DETECTION OF MULTIPLE OUTLIERS BY RANDOM ROBUST TESTING

Yong Hu 1, Yang Gao 2, Benoit St-Onge 1, Ruisheng Wang 3
1 Department of Geography, University of Quebec at Montreal
2 Department of Geomatics Engineering, The University of Calgary
3 Department of Geomatics Engineering, York University
E-mails: gao@geomatics.ucalgary.ca; {yhu; St-Onge.Benoit}@courrier.uqam.ca; rwang@yorku.ca

ABSTRACT

Numerous studies clearly show that least-squares estimators are vulnerable to the violation of these assumptions and they tend to smooth blunders or outliers into all observations. The robust estimators have been proved to be superior over the conventional least squares (LSs) estimator in processing observations contaminated by multiple blunders in both theoretical and experimental aspects. In this paper, we integrate the random sample consensus paradigm and the basic idea of a robust testing procedure to implement a random robust testing algorithm that is used to detect multiple blunders in linear systems. The algorithm can improve the computational efficiency and has equivalent blunder identification ability. We have implemented a general framework that is readily applicable to all kinds of robust estimators in a simple manner. The advantages and disadvantages of our algorithm compared with the robust testing procedure are discussed by comparing their numerical results. The experimental results show that all the M-estimators can correctly identify the blunders, and the local statistic test and scaled robust residual test have the same blunder detection ability when they are applied to M-estimators.

INTRODUCTION

In statistics one uses a parametric model implying a very limited set of probability distributions though possible, such as the common model of normally distributed errors, or that of exponentially distributed observations. Classical parametric statistics derives result under the assumption that these models were strictly true. However, apart from some simple discrete models perhaps, such models are never exactly true. In practice, many assumptions commonly made in classical parametric estimation (such as normality, linearity, and independence) are at most approximations of reality, and we never know the underlying density exactly.

The conventional least-squares estimator implicitly assume that the entire set of data can be interpreted by only one parameter vector of a given model and also assumes that the residuals of observations is of independent zero mean Gaussian distribution, which yields an uniform minimum variance unbiased parameter estimation. If a priori of observation covariance is known, a minimum variance parameter estimate can be obtained by choosing appropriate weights on the residuals. While numerous studies clearly show that least squares estimators are vulnerable to the violation of these assumptions. Sometimes even when the data contains only one blunder, least squares estimate may be completely perturbed. Furthermore, with conventional least squares, the smoothing process of blunders makes the detection and isolation of blunders not good enough because the blunders are distributed into good observations. As a result, the test procedure may be falsified and the blunders may be isolated incorrectly or may not even be detected.

Robust estimation methods have been involved by a lot of applications. The discussions in the literature, however, have been limited to obtaining a solution for the unknown parameters. Gao et al. (1992) thus proposes a robust test procedure. In his method, the theories of robust testing and statistical testing are integrated by combining their respective strengths and hereby providing a reliable procedure for the detection and identification of multiple blunders. First, the residuals are calculated through robust estimation, and the robust residuals are then used to formulate the test statistic. The detection and isolation of the blunders can thus be greatly improved compared with the conventional least squares method. The presence of blunders not only can be detected but also the blunders themselves can be precisely isolated. The $L_1$ estimator was discussed and compared with $L_2$. For detailed theoretical deduction and analysis of numerical results one can refer Gao et al. (1992).

This paper discusses and compares different robust estimators and conventional least squares using a framework that implements a random robust testing (RRT) algorithm. Section 2 recapitulates the basic concepts of robustness, describes fundamental assumption and tools used by robust statistics, and reviews some commonly used robust estimators. In Section 3, we develop the RRT algorithm that is a variation of the random sample consensus paradigm.
ROBUST ESTIMATION METHODS

Robust estimation dates back to the 1960s. During the last four decades, due to the work done by Tukey (1960), Huber (1964), Hample (1968) and Rousseeuw (1987), many robust estimation techniques have been proposed. They are not very sensitive to departure from the underlying assumptions. The advantage of robust estimation is that the negative effect of the blunders on the solution is greatly depressed or even eliminated (Chen, 1981). However, the statistical properties of a robust solution are not as clearly defined as the least-squares method, and the efficiency of the solution is inferior to a least-squares solution when only Gaussian noises are present.

Basic Concepts of Robustness and its Aims

The robustness can be understood from following aspects:

- Robustness may be defined as the property of a procedure that renders the answers. It gives insensitivity to departures, of a kind that occur in practice, from ideal assumption (Box, 1979).
- Robustness signifies insensitivity to small deviations from the assumptions (Huber, 1981). As to the implication of sensitivity,
  - It should have a reasonable good (optimal or nearly optimal) efficiency at the assumed model.
  - It should be robust in the sense that small deviation from the model assumptions should impair the performance slightly, that is, the latter should be close to the normal value calculated at the model.
  - Some what larger deviations from the model should not cause a catastrophe.

Huber also has given an exact explanation, that is,

- The aims of robust estimation are
  - To describe the structure best fitting the bulk of the data.
  - To identify deviating data points (outliers) or deviating substructures for further treatment.
  - To identify and give a warning about highly influential data points.
  - To deal with unsuspected serial correlations, or more generally, with deviations from the assumed correlation structure.

Fundamental Assumption and Tools

Robust estimation has developed many useful tools related with the estimator theory. Firstly, we seldom know the detailed tail structure of the density function in practise, and often assume a $\varepsilon$-contaminated Gaussian family that is defined as (Chen, 1981; Yang, 1993)

$$\Gamma_\varepsilon = \{F(u) = (1-\varepsilon)\Phi(u) + \varepsilon H(u)\}$$

where $\Phi$ is the standard Gaussian distribution, $H$ is symmetric contaminated distribution.

The influence function of an estimator $T$ given a distribution $F$ is given by

$$IF(x;T,F) = \lim_{t\to 0} \frac{T((1-t)F + tG) - T(F)}{t}$$

which describes the standardized effect of an infinitesimal contamination at the point $x$ on the asymptotic value of the estimator. Roughly speaking, the influence function is the first derivative of $T$ for an underlying distribution $F$.

The influence function is the basis for qualifying the robustness of an estimator. Based on the influence function, the asymptotic variance associated with an estimator $T$ can be determined:

$$V(T,F) = \int IF(x; T, F)^2 dF(x)$$

Asymptotic efficiency on the standard normal distribution is the ratio of the variance-covariance matrix of a given estimator $T$ to that of the least squares in the presence of Gaussian noise in the data. If there are no outliers and the data is only corrupted by Gaussian noise, we may expect the estimator to have an approximately normal distribution.

Through defining gross error sensitivity $\gamma*(T,F)$, rejection point $\rho*(T,F)$, local shift sensitivity $\lambda*(T,F)$ and breakdown point $\varepsilon*(T,F)$ (Hample, 1968), some criteria can be established to qualify the robustness of an estimator:
• Has a bounded influence function.
• Is moderately continuous (finite local shift sensitivity).
• Has a finite rejection point.
• The breakdown point is as large as possible.

These criteria can be used to formulate the optimal influence function and subsequently to define the robustness of an estimator.

Robust Estimators

Many methods have been proposed for the robust statistical testing of blunders (Rousseeuw, 1987). Regression diagnostics (Belsley, 1980) is one of earliest robust methods. It tries to iteratively detect possibly wrong data and reject them through analysis of globally fitted model, and depends heavily on a priori knowledge in choosing the thresholds for outlier rejection. It works as following:

1. Compute an initial fit to the whole set of data by LSs method.
2. Determine the residuals for each datum.
3. Remove data whose residuals are greater than a threshold; stop if no data should be removed.
4. Compute a new fit to the remaining data, and go to Step 2.

Huber considered a maximum-likelihood-type estimator (M-estimator) and approached the robustness problem by looking for a min-max solution, i.e., an estimator that minimizes the maximum asymptotic variance over some prescribed convex family of densities. The M-estimators are of the most sophisticated, and many applications can be found with this method in practice.

Let \( \nu \) be the residual vector, the standard least squares method tries to minimize \( \nu'\nu = \sum \nu_i^2 \), which is unstable if there are outliers present in the data. Outlying data give an effect so strong in the minimization that the parameters thus estimated are distorted. The M-estimator tries to reduce the effect of outliers by replacing the squared residuals by another function of the residuals, viz.,

\[
\min \sum \rho(\nu_i) \tag{1}
\]

where \( \rho \) is a symmetric, positive definite function with a unique minimum at zero, and is chosen to be less increasing than squares. Let \( p = [p_1, p_2, \cdots, p_u]^T \) be the parameter vector to be estimated. The M-estimator of \( p \) based on the function \( \rho(\nu) \) is the vector \( \hat{p} \) that is the solution of the following equations:

\[
\sum \psi(\nu_i) \frac{\partial \nu_i}{\partial p_j} = 0, \quad j = 1, 2, \cdots, u
\]

where the derivative \( \psi(\nu) = d\rho(\nu)/d\nu \) is equivalent to the influence function. It also measures the influence of a datum on the value of the parameter estimated. When an estimator is robust, it may be inferred that the influence of any single observation is insufficient to yield any significant offset.

Table 1. M-estimators

<table>
<thead>
<tr>
<th>M-estimator</th>
<th>( \rho(\nu) )</th>
<th>( \psi(\nu) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L_2 )</td>
<td>( \nu^2/2 )</td>
<td>( \nu )</td>
</tr>
<tr>
<td>( L_1 )</td>
<td>(</td>
<td>\nu</td>
</tr>
<tr>
<td>( L_{1,2} )</td>
<td>( 2[\sqrt{1+\nu^2}/2 - 1] )</td>
<td>( \nu/\sqrt{1+\nu^2}/2 )</td>
</tr>
<tr>
<td>( L_p )</td>
<td>(</td>
<td>\nu</td>
</tr>
<tr>
<td>Fair</td>
<td>( \frac{\nu^2}{2} \log(1 +</td>
<td>\nu</td>
</tr>
<tr>
<td>Huber</td>
<td>( k</td>
<td>\nu</td>
</tr>
<tr>
<td>Cauchy</td>
<td>( \frac{c^2}{2} \log(1 + (\nu/c)^2) )</td>
<td>( \nu/(1 + (\nu/c)^2) )</td>
</tr>
</tbody>
</table>

There are several constraints that a robust M-estimator should meet:
• Have a bounded influence function.
• The robust estimator should be unique. This implies that the objective function of parameter vector \( \mathbf{p} \) to be minimized should have a unique minimum. This requires that a \( \rho \) – function is convex in variable \( \mathbf{p} \). This is necessary because only requiring a \( \rho \) – function to have a unique minimum is not sufficient. This is the case with maximum when considering mixture distribution. The sum of uni-model probability distributions is very often multi-model. The convexity constraint is equivalent to imposing that \( \frac{\partial^2 \rho(.)}{\partial \mathbf{p}^2} \) be non-negative definite.

• Whenever \( \frac{\partial^2 \rho(.)}{\partial \mathbf{p}^2} \) is singular, the objective should have a gradient, i.e. \( \frac{\partial \rho(.)}{\partial \mathbf{p}} \neq 0 \), this avoids having to search through the complete parameter space.

A lot of M-estimators have been proposed although not all of them satisfy all the three constraints above. Table 1 gives some commonly used M-estimators (Chen, 1989; Zhang et al., 1995).

One important thing in the different M-estimators is the simultaneous estimation of \( \sigma \), the standard deviation of the residuals. If we can make a good estimate of the standard deviation of the errors of inliers, then datum whose error is larger than a certain number of standard deviations can be considered as outliers. Thus, the estimation of \( \sigma \) itself should be robust. The results of the M-estimators will depend on the method used to compute it. The robust standard deviation estimate is related to the median of the absolute values of the residuals, and is given by (Rousseeuw, 1987):

\[
\hat{\sigma} = 1.4826 \times \text{median} |\nu| 
\]  

where the constant 1.4826 is a coefficient to achieve the same efficiency as a least squares in the presence of only Gaussian noise, and the term \( 5/(n-u) \) is used to compensate the effect of little redundants. As proposed by Huber, one can start the iteration process with a convex function, iterate until convergence, and then apply a few iterations with one of those non-convex functions to eliminate the effect of large errors.

Two other classes of estimates that have received considerable attention are L-estimators (based on linear combination of order statistics) and R-estimates (derived from rank tests). Since these estimators require ordering operations, they may become computational unattractive for large sample. The least median of squares (LMedS) method estimates the parameters by solving the non-linear minimization problem (Rousseeuw, 1984; 1987):

\[
\min \text{median}_i |\nu_i| 
\]  

It turns out that this method is very robust to false matches as well as outliers due to bad localization. However, it’s probably impossible to write down a straightforward formula for the LMedS estimator. It must be solved by a search in the space of possible estimates generated from the data. Since this space is very large, only a randomly chosen subset of data can be analysed.

**RANDOM ROBUST TESTING FOR LINEAR SYSTEMS**

Let the parametric model under consideration be described by the following linear system in matrix form, namely,

\[
\mathbf{l} = \mathbf{Ax} + \nu \tag{4}
\]

where \( \mathbf{l} \) is an n-dimensional vector of observations that are assumed to be i.i.d., \( \mathbf{x} \) is an \( u \)-dimensional vector of unknown parameters, \( \nu \) is the error vector with dimension of \( n \), and \( \mathbf{A} \) is the design matrix.

M-estimators look for a parameter vector that minimizes the objective function in Equation 1. As described in Gao (1992), \( \mathbf{u} \) observations can give a close-form solution to Equation 4. So a subsample \( \mathbf{l}_1 \) of \( \mathbf{u} \) observations (called \( u \)-subsample) is randomly selected from the entire sample and is used to estimate the close-form solution of the unknowns

\[
\hat{\mathbf{x}} = (\mathbf{A}_1)^{-1}\mathbf{l}_1 \tag{5}
\]

where \( \mathbf{A}_1 \) is the corresponding design matrix. Because the residuals on observations \( \mathbf{l}_1 \) are all zeros, for each group of estimated values of unknowns, the objective function is calculated for all residuals as follows

\[
\sum \rho(\hat{\nu}) = \sum \rho(\hat{\nu}_z) \tag{6}
\]

where

\[
\hat{\nu} = \mathbf{l} - \mathbf{A}\hat{\mathbf{x}}
\]

\[
\hat{\nu}_z = \mathbf{l}_1 - \mathbf{A}_1\hat{\mathbf{x}} \tag{7}
\]
\( I_2 \) and \( A_2 \) are the remaining observations and the corresponding design matrix. That is, the estimator must yield the smallest value for Equation 6 computed for the entire observations.

As pointed out above, it is probably impossible to write down a straightforward formula for some robust estimators, including a majority of M-estimators and LMedS method. So they must be solved by a search in the space of possible estimates generated from the data. Theoretically, we should complete the trails for all the possible subsamples of the entire observation sample to obtain the possible minimum value of Equation 6. From the numerical example given in Table 3, we know that there exist different combinations up to \( C_{5985}^2 = 5985 \), so it is computationally impossible in practice especially when the sizes of the observation sample and the u-subsample are very large.

Gao et al. (1992) tackles this problem using linear programming. For example, \( L_I \) method can be described as follows:

\[
\min \sum_i |v_i| \quad \text{s.t.} \quad Ax = l - u
\]

which can be easily transformed into a standard linear programming problem. The unique solution which contains \( u \) observations selected from \( I \) is obtained by solving above linear programming problem, and the residuals of the remaining observations and their covariance matrix are calculated. Then a global test statistic \( T' \) and a local test statistic \( w' \) are formulated:

\[
T' = \hat{v}^T C_u^{-1} \hat{v} \sim \chi^2(0) \quad \text{w.r.t.} \quad H_0 : E(\hat{v}) = 0
\]

\[
w' = \frac{(\hat{v})}{\sqrt{(C_u^{-1})}} \sim N(0,1) \quad \text{w.r.t.} \quad H_0 : E[[v]] = 0
\]

The robust testing procedure has some advantages. It provides a robust basis for the calculation of the residuals, i.e., the predicted residual vector is robust in that it represents the actual discrepancies of the observations. Subsequently, the test statistic becomes a robust statistic and thus detection and isolation of blunders can be greatly improved. The method turns the problem of multiple-blunder detection into a one-dimensional test procedure. This avoids the difficulty of having to specify beforehand the possible blunders in the context of alternative hypothesis testing used by the conventional methods. On the other hand, the method also simplified the test procedure as only one global test and a one-dimensional test are involved.

Fischler and Bolles (1995) propose the random sample consensus paradigm to solve this problem by analysing several randomly chosen subsamples. They choose \( m \) trails in such a way that the probability of at least one u-subsample contains no blunders, that is, the probability of drawing at least one uncontaminated u-subsample, will be a large value \( \lambda \) close to 1. The expression for this probability is

\[
\lambda = 1 - [(1 - (1 - \epsilon)^{-m})]
\]

where \( \epsilon \) is a priori about the probability of blunders in the whole sample. From the numerical example given in Gao (1992), we know that \( \mu = 4, \epsilon = 4/21 \approx 0.19 \). So \( m \) can be computed when we assume \( \lambda = 0.999 \)

\[
m = \frac{\ln(1 - \lambda)}{\ln(1 - (1 - \epsilon)^{-m})} = 12.3097
\]

However, \( \epsilon \) is usually not a priori, so the number of u-subsamples \( \lceil m \rceil \) changes with respect to different value of \( \epsilon \) and \( \lambda \). Moreover, the larger the number of u-subsamples, the larger \( \epsilon \) and \( \lambda \). We can find in Table 2 that the number of u-subsamples needed to evaluate is much less than the possible combinations 5985 even if we ensure an extremely high reliability.

<table>
<thead>
<tr>
<th>( \lambda )</th>
<th>( \epsilon )</th>
<th>0.999</th>
<th>0.9999</th>
<th>0.99999</th>
<th>0.999999</th>
<th>0.9999999</th>
</tr>
</thead>
<tbody>
<tr>
<td>5%</td>
<td>10%</td>
<td>6</td>
<td>9</td>
<td>14</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>20%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Once all the \( m \) estimates \( \hat{x}_k, k = 1, \cdots, m \) are solved, we can choose the one \( \tilde{x} \) that minimizes the objective function in Equation 6, viz.

\[
\tilde{x} = \min \sum_i \rho(l - A\hat{x}_k, k = 1, \cdots, m)
\]
and the robust standard deviation estimate $\tilde{\sigma}$ is then calculated

$$\tilde{\sigma} = 1.4826 \text{median} |\tilde{u}_i|$$

where $\tilde{u}_i$ is computed by Equation 7.

Compared with Equation 2, the term $5/(n-u)$ is removed because we have many redundants. The outliers are those observations whose residuals are greater than $2.5\tilde{\sigma}$ which determines a 1.2% significance level (Isgro, 1999). After removing the observations with blunders from the original sample, the remaining ‘clean data set’ are used to solve the linear system by conventional least-squares method and the unknowns are robustly and efficiently estimated.

Here we make a variation to the RANSAC paradigm by integrating the basic idea of the robust testing procedure, called random robust testing (RRT) algorithm, and implement a feasible framework. The steps of the algorithm are:

1. Draw $m$ u-subsamples. Each u-subsample is drawn as follows:
   a. Get $n$ real values by a random number generator of uniform distribution in the close interval $[0, 1]$.
   b. Sort the $n$ real values in ascend or descend order, take the ranks of the first $u$ values as the indexes of the array of all observations, which form a u-subsample.
2. For $m$ u-subsamples, calculate the estimates $\hat{x}_k, k = 1, \ldots, m$ and the residuals in Equation 7.
3. Choose the solution $\tilde{x}$ of unknowns that minimizes the objective function among all estimates.
4. Calculate the robust standard deviation estimate $\tilde{\sigma}$ and the scaled robust residuals in Equation 11.
5. Find a clean data set by pruning blunders detected using scaled robust residual test or one-dimensional residual test. For the scaled robust residual test, an observation with scaled robust residuals greater than $2.5\tilde{\sigma}$ is rejected as a blunder; while for the one-dimensional residual test, a blunder is the observation whose residual is larger than the local test statistic.
6. The clean data set are used to solve the linear system by conventional least-squares method, and obtain the robust estimate of unknowns.

In geomatics, observations are often related with spatial positions, so the observations of a u-subsample generated in step 1 maybe very close to each other. Such a distribution should be avoid because the unknowns calculated using such observations is likely instable. In order to gain higher stability and reliability, random selection methods based on the bucketing technique (Zhang et al., 1995) should be utilized to obtain evenly distributed subsamples. The idea is that no more than one observation is selected for a given neighbourhood and each observation has almost the same probability to be selected. Because the observations used in our experiment do not have spatial distribution, we just apply the simple random selection method mentioned in Step 1.

The advantages of the RRT algorithm we implement relative to the robust testing procedure are

- RRT algorithm reacts the basic principles of robust estimation in a more simple and intuitive way.
- Reduce the number of u-subsamples needed to evaluate by using a Monte-Carlo type selection technique.
- Can directly identify blunders without global testing, and the computation of a posteriori covariance $C_{\text{L}}$ is not needed.

There are also some drawbacks of the RRT algorithm. They are

- A priori of $\varepsilon$ is needed or must be approximately assumed.
- The solution is not unique for every running of the algorithm, while linear programming can get a determinative one.

### NUMERICAL RESULTS AND ANALYSIS

We have implemented a general framework to evaluate the RRT algorithm. The source code of a simplified version of the framework is provided in Appendix A for the evaluation purpose.

**Results by the RRT Algorithm**

For consistency and convenience, we apply the RRT algorithm to all M-estimators and the LMedS method although M-estimators robust testing can be solved by weighted least-squares method (Rousseeuw, 1987). We follow the numerical results for $L_1$ and $L_2$ estimators described in Gao et al. (1992) and also take the numerical example of the design matrix and the raw observations shown in Table 3.
Table 3. Observations and design matrix (Gao, 1992)

<table>
<thead>
<tr>
<th>No.</th>
<th>Observation vector</th>
<th>Design matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>42</td>
<td>1 80 27 89</td>
</tr>
<tr>
<td>2</td>
<td>37</td>
<td>1 80 27 88</td>
</tr>
<tr>
<td>3</td>
<td>37</td>
<td>1 75 25 90</td>
</tr>
<tr>
<td>4</td>
<td>28</td>
<td>1 62 24 87</td>
</tr>
<tr>
<td>5</td>
<td>18</td>
<td>1 62 22 78</td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td>1 62 23 78</td>
</tr>
<tr>
<td>7</td>
<td>19</td>
<td>1 62 24 93</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>1 62 24 93</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>1 58 23 87</td>
</tr>
<tr>
<td>10</td>
<td>14</td>
<td>1 58 18 80</td>
</tr>
<tr>
<td>11</td>
<td>14</td>
<td>1 58 18 89</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>1 58 17 88</td>
</tr>
<tr>
<td>13</td>
<td>11</td>
<td>1 58 18 82</td>
</tr>
<tr>
<td>14</td>
<td>12</td>
<td>1 58 19 93</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>1 50 18 89</td>
</tr>
<tr>
<td>16</td>
<td>7</td>
<td>1 50 18 86</td>
</tr>
<tr>
<td>17</td>
<td>8</td>
<td>1 50 19 72</td>
</tr>
<tr>
<td>18</td>
<td>8</td>
<td>1 50 19 79</td>
</tr>
<tr>
<td>19</td>
<td>9</td>
<td>1 50 20 80</td>
</tr>
<tr>
<td>20</td>
<td>15</td>
<td>1 56 20 82</td>
</tr>
<tr>
<td>21</td>
<td>15</td>
<td>1 70 20 91</td>
</tr>
</tbody>
</table>

Table 4. Robust estimate \( \tilde{x} \) of unknowns

<table>
<thead>
<tr>
<th>Estimator</th>
<th>( \tilde{x} ) (1)</th>
<th>( \tilde{x} ) (2)</th>
<th>( \tilde{x} ) (3)</th>
<th>( \tilde{x} ) (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L_2 )</td>
<td>-4.60e+1</td>
<td>7.50e-1</td>
<td>7.50e-1</td>
<td>3.12e-2</td>
</tr>
<tr>
<td>( L_1 )</td>
<td>-3.96e+1</td>
<td>8.31e-1</td>
<td>5.73e-1</td>
<td>-6.08e-2</td>
</tr>
<tr>
<td>( L_{1-2} )</td>
<td>-3.64e+1</td>
<td>8.00e-1</td>
<td>6.86e-1</td>
<td>-1.04e-1</td>
</tr>
<tr>
<td>( L_p, p = 1.2 )</td>
<td>-3.96e+1</td>
<td>8.30e-1</td>
<td>5.80e-1</td>
<td>-6.20e-2</td>
</tr>
<tr>
<td>Fair</td>
<td>-3.64e+1</td>
<td>8.00e-1</td>
<td>6.86e-1</td>
<td>-1.04e-1</td>
</tr>
<tr>
<td>Huber</td>
<td>-3.64e+1</td>
<td>8.00e-1</td>
<td>6.86e-1</td>
<td>-1.04e-1</td>
</tr>
<tr>
<td>Cauchy</td>
<td>-3.96e+1</td>
<td>8.30e-1</td>
<td>5.80e-1</td>
<td>-6.20e-2</td>
</tr>
<tr>
<td>LMedS</td>
<td>-3.71e+1</td>
<td>7.88e-1</td>
<td>1.52e-1</td>
<td>3.38e-2</td>
</tr>
</tbody>
</table>

Tables 4 and 5 give a part of numerical results by an instance of running the algorithm when \( \varepsilon = 20\% \) and \( \lambda = .9999999999 \), which are reasonably good. In Table 5 and following tables, the bald zeros (i.e., 0.00e+0) represent observations composed of the best subsample, and the stroked grey numbers (e.g., 3.18e+0) are observations with outliers. The predicated robust residuals presented in each column of Table 5 have been divided by the associated robust standard deviation estimate \( \tilde{\sigma} \), viz.

\[
\tilde{\sigma} = \tilde{\sigma} / \tilde{\sigma}
\]

Nevertheless, as mentioned above, the result of one running of the algorithm is subject to the values of \( \varepsilon \) and \( \lambda \). When the value of \( \varepsilon \) is less than the actual contaminated percentage, less number of u-subsamples will be selected. So every u-subsample may contain at least one blunder. As a result, the estimation of the unknowns is highly unreliable and the result is useless. For example, when we assume \( \varepsilon = 5\% \), \( \lambda = .9999999999 \), that is, only 14 u-subsamples are selected, the result is nearly always significantly inferior to the one listed in Table 4. Thus we should assume a large contaminated percentage to draw a number of random u-subsamples sufficient to obtain good result when no a priori knowledge is available.

The scaled robust residuals in Table 5 (except \( L_2 \)) clearly identify all the observations with outliers (i.e.,
observations 1, 3, 4, and 21) and distinguish them from the inliers. So all the M-estimators including $L_1, L_{1-2}, L_p, p=1,2$, Fair, Huber, Cauchy, LMedS can correctly identify the observations with blunders even if they take different subsamples as the minimum number of observations for a robust close-form solution and thus get slightly different estimate to the unknowns when using the same group of 44 u-samples. While using $L_2$, method, only observations 4, 21 are correctly identified as blunders and other two blunders 1, 3 are hidden by its strong error smoothing ability.

A lot of running instances show that LMedS is not as stable as M-estimators and the robust standard deviation estimate need to be properly scaled to prevent copy results. The robust standard deviation estimate $\hat{\sigma}$ is a little smaller (calculated by Equation 10) when several inliers are incorrectly rejected except those true blunders. It would be a little larger (calculated by Equation 2) when a part of blunders are misclassified as inliers. So we replace term $5/(n-u)$ with $2.5/(n-u)$ and have gained better results. The reason may be that there are not so many blunders (only 4) relative to the large size (up to 21) of the sample, and the high breakdown point 0.5 of LMedS makes the rule (3) less distinguishable among several close-form solutions.

Table 5. Scaled robust residuals and blunders identification

<table>
<thead>
<tr>
<th>No.</th>
<th>$L_2$</th>
<th>$L_1$</th>
<th>$L_{1-2}$</th>
<th>$L_p, p=1.2$</th>
<th>Fair</th>
<th>Huber</th>
<th>Cauchy</th>
<th>LMedS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.33e+0</td>
<td>1.75e+0</td>
<td>1.85e+0</td>
<td>1.78e+0</td>
<td>1.85e+0</td>
<td>1.85e+0</td>
<td>1.78e+0</td>
<td>1.70e+0</td>
</tr>
<tr>
<td>2</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>2.36e+0</td>
</tr>
<tr>
<td>3</td>
<td>2.43e+0</td>
<td>3.00e+0</td>
<td>3.00e+0</td>
<td>3.05e+0</td>
<td>3.00e+0</td>
<td>3.00e+0</td>
<td>3.05e+0</td>
<td>4.81e+0</td>
</tr>
<tr>
<td>4</td>
<td>3.18e+0</td>
<td>4.33e+0</td>
<td>3.96e+0</td>
<td>4.28e+0</td>
<td>3.96e+0</td>
<td>3.96e+0</td>
<td>4.28e+0</td>
<td>5.70e+0</td>
</tr>
<tr>
<td>5</td>
<td>-8.06e-1</td>
<td>-6.94e-1</td>
<td>-6.74e-1</td>
<td>-6.79e-1</td>
<td>6.74e-1</td>
<td>-6.79e-1</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>6</td>
<td>-1.15e+0</td>
<td>-1.02e+0</td>
<td>-1.04e+0</td>
<td>-1.00e+0</td>
<td>1.04e+0</td>
<td>-1.00e+0</td>
<td>-8.96e-2</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>7</td>
<td>-1.12e+0</td>
<td>-5.70e-1</td>
<td>-5.37e-1</td>
<td>-5.61e-1</td>
<td>5.37e-1</td>
<td>-5.61e-1</td>
<td>2.89e-1</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>8</td>
<td>-6.59e-1</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>9</td>
<td>-1.15e+0</td>
<td>-8.34e-1</td>
<td>-9.34e-1</td>
<td>-8.25e-1</td>
<td>9.34e-1</td>
<td>-8.25e-1</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>10</td>
<td>2.34e+1</td>
<td>-1.15e-1</td>
<td>-1.87e-1</td>
<td>0.00e+0</td>
<td>1.87e-2</td>
<td>1.87e-2</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>11</td>
<td>1.02e+1</td>
<td>3.00e-1</td>
<td>4.87e-1</td>
<td>3.13e-1</td>
<td>4.87e-1</td>
<td>3.13e-1</td>
<td>-1.79e-1</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>12</td>
<td>0.00e+0</td>
<td>2.31e-1</td>
<td>2.62e-1</td>
<td>4.35e-1</td>
<td>2.62e-1</td>
<td>2.62e-1</td>
<td>4.35e-1</td>
<td>6.57e-1</td>
</tr>
<tr>
<td>13</td>
<td>1.20e+0</td>
<td>-1.65e+0</td>
<td>-1.52e+0</td>
<td>-1.61e+1</td>
<td>1.52e+0</td>
<td>-1.61e+0</td>
<td>1.80e+0</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>14</td>
<td>-1.24e+0</td>
<td>-1.02e+0</td>
<td>-7.33e-1</td>
<td>-9.95e-1</td>
<td>7.33e-1</td>
<td>-9.95e-1</td>
<td>1.52e+0</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>15</td>
<td>1.02e+1</td>
<td>6.74e-1</td>
<td>7.06e-1</td>
<td>6.74e-1</td>
<td>7.06e-1</td>
<td>7.06e-1</td>
<td>6.74e-1</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>16</td>
<td>-3.22e-1</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>8.71e-1</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>17</td>
<td>0.00e+0</td>
<td>-2.43e-1</td>
<td>-6.18e-1</td>
<td>-2.43e-1</td>
<td>6.18e-1</td>
<td>-2.43e-1</td>
<td>2.49e-1</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>18</td>
<td>-1.02e+1</td>
<td>0.00e+0</td>
<td>-2.24e-1</td>
<td>0.00e+0</td>
<td>2.24e-1</td>
<td>2.24e-1</td>
<td>0.00e+0</td>
<td>1.09e+0</td>
</tr>
<tr>
<td>19</td>
<td>0.00e+0</td>
<td>2.77e-1</td>
<td>0.00e+0</td>
<td>2.70e-1</td>
<td>0.00e+0</td>
<td>2.70e-1</td>
<td>5.88e-1</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>20</td>
<td>6.74e-1</td>
<td>9.22e-1</td>
<td>7.54e-1</td>
<td>9.10e-1</td>
<td>7.54e-1</td>
<td>9.10e-1</td>
<td>1.29e+0</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>21</td>
<td>-4.38e+0</td>
<td>-5.40e+0</td>
<td>-4.77e+0</td>
<td>-5.30e+0</td>
<td>4.77e+0</td>
<td>4.77e+0</td>
<td>-5.32e+0</td>
<td>0.00e+0</td>
</tr>
</tbody>
</table>

Table 6. Adjusted residuals and RMS errors using 'clean data set' by Ls method

<table>
<thead>
<tr>
<th>No.</th>
<th>$L_2$</th>
<th>$L_1$*</th>
<th>$L_2$</th>
<th>$L_1$*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.60e+0</td>
<td>---</td>
<td>6.49e-1</td>
<td>9.62e-1</td>
</tr>
<tr>
<td>2</td>
<td>-2.50e+0</td>
<td>1.15e+0</td>
<td>9.67e-2</td>
<td>4.73e-1</td>
</tr>
<tr>
<td>3</td>
<td>3.60e+0</td>
<td>---</td>
<td>-3.11e+0</td>
<td>-2.50e+0</td>
</tr>
<tr>
<td>4</td>
<td>---</td>
<td>---</td>
<td>-1.47e+0</td>
<td>-1.34e+0</td>
</tr>
<tr>
<td>5</td>
<td>-1.61e+0</td>
<td>-6.71e-1</td>
<td>2.30e+0</td>
<td>1.34e+0</td>
</tr>
<tr>
<td>6</td>
<td>-2.17e+0</td>
<td>-1.24e+0</td>
<td>9.76e-1</td>
<td>1.43e-1</td>
</tr>
<tr>
<td>7</td>
<td>-1.07e+0</td>
<td>-4.23e-1</td>
<td>-1.01e-1</td>
<td>-3.72e-1</td>
</tr>
<tr>
<td>8</td>
<td>-7.48e-2</td>
<td>5.76e-1</td>
<td>6.59e-1</td>
<td>9.64e-2</td>
</tr>
<tr>
<td>9</td>
<td>-1.34e+0</td>
<td>-1.05e+0</td>
<td>1.21e+0</td>
<td>5.86e-1</td>
</tr>
<tr>
<td>10</td>
<td>-3.28e-1</td>
<td>3.59e-1</td>
<td>1.69e+0</td>
<td>1.93e+0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RMSE</th>
<th>3.98e+0</th>
<th>1.56e+0</th>
<th>RMSE</th>
<th>3.98e+0</th>
<th>1.56e+0</th>
</tr>
</thead>
<tbody>
<tr>
<td>---: blunders</td>
<td>---: ---</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In Table 6, the adjusted residuals and RMS errors using cleaned data set by least-squares method are listed. The estimator $L_1, L_{p=1,2}, L_p=1.2$, Fair, Huber, Cauchy, and LMedS have produced same results. Using LSs method over ‘clean data set’, it’s obvious that the adjusted residuals and RMS of $L_2$ are much greater than those of robust estimators. This is because $L_2$ absorbs two blunders and makes the real magnitudes of the blunders spread among all observations, thus the predicated residuals don’t show the actual deviations of observations, especially of the observations with large variance or blunders.

Results by Robust Testing Procedure

In Step 3 of the RRT algorithm, we obtain best solutions of unknowns $\hat{X}$ that minimize the object function for different estimators using the same group of m u-subsamples. To test and illustrate its performance, we make global tests and local tests following the robust testing procedure. Global tests are performed on the predicated residuals by Equation 7 with a significance level of $\alpha = 0.05$ and $(\tilde{\chi}^2_{0.9975}(17))^{-1} = 27.587$. The global statistics ($T^c = 178.8$) for eight estimators using Equation 8 are all rejected, and thus the observations are confirmed containing blunders. Then one-dimensional tests using Equation 9 are carried out with significance level of $\alpha = 0.005$ (so $(\Phi_{0.9975})^2 = 2.807$) to identify the blunders. Table 7 lists results with the best solutions of unknowns presented respectively in Table 5.

The one-dimensional testing results in Table 7 clearly show that $L_1, L_{p=1,2}, L_p=1.2$, Fair, Huber, Cauchy estimators can correctly identify all the observations with blunders that are observations 2, 3, 4, and 21. Compared with the results listed in Table 5, two interesting cases can be immediately found from Table 7:

- Taking observations 2, 12, 17, and 19 as the best subsample to get the best unique solution, $L_1$ estimator also can correctly identify all blunders by local test statistic, while it cannot by scaled robust residual test.
- Taking observations 5, 9, 10, 15 to get the best unique solution, LMedS can only isolate observation 4 from inliers and hides other three blunders 2, 3, and 21 by local test statistic, while it cannot by scaled robust residual test.

A lot of running instances show that the local test statistic and scaled robust residual test have the same blunder identification ability when they are applied to M-estimators (except $L_2$). We cannot say which algorithm is better without being rather arbitrary when they are applied to solve linear system problem together with M-estimators.

Table 7. Statistical test $w'$ and blunders identification

<table>
<thead>
<tr>
<th>No</th>
<th>$L_1$</th>
<th>$L_{p=1,2}$</th>
<th>$L_{p=1,2}$</th>
<th>Fair</th>
<th>Huber</th>
<th>Cauchy</th>
<th>LMedS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\sigma$</td>
<td>2.13e+0</td>
<td>1.75e+0</td>
<td>1.85e+0</td>
<td>1.78e+0</td>
<td>1.85e+0</td>
<td>1.85e+0</td>
</tr>
<tr>
<td>1</td>
<td>-5.02e+0</td>
<td>3.73e+0</td>
<td>3.74e+0</td>
<td>3.65e+0</td>
<td>3.64e+0</td>
<td>3.61e+0</td>
<td>3.65e+0</td>
</tr>
<tr>
<td>2</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>-1.05e+0</td>
<td>-1.07e+0</td>
<td>-1.07e+0</td>
<td>-1.04e+0</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>3</td>
<td>3.89e+0</td>
<td>3.65e+0</td>
<td>4.02e+0</td>
<td>4.03e+0</td>
<td>3.98e+0</td>
<td>3.98e+0</td>
<td>4.03e+0</td>
</tr>
<tr>
<td>4</td>
<td>3.83e+0</td>
<td>5.18e+0</td>
<td>6.08e+0</td>
<td>5.69e+0</td>
<td>5.65e+0</td>
<td>5.65e+0</td>
<td>6.09e+0</td>
</tr>
<tr>
<td>5</td>
<td>-1.27e+0</td>
<td>-1.05e+0</td>
<td>-9.85e-1</td>
<td>-1.04e+0</td>
<td>-1.07e+0</td>
<td>-1.04e+0</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>6</td>
<td>-1.61e+0</td>
<td>-1.53e+0</td>
<td>-1.42e+0</td>
<td>-1.53e+0</td>
<td>-1.71e+0</td>
<td>-1.71e+0</td>
<td>-1.53e+0</td>
</tr>
<tr>
<td>7</td>
<td>-9.85e-1</td>
<td>-7.07e-1</td>
<td>7.07e-1</td>
<td>-7.07e-1</td>
<td>-7.07e-1</td>
<td>-7.07e-1</td>
<td>-7.07e-1</td>
</tr>
<tr>
<td>8</td>
<td>-5.76e-1</td>
<td>0.00e+0</td>
<td>-8.50e-1</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>9</td>
<td>-1.27e+0</td>
<td>-9.75e-1</td>
<td>0.00e+0</td>
<td>-1.09e+0</td>
<td>-1.31e+0</td>
<td>-1.31e+0</td>
<td>-1.09e+0</td>
</tr>
<tr>
<td>10</td>
<td>3.37e-1</td>
<td>-3.91e-3</td>
<td>-5.14e-1</td>
<td>0.00e+0</td>
<td>-1.68e-2</td>
<td>-1.68e-2</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>11</td>
<td>1.58e-1</td>
<td>2.06e-1</td>
<td>2.50e-1</td>
<td>2.39e-1</td>
<td>4.10e-1</td>
<td>4.10e-1</td>
<td>2.39e-1</td>
</tr>
<tr>
<td>12</td>
<td>0.00e+0</td>
<td>1.33e-2</td>
<td>-6.33e-2</td>
<td>2.99e-2</td>
<td>1.88e-1</td>
<td>1.88e-1</td>
<td>2.99e-2</td>
</tr>
<tr>
<td>13</td>
<td>-1.89e+0</td>
<td>-1.37e+0</td>
<td>-1.81e+0</td>
<td>-1.87e+0</td>
<td>-1.38e+0</td>
<td>-1.87e+0</td>
<td>-2.28e+0</td>
</tr>
<tr>
<td>14</td>
<td>-1.48e+0</td>
<td>-7.52e-1</td>
<td>-5.79e-1</td>
<td>-7.01e-1</td>
<td>-6.62e-1</td>
<td>-6.62e-1</td>
<td>-7.01e-1</td>
</tr>
<tr>
<td>15</td>
<td>1.16e-1</td>
<td>7.27e-1</td>
<td>9.88e-1</td>
<td>6.27e-1</td>
<td>8.44e-1</td>
<td>8.44e-1</td>
<td>6.27e-1</td>
</tr>
<tr>
<td>16</td>
<td>-4.37e+0</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>9.91e-3</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>9.91e-3</td>
</tr>
<tr>
<td>17</td>
<td>0.00e+0</td>
<td>-2.03e-1</td>
<td>-9.86e-1</td>
<td>-1.96e-1</td>
<td>-5.75e-1</td>
<td>-5.75e-1</td>
<td>-1.96e-1</td>
</tr>
<tr>
<td>18</td>
<td>-1.78e+0</td>
<td>0.00e+0</td>
<td>-5.13e-1</td>
<td>0.00e+0</td>
<td>-2.99e-1</td>
<td>-2.99e-1</td>
<td>0.00e+0</td>
</tr>
<tr>
<td>19</td>
<td>0.00e+0</td>
<td>2.99e-1</td>
<td>0.00e+0</td>
<td>3.06e-1</td>
<td>0.00e+0</td>
<td>0.00e+0</td>
<td>3.06e-1</td>
</tr>
<tr>
<td>20</td>
<td>1.24e+0</td>
<td>1.33e+0</td>
<td>8.70e-1</td>
<td>1.39e+0</td>
<td>1.09e+0</td>
<td>1.09e+0</td>
<td>1.39e+0</td>
</tr>
<tr>
<td>21</td>
<td>-6.26e+0</td>
<td>-2.81e+0</td>
<td>-3.43e+0</td>
<td>-3.45e+0</td>
<td>-3.20e+0</td>
<td>-3.20e+0</td>
<td>-3.45e+0</td>
</tr>
</tbody>
</table>

ASPRS 2006 Annual Conference
Reno, Nevada • May 1-5, 2006
CONCLUSIONS

The robust estimators have been proved to be superior over the conventional least-squares estimator in processing observations contaminated by multiple blunders in both theoretical and experimental aspects. In this paper, we have implemented a general framework that is readily applicable to all kinds of robust estimators in a simple way. The random robust testing algorithm is a variation of the random sample consensus paradigm by integrating the basic idea of a robust testing procedure to detect multiple blunders in linear systems. The algorithm can significantly improve the computational efficiency, and has equivalent blunder identification ability as validated by numerical experiments.

According to our numerical examples, different estimators may take different u-subsamples to get their best unique solutions when providing a same group of u-subsamples. The experimental results show that all M-estimators have nearly equivalent ability in identifying blunders in linear systems. In a summary, the following findings can be drawn:

• Least squares estimator $L_2$ is vulnerable to the violation of underlying assumptions, and spreads the blunders in the bad observations among other good or bad observations. So the majority of bad observations are protected from being identified and the predicated residuals don’t show the actual deviations of observations.

• Robust techniques are not very sensitive to departure from the underlying assumptions about sample distributions, and help understand the behavior of statistical procedures in real-life situations. The negative effect of the blunders on the robust solution is greatly depressed or even eliminated when enough number of random u-subsamples is drawn.

• Robust estimators can significantly improve the blunder identification ability as compared to the conventional LSs method. However, LMedS estimator seems not suitable for samples that are only slightly contaminated.

• It is difficult to select a $\rho$ function for general use or to pick out a best one without being rather arbitrary.

REFERENCES

APPENDIX A: FRAMEWORK FOR RANDOM ROBUST TESTING ALGORITHM

This appendix shows a general framework to evaluate the RRT algorithm using MATLAB. The robust testing procedure is also included for evaluating the performance of both algorithms. To add a new robust estimator, what we have to do is to write a function for the new estimator similar to the function ‘lsq’ or others and append its function name in the array named ‘est’.

%%-------------------------------------------------------------
% Main function
function main_rls( opt )
% opt - options [ep, la]

% A – design matrix
A = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1; 80 80 75 62 62 62 62 62 58 58 58 58 58 58 50 50 50 50 50 50 50; 27 27 25 24 22 23 24 24 23 18 18 17 18 19 18 18 19 19 20 20 20; 89 89 90 87 87 87 93 93 87 80 89 88 82 93 89 86 72 79 80 82 91];

% L – observation vector
L = [42 37 37 28 18 18 19 20 15 14 14 13 11 12 8 7 8 9 15 15];

% array of function names for eight estimators
est = ['lsq'; 'lar'; 'l12'; 'lpn'; 'fai'; 'hub'; 'cau'; 'lms'];

% solutions by robust estimators
sub_rls(A, L, est, opt);

%%-------------------------------------------------------------
function sub_rls( A, L, est, opt )
% solution by a robust estimator

% A - design matrix
% L - observation vector
% est - name of a robust estimator
% opt - options [ep, la]
% ep - contaminated percentage
% la - confidence level

if nargin<4 | isempty(opt),
  ep = .2; la = .9999999999;
else
  ep = opt(1); la = opt(2);
end

[N, U] = size(A);
C = size(est, 1);
M = ceil(log(1-la)/log(1-(1-ep)^U));

% re-permute sample M times
[ig, P] = sort(rand(N,M));

for k = 1:M,
  l1 = L(P(1:U,k));
  a1 = A(P(1:U,k), :);
  x1(:,k) = a1 \ l1;
  % residuals matrix for subsamples
  v1(:,k) = L - A*x1(:,k);
end

% random robust testing algorithm
for k = 1:C,
  [p, sig, r] = feval(est(k,:), v1, N, U);
  p1(k) = p;
  sig1(k) = sig;
  r1(:, k) = r;
end

null = 0;

% best u-subsamples for each estimator
i1 = P(1:U,p1);
i2 = P(U+1:N,p1);
x1 = x1(:,p1);

% estimated robust residuals
save c:\rrt\res0.txt p1 null i1 null x1 null sig1 null r1 '-ascii'

% solutions by conventional least-squares for the clean data set after removing blunders
for k = 1:C,
  I = find(abs(r1(:,k))<=2.5);
  l = L(I);
  a = A(I, :);
  x(:,k) = (a'*a) \ (a'*l);
  v(:,k) = L - A*x(:,k);
  num(k) = length(I);
  rms(k) = sum(v(I,k).^2) / (num(k)-U);
end

save c:\rrt\res1.txt num null x null rms null v '-ascii'

% robust testing procedure
T0 = zeros(C, 1);
w0 = zeros(N-U, C);

for k = 1:C,
  L1 = L(i1(:,k));
  A1 = A(i1(:,k), :);
  x = A1 \ L1;
  L2 = L(i2(:,k));
  A2 = A(i2(:,k), :);
end
% residuals of remaining observations
v2 = L2 - A2*x;
Cv2 = eye(N-U) + A2*inv(A1)*eye(U) *(A2*inv(A1))';
T = v2*inv(Cv2)*v2;
d = diag(Cv2); d = d(:);
w = v2 ./ sqrt(d);
T0(k) = T;
w0(:, k) = w;
end

save c:\temp\res2.txt T0 null w0 '-ascii'

%-------------------------------------------------------------
function [p, sig, r] = lsq( v, n, u )
% L2, squared regression residuals
% p - the best subsample
% sig- robust standard deviation
% r - scaled robust residuals
% v - residuals, each column is a residual vector for each subsample
% n - number of observations
% u - number of unknowns

[ig, p] = min(sum(v.^2));
sig = rsig( v(:,p), n, u );
r = v(:,p) / sig;

%-------------------------------------------------------------
function [p, sig, r] = lar( v, n, u )
% L1, Summation of absolute regression residuals

[ig, p] = min(sum(abs(v)));
sig = rsig( v(:,p), n, u );
r = v(:,p) / sig;

%-------------------------------------------------------------
function [p, sig, r] = l12( v, n, u )
% L1-2, L1-L2 norm

[ig, p] = min(sum(sqrt(1+v.^2/2)-1));
sig = rsig( v(:,p), n, u );
r = v(:,p) / sig;

%-------------------------------------------------------------
function [p, sig, r] = lpn( v, n, u, lp )
% Lp norm
% lp - order of Lp norm

if nargin<4 | isempty(lp),
    lp = 1.2;
end

%-------------------------------------------------------------
function [p, sig, r] = fai( v, n, u )
% Fair estimator

k = 1.3998;  %tuning constant
v = abs(v);

[ig, p] = min(sum(v/k-log(1+v/k)));  %sig = rsig( v(:,p), n, u );
r = v(:,p) / sig;

%-------------------------------------------------------------
function [p, sig, r] = hub( v, n, u )
% Huber estimator

k = 1.345;  %tuning constant
v = abs(v);
for j = 1:size(v,2),
    I = find(v(:,j) <= k);
    I1 = find(v(:,j) > k);
    h(j) = sum(v(I, j).^2/2) + sum(k*(v(I1,j)-k/2));
end

[ig, p] = min(h);
sig = rsig( v(:,p), n, u );
r = v(:,p) / sig;

%-------------------------------------------------------------
function [p, sig, r] = cau( v, n, u )
% Cauchy function

k = 2.3849;  %tuning constant
v = abs(v);

[ig, p] = min(sum(log(1+(v/k).^2)));
sig = rsig( v(:,p), n, u );
r = v(:,p) / sig;

%-------------------------------------------------------------
function [p, sig, r] = lms( v, n, u )
% least median of the squared residuals

[ig, p] = min(median(v.^2));
sig = 1.4826*(1+2.5/(n-u))*sqrt(ig);
r = v(:,p) / sig;

%-------------------------------------------------------------
function [sig] = rsig( v, n, u )
% robust standard deviation

sig = 1.4826*median(abs(v));  %*(1+5/(n-u)