possible $(p + 1)$ -subsets of P_k . If this is less than $nrep(k)$, we of course consider all these combinations. (Note that the original PROGRESS contains a "quick" option using a different number of subsamples. The corresponding "quick" option in PROGROUP again takes one fourth of those numbers.)

To illustrate the effectiveness of the new algorithm we will first apply it to a generated data set. We constructed 4 groups of 100 points each, following the model

$$y_i = x_{i1} + x_{i2} + x_{i3} + \sum_{k=1}^4 \delta_k I_{ik} + \varepsilon_i, \quad i = 1, \dots, 400 \quad (3.1)$$

where

$$\begin{aligned} x_{ij} &\sim N(0, 10) & i = 1, \dots, 400; j = 1, 2, 3 \\ \varepsilon_i &\sim N(0, 1) & i = 1, \dots, 400 \\ \delta_k &= 10, 20, 30 \text{ or } 40. \end{aligned}$$

This means that we are dealing with 3 continuous factors and 4 groups, with intercept parameters 10, 20, 30 and 40. The slope parameters of the hyperplanes are equal to 1. We then included contaminated points by replacing 20% of the x_{i1} by $\tilde{x}_{i1} \sim N(100, 10)$, yielding bad leverage points.

Table 2 shows the parameter estimates given by both methods. The number of subsamples used, the objective function and the computation time on a Sun workstation and on a 486-PC are also given. The latter are measured in seconds and represent the processing time needed for calculating the coefficients (without drawing the residual plots).

Both methods provide accurate estimates. A complete output shows that they find the same outliers and have similar residual plots. As expected, we see that the new algorithm is executed about four times faster. The actual ratio was 3.5, due to some input/output operations which both algorithms have in common. Because the objective function values are comparable too, we thus may conclude that the new algorithm performs as well as the slower one.

Table 2: Results of PROGROUPE on a generated data set

	Method 1 (without proj.)	Method 2 (with proj.)
$\hat{\beta}_1$	1.015	0.986
$\hat{\beta}_2$	0.995	0.976
$\hat{\beta}_3$	0.983	0.995
$\hat{\delta}_1$	9.919	9.893
$\hat{\delta}_2$	20.342	20.076
$\hat{\delta}_3$	30.128	30.039
$\hat{\delta}_4$	40.111	40.292
# subsamples	2000	500
obj. function	1.309	1.307
time (Sun)	19.9 sec.	5.7 sec.
time (PC)	147 sec.	43 sec.

Similar results were obtained in other simulation experiments where 20% of vertical outliers were generated (as in Rousseeuw and Leroy 1987, page 209), and for other choices of p and q .

4 Case study

In econometrics, Mincer (1974) has introduced the so-called *earnings function* which corresponds to the model

$$\ln(y) = \beta_1 s + \beta_2 e + \beta_3 e^2 + \delta + \varepsilon, \quad (4.1)$$

where y = personal income, s = years of schooling, and e = years of experience. As it stands, (4.1) is a linear model with a single intercept, but it is now generally assumed that the intercept may vary substantially across sectors (that is, in different industries). For a recent study, see Wagner (1990).

Our data set consists of 296 observations (=persons) belonging to 7 different sectors in Switzerland. (This data set originated from a social survey in 1987 and was analyzed by Rousseeuw and Wagner (1994) with the earlier algorithm, which makes it a good basis of

Table 3: List of sectors

k	Sector
1	high-tech manufacturing
2	other manufacturing industries
3	building and construction
4	wholesale and retail trade
5	transportation and communication
6	banking and insurance
7	other services

comparison.) We want to estimate the parameters $\beta_1, \dots, \beta_3, \delta_1, \dots, \delta_7$ of the model

$$\ln(y_i) = \beta_1 s_i + \beta_2 e_i + \beta_3 e_i^2 + \sum_{k=1}^7 \delta_k I_{ik} + \varepsilon_i \quad (i = 1, \dots, 296) \quad (4.2)$$

with $I_{ik} = 1$ if person i works in sector k . Table 3 specifies the sectors.

We first looked at the distributions of the individual variables, and searched for multicollinearity between the predictor variables. Scatterplots and correlation coefficients did not turn up a linear relationship between any two of them. (Of course, we know that x_2 and x_3 are quadratically related.) We also considered a rotating point cloud graphic of the three regressors, which only gave an indication about the connection between x_2 and x_3 , so multicollinearity does not seem to be a problem here.

Table 4 lists the results of both algorithms on this data set. We see that the objective function values are again close to each other, and that the execution time is reduced about 3.5 times. The coefficients differ slightly, but they all have the same sign and magnitude. Let us also look at the relative ranks of the intercepts, shown in separate columns. Only the groups with ranks 1 and 3 are interchanged, but the other ranks remain. We may conclude that sectors 1, 2, 5 and 6 pay the best salaries, while sectors 3, 4 and 7 score less.

A gaussian $Q - Q$ plot (Figure 1) confirms that the residuals are roughly normally distributed. Only a small number of outliers causes the distribution to be heavier-tailed. Therefore, the normality assumption holds for a majority of the residuals.

Table 4: Regression results on wage differences in Switzerland

	Method 1		Method 2	
	estimate	rank	estimate	rank
$\hat{\beta}_1$	0.08561		0.05484	
$\hat{\beta}_2$	0.04778		0.05242	
$\hat{\beta}_3$	-0.00070		-0.00081	
$\hat{\delta}_1$	6.72751	4	6.99306	4
$\hat{\delta}_2$	6.80748	6	7.02210	6
$\hat{\delta}_3$	6.56816	2	6.91435	2
$\hat{\delta}_4$	6.62184	3	6.87065	1
$\hat{\delta}_5$	6.77848	5	7.00435	5
$\hat{\delta}_6$	7.05075	7	7.32903	7
$\hat{\delta}_7$	6.55178	1	6.92371	3
# subsamples	2000		497	
obj. function	0.288		0.293	
time (Sun)	15.2 sec		4.4 sec	
time (PC)	111 sec		30 sec	

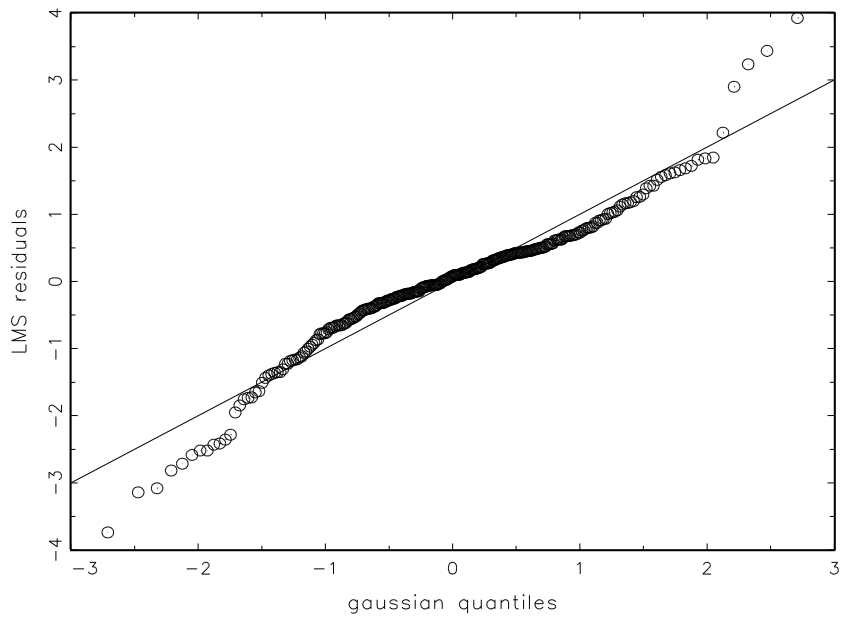


Figure 1: Gaussian $Q - Q$ plot of the residuals

Note

The Fortran source code of PROGROUP is available from the authors, at the email addresses hubert@wins.uia.ac.be and rousse@wins.uia.ac.be.

References

- Armstrong, R.D., and Frome, E.L. (1977), A special purpose linear programming algorithm for obtaining least absolute value estimators in a linear model with dummy variables, *Communications in Statistics: Simulation and Computation*, B6, 383-398.
- Chatterjee, S., and Price, B. (1977), *Regression Analysis by Example*, New York: John Wiley.
- Donoho, D.L., and Gasko, M. (1992), Breakdown properties of location estimates based on halfspace depth and projected outlyingness, *The Annals of Statistics*, 20, 1803-1827.
- Draper, N.R., and Smith, H. (1981), *Applied Regression Analysis*, New York: John Wiley.
- Edwards, A.L. (1985), *Multiple Regression and the Analysis of Variance and Covariance*, New York: W.H. Freeman and Company.
- Mincer, J. (1974), *Schooling, Experience, and Earnings*, New York: Columbia University Press.
- Montgomery, D.C. (1991), *Design and Analysis of Experiments*, New York: John Wiley.
- Montgomery, D.C., and Peck, A.E. (1982), *Introduction to Linear Regression Analysis*, New York: John Wiley.
- Rousseeuw, P.J. (1984), Least median of squares regression, *Journal of the American Statistical Association*, 79, 871-880.
- Rousseeuw, P.J., and Leroy, A.M. (1987), *Robust Regression and Outlier Detection*, New York: John Wiley.
- Rousseeuw, P.J., and Wagner, J. (1994), Robust regression with a distributed intercept using least median of squares, *Computational Statistics & Data Analysis*, 17, 65-76.

- Rousseeuw, P.J., and van Zomeren, B.C. (1990), Unmasking multivariate outliers and leverage points, *Journal of the American Statistical Association*, 85, 633-639.
- Rousseeuw, P.J., and van Zomeren, B.C. (1992), A comparison of some quick algorithms for robust regression, *Computational Statistics & Data Analysis*, 14, 107-116.
- Wagner, J. (1990), Sektorlohndifferentiale in der Bundesrepublik Deutschland: Empirische Befunde und ökonometrische Untersuchungen zu theoretischen Erklärungen, *Discussion Paper No. 154, Fachbereich Wirtschaftswissenschaften, Universität Hannover*.