Stochastic Optimization Algorithm Applied to Least Median of Squares Regression

Dimitar L. Vandev

Institute of Math., Sofia

Abstract

The paper presents a stochastic optimization algorithm for computing of least median of squares regression (LMS) introduced by (Rousseeuw and Leroy 1986). As the exact solution is hard to obtain a random approximation is proposed, which is much cheaper in time and easy to program. A MATLAB program is included.

Keywords: robust high–breakdown estimators, least median of squares, stochastic approximation algorithm, Monte – Carlo study.

1 Introduction

Many authors considered robust estimators of the covariance matrix and the location in the multidimensional case. When a high level of contamination is expected it is appropriate to use estimators with high breakdown point. Such estimators are the minimum volume ellipsoid (MVE) and the minimum covariance determinant (MCD), introduced by Rousseeuw and Leroy (1986). On the other hand in the robust regression literature very popular is the Least Median of the squares (LME) estimator which also has high breakdown point. Recently Neykov and Neytchev (1990) proposed a robust alternative of the maximum likelihood estimators. Namely let $f(\theta, x)$ be the likelihood functions of the individual observation x. We denote by X the finite set of all observations. Here θ is the vector of unknown parameters. Let $A(\theta) = \{-log(f(\theta, x)), x \in X\}$ be the (increasingly) ordered set of the values of f at a fixed point θ. Denote by $M(k, \theta)$ the k-smallest and by $S(k, \theta)$ the sum of the k smallest numbers of the set $A(\theta)$. The minimizers of these two random functions are to be considered as estimators in statistical sense.

Vandev[1] (1992) has shown that MVE and MCD estimators may be extracted from this robustified version in the gaussean case. The same is true for LME in regression. It was also shown that in general all robustified maximum likelihood estimators have high break down point.

Computationally both (trimmed and least median) problems are not easy to solve in a conventional way because the functions involved have many local minima. Thus the minimization turns out to be a serious combinatorial problem. Up to now mainly the resampling

[1]Research partially supported by Ministry of Education and Science under contracts I19/91 and M60/91.

technique is used for the purpose, see Rousseeuw & Leroy (1986). Todorov (1992) successfully used the simulated annealing algorithm for MCD estimator.

In this paper an algorithm is presented for approximate calculating of LME(k). Hawkins (1993) used a feasible set algorithm for exact calculation of the minima. Our proposition is based on the well known Robins-Monro (1951) procedure for stochastic optimization, which was already successfully used by Martin and Masreliez (1975) in the robust estimation. We will call the algorithm RM algorithm. An early version of this paper was presented as a short talk at the seminar of Statistical Data Analysis held in Varna (Vandev, 1992a).

2 Definitions and Notations

Let X be set of observations of size n. Let E be the p-dimensional Euclidean space and f - function defined on $E \times X$. Let $A(\theta) = \{f(\theta, x), x \in X\}$ be the (increasingly) ordered set of the values of f at a fixed point θ. Denote by $M(k, \theta)$ the k-smallest number in the set $A(\theta)$. Denote by LME(k) the set of θ which minimizes $M(k, \theta)$, i.e.:

$$LME(k) = \arg\min_{\theta} f(\theta, x_{(k)}) = \arg\min_{\theta} F(\theta),$$

where $f(\theta, x_{(1)}) \leq f(\theta, x_{(2)}) \leq ... \leq ... \leq f(\theta, x_{(n)})$. As usual here the subindex denote the element of the corresponding permutation which depends on the value of θ.

One easily comes to the idea to use some of the gradient methods for the case. It is clear that the gradient of the function which is to be minimized equals "almost everywhere" the gradient of $f(\theta, x_{(k)})$ for some k. The usual minimizing of the function $F(\theta)$ simply follows the procedure of iterative updating of the unknown parameter at the step i:

$$\theta_{i+1} = \theta_i - \alpha_i * grad(F(\theta_i)).$$

It could work in our case. However there are two difficulties:

- We do not know which of all functions is at k-th place in the ordered set $A(\theta)$. So all these functions are to be computed and their values sorted.
- The global function turns out to be not convex, i.e. it has multiple local minima. So this simple procedure does not assure convergence to the global minima.

To overcome those difficulties we propose a random search of the minima. Our idea is to calculate the gradient approximately in the sense that we will fix the function among small number of randomly chosen functions.

We will illustrate the algorithm in the case of multiple linear regression. It is clear that the same algorithm could be used as well in the case of joint estimation of the parameters location and scatter.

Consider x a vector in a m-dimensional Euclidean space. In the case of multiple regression the function f is of the form:

$$f(\theta, x) = abs(x^0 - x'\theta - \theta^0).$$

Here $p = m$ and the gradient of f may be calculated as follows:

$$\frac{df(\theta, x)}{d\theta} = -sign(x^0 - x'\theta - \theta^0).(x', 1).$$

3 The RM Algorithm

The famous Robins-Monro (1951) procedure when applied to the problem of minimizing the function $F(\theta)$ consists in the following. Let start with some $\theta = \theta_0$. Let now calculate the gradient $grad(F(\theta))$ at this point. It may be randomly disturbed by some random variable with zero expectation. At the step i the parameter will be changed according the following formula:

$$\theta_{i+1} = \theta_i - \gamma_i * \frac{grad(F(\theta_i))}{||grad(F(\theta_i))||}.$$

The sequence $\{\gamma_i, i = 1, 2, ...\}$ is chosen to satisfy the relations: $\sum_{i=1}^{\infty} \gamma_i^2 < \infty$, $\sum_{i=1}^{\infty} \gamma_i = \infty$.

Denote the iteration number by i and the estimated parameter - by B. The parameter δ is chosen to be about p times the maximal expected absolute value of the elements of B. The iterations are performed fixed number of times usually up to 300.

- Step 0. Set maxn, set i=1, set δ.
- Step 1. Chose at random 10 indexes among the numbers from 1 to n. Calculate these 10 functions. Sort their values.
- Step 2. Chose the value corresponding to the desired proportion ($j/10 = k/n$) and the function which produces that value (say f).
- Step 3. Calculate the normalized gradient $D(f)$ of f.
- Step 4. Set $B := B - D(f).\delta/i$. Set $i = i + 1$. If $i \leq$ maxn then Goto step 1.

Comments:

1. The number of calculations needed does not depend on the number of functions, i.e. the total number of the observations does not affect the efficiency of the algorithm.
2. The number of randomly chosed functions is chosen for convenience (here it is equal to 10). In fact one need to investigate the dependence of the efficiency of the algorithm of this number. We expect a changing (slightly increasing with the number of iterations) value to be the optimal one.

4 The Program

Here follows the MATLAB program used to test the above algorithm in the case with LME in the multiple linear regression:

```
function [b] = lmereg(x, y, prop, delta, iter)
  rand('uniform');                    % init random numbers
  [ n, m ] = size(x);                 % get size of x
```

```
  b = zeros(m,1);                                  % starting values of b
  if (nargin < 3), prop=6 ; end                    % default prop
  if (nargin < 4), delta = 10.; end                % default delta
  if (nargin < 5), iter = 100; end                 % default iter
  for k = 1 : iter
    J = round(ones(10,1)/2+rand(10,1)*n);          % 10 random integers
    res = y(J)-x(J, : )* b;                        % calculate residuals
    ares = abs(res);                               % absolute values
    [ xw, list ] = sort(ares);                     % sort
    ti = x(J(j), : )'* sign(res(j));               % gradient
    theta = sqrt(ti'* ti);                         % norm of gradient
    gamma = delta / k;                             % new gamma
    b = b + gamma * ti./theta;                     % update b
  end
end
```

The MATLAB version is presented here as a shortest. A FORTRAN program is available from the author upon a request.

5 Simulated Examples

Two examples are presented of simulated simple and multiple linear regression to show the advantages and quality of the algorithm.

5.1 Simple Linear Regression

The first model was chosen to illustrate the robust properties of the used version of maximum likelihood. The response Y is generated by the following model:

$$y = 5 - 2 * x + e.$$

Here e is a standard normal random variable. The sample consists of 1000 observations. It was corrupted by destroying 30% of the observations. A reasonable estimations is achieved after 150 iterations despite of the large number of outliers. The algorithm was used with fixed number of iterations equal to 150 and $\delta = 10$.

On the fig. 1 bellow one random solution is presented for the estimator 6/10. For a comparison the unique least squares solution is also plotted and quite obvious is far from the

model. Thus the robust properties of the estimator are obvious.

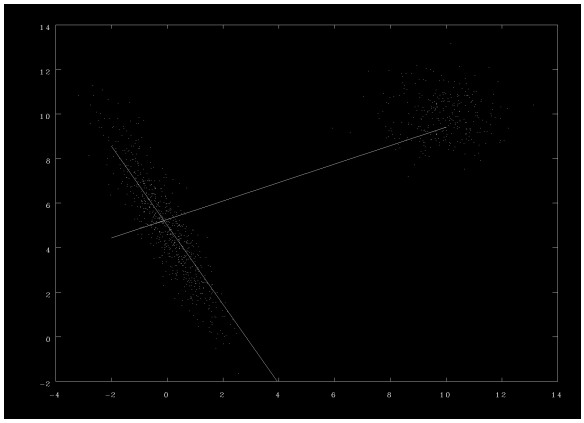

fig.1

The further Monte - Carlo study of the same data set shows that all the local minima are not far from the exact solution which is not shown on the picture. The statistical error of the estimator consists of two almost independent parts: one determined by the sample, and another of the randomness of the algorithm. Our feeling is that the first one is larger, but this conjecture is to be investigated further.

5.2 Multiple Linear Regression

The second example serves another aim. We try to study merely the computational properties of the proposed algorithm. The response Y is generated by the following model:

$$y = 2 - 2 * x_1 + 5 * x_2 - 5 * x_3 + x_4 + e.$$

The same estimator and the same number of observations was used as in the previous example. In this case we use different percent of contamination and each time generate totally new data set. A single precision FORTRAN version of the above program was used for that purpose. The computations have taken about half hour on 12 mhz AT-286 with floating point coprocessor. So the average time to solve one regression was about 1.8 seconds including the simulation of the 5000 random numbers.

The results are presented in the following table. The number of contaminated observations is shown in the first column. The form of used estimator is in the second column. Each cell in the table contains the average (with the sample standard error below) for 100 simulated with the same model data sets. In the next 4 columns are the results for the parameters of the model. The last column contains the estimated value of the functional minimized.

Cont.	Est.	a_0	a_1	a_2	a_3	a_4
100	9/10	1.9235	-1.9421	4.9492	-4.8944	3.3777
		.1149	.1569	.1249	.1265	2.4862
	8/10	1.9644	-1.9821	4.9029	-4.9164	1.2839
		.0990	.0987	.1401	.1136	.1512
	7/10	1.9390	-2.0412	4.9467	-4.8380	1.1343
		.1596	.1834	.1959	.1705	.2168
	6/10	1.9823	-1.9756	4.7355	-4.7443	.9773
		.2313	.2328	.3107	.2601	.2017
200	8/10	1.9664	-2.0136	4.7889	-4.7541	5.9930
		.1534	.1828	.1808	.2410	3.7446
	7/10	1.9103	-1.9337	4.9010	-4.8629	1.3670
		.2338	.2833	.2781	.2783	.5853
	6/10	1.9484	-1.9812	4.8867	-4.9113	1.0957
		.1811	.2229	.2409	.1701	.2407
300	7/10	1.7643	-1.7374	4.4975	-4.5186	7.7961
		.4012	.3970	.6973	.7480	4.5630
	6/10	1.8873	-1.8956	4.8093	-4.7899	1.5153
		.3159	.2421	.5369	.4467	.8834
400	6/10	1.5886	-1.6556	4.2696	-4.1614	9.5648
		.5058	.4803	.7968	.9176	4.7157

Table 1

It is seen in this table that the solutions with smallest variance are achieved when the ratio $(k+1)/10$ is close to the number of the outliers. When it is higher, some outliers effect the final solution. When it is lower the effectivity of the estimator falls down. Especially sensitive to the choice of this ratio is the value of the functional been minimized (it is not shown in the table) and it might be used to choose among the estinators. One needs additional investigation of this phenomenon also.

Strictly speaking the proposed algorithm does not reach the global minima of the functional. The estimated values of the parameters correspond to one of the local minima when infinite number of iterations is performed.

To solve this problem we tried to use as a second step the Least Maximum of Absolute Values regression applied to the k observations with smallest "residuals". In our experiments the IMSL program RLLMV was used several times until the set of k observations with smallest residuals remains unchanged. However this second stage approximation turned out to be more time consuming. More over here the number of calculations depends linearly of the amount n of the observations. It does achieve better values of the functional of course, but not so good that one should feel obliged to use it. Again the global minima is not reached.

Probably better solutions may be achieved when changing the form of the functional (1). One may use sum instead maximum in order to improve the quality of the estimation. These are so called "trimmed mean and minimum determinant of the covariance matrix (MCD)" estimators considered by Rousseeuw and Leroy (1986). The modifications of the algorithm (and the program) are obvious — one needs to calculate the gradient of the sum of prop lowest functions instead of one.

6 Conclusion

The proposed algorithm is very effective what makes it possible to be used for robust estimation of covariance and mean in the multidimensional case especially as a first "rough" step. At this step the most outliers are detected and subsequent estimations with Reweighted Least Squares or Least Absolute Values algorithms may use only the "best" observations. This second step (when performed only once) preserves the breakdown point of the original estimator (Rousseeuw and Zomeren, 1989).

REFERENCES

1. Vandev, D.L.(1992). A Note on the Breakdown Point of the Least Median and Least Trimmed Estimators, *Statistics and Probability Letters,* 16, pp. 117 – 119.

2. Hampel, F.R.(1971). A general qualitative definition of robustness. *Annals of Math. Stat.*, 42, 1887-1896.

3. Rousseeuw, P.J. & Leroy, A.M. (1986). *Robust Regression and Outlier Detection.* New York:Wiley.

4. Robins, H. & Monro, S. (1951). A stochastic approximation method. *Annals of Math. Stat.*, 22, 400-407.

5. Martin, R.D. & Masreliez, C.J. (1975) Robust estimation via stochastic approximation. *IEEE Trans. Inform. Theory* 21, 263-271.

6. & Neytchev, P.N. (1990) A Robust Alternative of the ML estimators. *COMPSTAT'90, Short communications,* Dubrovnik, Yugoslavia, pp. 99 – 100.

7. Hawkins, D.M. (1993) The feasible set algorithm for least median of squares regression, *Computational Statistics & Data Analysis*, pp. 1681 – 101.

8. Vandev, D.L. & Neykov, N.M. (1992). Robust Maximum Likelihood in the Gaussian Case. In: *New Directions in Data Analysis and Robustness,* Monte Verita, Birkhauser Verlag, Basel, pp.259 – 264.

9. Vandev, D.L. (1992a). Stochastic Optimization Algorithm Applied to Least Median Type Estimators. In: *Statistical Data Analysis, Proceedings of SDA92,*, Varna, 1992, pp.114 – 119.