Fast Robust Regression Algorithms for Problems with Toeplitz Structure

Nicola Mastronardi
IAC–CNR, Bari & CS Dept., Katholieke Universiteit Leuven
Dianne P. O'Leary
Department of Computer Science, University of Maryland, College Park, USA

Due giornate di Algebra Lineare Numerica

Padova, 26-27 febbraio 2007

Outline

Outline

- Consider the approximation problem

$$Ax \approx b$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ ($m \geq n$) are given and $x \in \mathbb{R}^n$ is to be determined.

- Consider the approximation problem

$$Ax \approx b$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ ($m \geq n$) are given and $x \in \mathbb{R}^n$ is to be determined.

- We define the residual

$$r = b - Ax.$$

- The usual approach to the problem is least squares, in which we minimize the 2-norm of the residual over all choices of x,

$$\min_x \|r\|_2, \quad r = b - Ax.$$

- The usual approach to the problem is least squares, in which we minimize the 2-norm of the residual over all choices of x,

$$\min_x \|r\|_2, \quad r = b - Ax.$$

- This produces the minimum variance unbiased estimator of the solution when the errors in the observation b are independent and normally distributed with mean 0 and constant variance.

- The usual approach to the problem is least squares, in which we minimize the 2-norm of the residual over all choices of x,

$$\min_x \|r\|_2, \quad r = b - Ax.$$

- This produces the minimum variance unbiased estimator of the solution when the errors in the observation b are independent and normally distributed with mean 0 and constant variance.
- However, the least squares solution is not robust if outliers occur, i.e., if some of the components of b are contaminated by large error.

- The usual approach to the problem is least squares, in which we minimize the 2-norm of the residual over all choices of x,
$$\min_x \|r\|_2, \quad r = b - Ax.$$
- This produces the minimum variance unbiased estimator of the solution when the errors in the observation b are independent and normally distributed with mean 0 and constant variance.
- However, the least squares solution is not robust if outliers occur, i.e., if some of the components of b are contaminated by large error.
- Alternate approaches have been proposed which judge the size of the residual in a way that is less sensitive to these components.

Outline

- In order to reduce the influence of the outliers, might replace the least squares problem

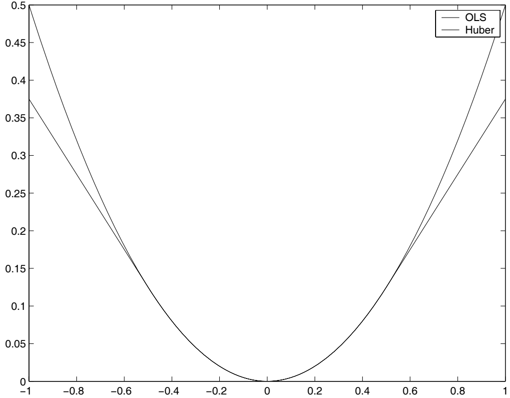
$$\min_x \|r\|_2$$

might be replaced by

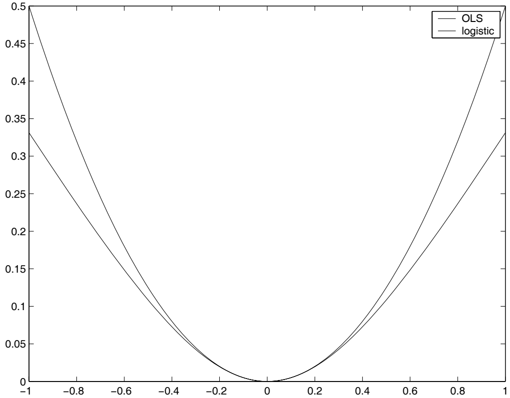
$$\min_x \sum_{i=1}^{m} \rho(r_i(x)) \tag{1}$$

subject to $r = b - Ax$, where ρ is a given function.
We call (1) a weighted Least Squares (WLS) problem.

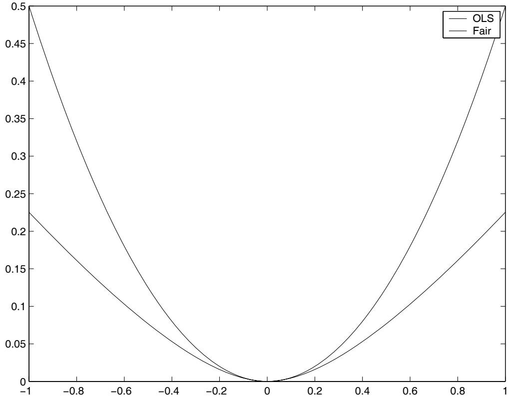
- Huber: $\rho(z) = \begin{cases} z^2/2, & |z| \leq \beta, \\ \beta|z| - \beta^2/2, & |z| > \beta, \end{cases}$



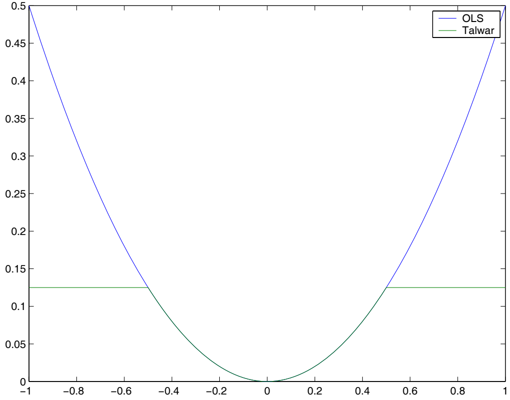
- logistic: $\rho(z) = \beta^2 \log(\cosh(z/\beta))$



- Fair: $\rho(z) = \beta^2(|z|/\beta - \log(1 + |z|/\beta))$.



- Talwar: $\rho(z) = \{ \; z^2/2, \quad |z| \leq \beta,$



In this talk we consider how the solution to weighted problems can be computed efficiently, in particular when the matrix A has small displacement rank (Toeplitz–like matrices)

- This structure has been effectively exploited in solving least squares problems (Kailath,Sayed '99, Ng '96),

In this talk we consider how the solution to weighted problems can be computed efficiently, in particular when the matrix A has small displacement rank (Toeplitz–like matrices)

- This structure has been effectively exploited in solving least squares problems (Kailath,Sayed ’99, Ng ’96),
- weighted least squares problems (Benzi,Ng ’06),

In this talk we consider how the solution to weighted problems can be computed efficiently, in particular when the matrix A has small displacement rank (Toeplitz–like matrices)

- This structure has been effectively exploited in solving least squares problems (Kailath,Sayed '99, Ng '96),
- weighted least squares problems (Benzi,Ng '06),
- total least squares problems (Kalsi, O'Leary, '06, Mastronardi et al. '04,'06)

In this talk we consider how the solution to weighted problems can be computed efficiently, in particular when the matrix A has small displacement rank (Toeplitz–like matrices)

- This structure has been effectively exploited in solving least squares problems (Kailath,Sayed '99, Ng '96),
- weighted least squares problems (Benzi,Ng '06),
- total least squares problems (Kalsi, O'Leary, '06, Mastronardi et al. '04,'06)
- regression using other norms (Pruessner, O'Leary, '03)

Outline

- Assume first we have a well-conditioned model in which we know which components of the vector b have large errors. This situation arises, for example, if there are known defects in a set of detectors which reduce the effectiveness of some of them without making their data worthless.

- Assume first we have a well-conditioned model in which we know which components of the vector b have large errors. This situation arises, for example, if there are known defects in a set of detectors which reduce the effectiveness of some of them without making their data worthless.
- Suppose that p of our observations are defective, where $p \ll m$.
  Let D be a diagonal weighting matrix that takes this into account, with $d_{jj} = 1$ for the trusted observations b_j and $0 < d_{jj} < 1$ for the others.

- Goal: solve

$$\min_{x} \|D^{1/2}(b - Ax)\|_2 \,. \tag{2}$$

- Goal: solve

$$\min_x \|D^{1/2}(b - Ax)\|_2 \,. \tag{2}$$

- By setting the derivatives to zero we obtain the normal equations

$$A^T DAx = A^T Db. \tag{3}$$

- Goal: solve
$$\min_x \left\|D^{1/2}(b - Ax)\right\|_2 . \tag{2}$$
- By setting the derivatives to zero we obtain the normal equations
$$A^T DAx = A^T Db. \tag{3}$$
- If A is structured, then fast algorithms exist for solving the problem when $D = I$.
  We show that such algorithms can still be efficiently used when $p \ll m$ entries of the main diagonal of D are different from 1.

Without loss of generality, we can order the observations so that the first p components $d_{jj} \neq 1$.
Let V contain the first p columns of the $(m+n) \times (m+n)$ identity matrix, let S_p be the $p \times p$ diagonal matrix with entries $\sqrt{d_{jj}^{-1} - 1}$, and let $W = VS_p$.
In the following, three methods for solving the problem are described.

Outline

- If none of the weights is zero, the WLS problem is equivalent to solving

$$\begin{bmatrix} D^{-1} & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}, \qquad (4)$$

where $D^{-1}y = r$.

- If none of the weights is zero, the WLS problem is equivalent to solving

$$\begin{bmatrix} D^{-1} & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}, \qquad (4)$$

  where $D^{-1}y = r$.
- If we use constraint preconditioning (Dyn and Ferguson, '83) on this system, we obtain the iteration matrix

$$N = \begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} D^{-1} & A \\ A^T & 0 \end{bmatrix} \equiv B_I^{-1} B_D.$$

- If none of the weights is zero, the WLS problem is equivalent to solving

$$\begin{bmatrix} D^{-1} & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}, \qquad (4)$$

  where $D^{-1}y = r$.
- If we use constraint preconditioning (Dyn and Ferguson, '83) on this system, we obtain the iteration matrix

$$N = \begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} D^{-1} & A \\ A^T & 0 \end{bmatrix} \equiv B_I^{-1} B_D.$$

- A similar approach is used in (Benzi and Ng, '06), with I replaced by γI, where γ is the average value of the diagonal entries of D^{-1}.

- Taking $\gamma = 1$, as in the original constraint preconditioning procedure, has a great advantage, though. Since

$$B_D = B_I + WW^T,$$

the matrix $N = B_I^{-1} B_D$ is the identity matrix plus a rank p correction.
Therefore, all but p eigenvalues of N are equal to 1, and an iterative method such as GMRES will terminate with the solution in $p + 1$ iterations.

- Taking $\gamma = 1$, as in the original constraint preconditioning procedure, has a great advantage, though. Since

$$B_D = B_I + WW^T,$$

the matrix $N = B_I^{-1} B_D$ is the identity matrix plus a rank p correction.
Therefore, all but p eigenvalues of N are equal to 1, and an iterative method such as GMRES will terminate with the solution in $p + 1$ iterations.
- The bulk of the work in each iteration is multiplication of vectors by the matrices B_D and B_I^{-1}.
To multiply by B_D, we multiply by the Toeplitz matrix A and then by A^T, which can be done in $O((m+n)\log(m+n))$ operations.

- Application of the preconditioning matrix B_I^{-1} is related to solving a standard least squares problem involving the Toeplitz matrix A.

- Application of the preconditioning matrix B_I^{-1} is related to solving a standard least squares problem involving the Toeplitz matrix A.
- We can solve

$$B_I \begin{bmatrix} z \\ w \end{bmatrix} = \begin{bmatrix} c \\ d \end{bmatrix}$$

  in $O(mn + n^2)$ operations; first we compute R, the upper triangular factor from the QR factorization of A, by means of the generalized Schur algorithm (GSA), and then, since $A^T A = R^T R$, we solve $R^T R w = A^T c - d$ and then form $z = c - Aw$.
  Thus the entire algorithm is $O(pmn)$.

Outline

[Method 2: Use the Sherman-Morrison-Woodbury Formula]

- An alternative approach to solving (4) uses the Sherman-Morrison-Woodbury formula to solve a linear system involving B_D using an algorithm to solve a linear system involving B_I.

- An alternative approach to solving (4) uses the Sherman-Morrison-Woodbury formula to solve a linear system involving B_D using an algorithm to solve a linear system involving B_I.
- Since $B_D = B_I + WW^T$, then

$$B_D^{-1} = B_I^{-1} - B_I^{-1} W (I + W^T B_I^{-1} W)^{-1} W^T B_I^{-1},$$

.

- The linear system $B_D f = g$ can be solved at a cost about equal to that of solving $p+1$ linear systems involving B_I:
  - Find the $(m+n) \times p$ matrix Z that solves the linear system $B_I Z = W$.
  - Find the vector $\hat{f}$ that solves $B_I \hat{f} = g$.
  - Solve the $p \times p$ linear system $(I + W^T Z)q = W^T \hat{f}$.
  - Form $f = \hat{f} - Zq$.

  This approach will work when none of the weights is too close to zero, but will be inaccurate when $I + W^T Z$ is too ill-conditioned.

- Suppose the weights are 1s and 0s.
  Replace the 0 weights by some small number ϵ, so that $W = \sqrt{\epsilon^{-1} - 1}\, V$. If $0 < \epsilon \ll 1$, the SMW formula is still valid, and as $\epsilon \to 0$,

$$B_I^{-1} W (I + W^T B_I^{-1} W)^{-1} W^T B_I^{-1} \to B_I^{-1} V (V^T B_I^{-1} V)^{-1} V^T B_I^{-1}.$$

- Suppose the weights are 1s and 0s.
  Replace the 0 weights by some small number ϵ, so that $W = \sqrt{\epsilon^{-1} - 1}\, V$. If $0 < \epsilon \ll 1$, the SMW formula is still valid, and as $\epsilon \to 0$,

$$B_I^{-1} W (I + W^T B_I^{-1} W)^{-1} W^T B_I^{-1} \to B_I^{-1} V (V^T B_I^{-1} V)^{-1} V^T B_I^{-1}.$$

- Thus we have an algorithm for zero weights:
  - Find the $(m+n) \times p$ matrix Z that solves the linear system $B_I Z = V$.
  - Find the vector $\hat{f}$ that solves $B_I \hat{f} = g$.
  - Solve the $p \times p$ linear system $(V^T Z) q = V^T \hat{f}$.
  - Form $f = \hat{f} - Zq$.

- Suppose the weights are 1s and 0s.
  Replace the 0 weights by some small number ϵ, so that $W = \sqrt{\epsilon^{-1} - 1}\, V$. If $0 < \epsilon \ll 1$, the SMW formula is still valid, and as $\epsilon \to 0$,

$$B_I^{-1} W (I + W^T B_I^{-1} W)^{-1} W^T B_I^{-1} \to B_I^{-1} V (V^T B_I^{-1} V)^{-1} V^T B_I^{-1}.$$

- Thus we have an algorithm for zero weights:
  - Find the $(m+n) \times p$ matrix Z that solves the linear system $B_I Z = V$.
  - Find the vector $\hat{f}$ that solves $B_I \hat{f} = g$.
  - Solve the $p \times p$ linear system $(V^T Z) q = V^T \hat{f}$.
  - Form $f = \hat{f} - Zq$.
- Thus, linear systems involving B_I can be solved in $O(mn)$ operations when A is Toeplitz, so for either of these two algorithms the cost is $O(pmn)$ operations.

Outline

- In order to reduce the effects of the observations in model (2) that are contaminated by large errors, the corresponding diagonal entries in the weight matrix D must have positive values much smaller than 1.

- In order to reduce the effects of the observations in model (2) that are contaminated by large errors, the corresponding diagonal entries in the weight matrix D must have positive values much smaller than 1.
- We can write our WLS problem as

$$\min_x \|D^{1/2}(b - Ax)\|_2 = \min_x \|(I - S^2)^{1/2}(b - Ax)\|_2, \qquad (5)$$

where S is a diagonal matrix with diagonal entries $\sqrt{1 - d_{jj}}$, $j = 1, \ldots, m$.

- The normal equation (3) becomes

$$(A^T A - A^T S^2 A)x = A^T (I - S^2) b.$$

Since A is a Toeplitz matrix, the matrix

$$M_p = A^T A - A^T S^2 A \tag{6}$$

is a rank–p modification of the Toeplitz–like matrix $A^T A$.

- The normal equation (3) becomes

$$(A^TA - A^TS^2A)x = A^T(I - S^2)b.$$

  Since A is a Toeplitz matrix, the matrix

$$M_p = A^TA - A^TS^2A \qquad (6)$$

  is a rank–p modification of the Toeplitz–like matrix A^TA.
- Therefore, the displacement rank of (6) is at most $4 + 2p$.

- The normal equation (3) becomes

$$(A^TA - A^TS^2A)x = A^T(I - S^2)b.$$

Since A is a Toeplitz matrix, the matrix

$$M_p = A^TA - A^TS^2A \tag{6}$$

is a rank–p modification of the Toeplitz–like matrix A^TA.

- Therefore, the displacement rank of (6) is at most $4 + 2p$.
- The least squares problem (5) can be solved via the seminormal equation

$$R^TRx = A^T(I - S^2)b, \tag{7}$$

where $R^T \in \mathbb{R}^{n \times n}$ is the Cholesky factor of (6). In general, if $p \ll m$, the Cholesky factorization can be computed in $O(mn + n^2)$ flops by means of GSA (Kailath et al., '95).

Generalized Schur Algorithm applied to WLS problems

Let

$$A = T = \begin{bmatrix} t_n & t_{n-1} & \cdots & t_2 & t_1 \\ t_{n+1} & t_n & \ddots & \ddots & t_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ t_{m+n-2} & \ddots & \ddots & t_m & t_{m-1} \\ t_{m+n-1} & t_{m+n-2} & \cdots & t_{m+1} & t_m \end{bmatrix}$$

- The GSA allows to compute the R factor of the QR factorization of A relying only on the knowledge of the generators of A^TA, instead of the knowledge of the matrix.

- The GSA allows to compute the R factor of the QR factorization of A relying only on the knowledge of the generators of A^TA, instead of the knowledge of the matrix.
- Define $v \equiv T^T Te_1/\sqrt{\sum_{i=n}^{\hat{m}} t_i^2}$, where e_1 is the first column of the identity matrix, the generators of M with respect to Z_n are

$$
\begin{aligned}
g_1^{(p)} &\equiv v, \\
g_2^{(p)} &\equiv \begin{bmatrix} 0 & t_{n-1} & \cdots & t_1 \end{bmatrix}^T, \\
g_1^{(n)} &\equiv Z_n Z_n^T v, \\
g_2^{(n)} &\equiv \begin{bmatrix} 0 & t_{\hat{m}} & t_{\hat{m}-1} & \cdots & t_{m+2} & t_{m+1} \end{bmatrix}^T.
\end{aligned}
$$

- The GSA allows to compute the R factor of the QR factorization of A relying only on the knowledge of the generators of A^TA, instead of the knowledge of the matrix.
- Define $v \equiv T^T Te_1/\sqrt{\sum_{i=n}^{\hat{m}} t_i^2}$, where e_1 is the first column of the identity matrix, the generators of M with respect to Z_n are

$$\begin{aligned} g_1^{(p)} &\equiv v, \\ g_2^{(p)} &\equiv \begin{bmatrix} 0 & t_{n-1} & \cdots & t_1 \end{bmatrix}^T, \\ g_1^{(n)} &\equiv Z_n Z_n^T v, \\ g_2^{(n)} &\equiv \begin{bmatrix} 0 & t_{\hat{m}} & t_{\hat{m}-1} & \cdots & t_{m+2} & t_{m+1} \end{bmatrix}^T. \end{aligned}$$

- Using these four generators, GSA can compute the R factor of the QR factorization of T. Since $p = 4 \ll n$, the computational complexity of GSA is $O(n^2)$.

The computation of the Cholesky factor of

$$M_p = A^T A - A^T S^2 A$$

can be done in a similar way, since its displacement rank is $\leq 4 + 2p$. The generators of M_p are

$$\begin{aligned}
g_1^{(p)} &\equiv v, \\
g_2^{(p)} &\equiv \begin{bmatrix} 0 & t_{n-1} & \cdots & t_1 \end{bmatrix}^T, \\
g_{k+2}^{(p)} &\equiv \sqrt{1 - d_{kk}} Z_n A(k,:)^T, & k = 1, \ldots, p, \\
g_1^{(n)} &\equiv Z_n Z_n^T v, \\
g_2^{(n)} &\equiv \begin{bmatrix} 0 & t_{\hat{m}} & t_{\hat{m}-1} & \cdots & t_{m+2} & t_{m+1} \end{bmatrix}^T, \\
g_{k+2}^{(n)} &\equiv \sqrt{1 - d_{kk}} A(k,:)^T, & k = 1, \ldots, p,
\end{aligned}$$

with $v \equiv T^T T e_1 / \sqrt{\sum_{i=n}^{\hat{m}} t_i^2}$. GSA requires $O(mn + n^2)$ operations.

Outline

[Weighting for the Ill-Conditioned or Rank-Deficient Case]

If the matrix A is ill–conditioned or rank–deficient, then a regularization term is often added, transforming our problem to

$$\min_x \|D^{1/2}(b - Ax)\|_2^2 + \lambda\|Fx\|_2^2\,, \tag{8}$$

where F is a matrix designed to control the size of x and $\lambda > 0$ defines the relative importance of the two terms.
By setting the derivatives to zero we obtain the normal equations

$$(A^T DA + \lambda F^T F)x = A^T Db.$$

Outline

- If A and F are structured, then fast algorithms exist for solving the problem when $D = I$.

- If A and F are structured, then fast algorithms exist for solving the problem when $D = I$.
- If none of the weights is zero, the problem is equivalent to solving

$$\begin{bmatrix} D^{-1} & A \\ A^T & \lambda F^T F \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

- If A and F are structured, then fast algorithms exist for solving the problem when $D = I$.
- If none of the weights is zero, the problem is equivalent to solving

$$\begin{bmatrix} D^{-1} & A \\ A^T & \lambda F^T F \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

- If we use constraint preconditioning on this system, we obtain

$$N = \begin{bmatrix} I & A \\ A^T & \lambda F^T F \end{bmatrix}^{-1} \begin{bmatrix} D^{-1} & A \\ A^T & \lambda F^T F \end{bmatrix} \equiv B_I^{-1} B_D.$$

Again, N is the identity plus a rank p matrix, so an iterative method such as GMRES will find the solution in $p+1$ iterations.

Outline

Again we can use the Sherman-Morrison-Woodbury formula to solve the linear system involving B_D using at a cost about equal to that of solving $p+1$ linear systems involving B_I.

Outline

The generalized Schur algorithm can be easily updated to handle this case. In fact, if $F \equiv I$, the generators are the same except that we redefine

$$v \equiv \frac{T^T T e_1 + \mu e_1}{\sqrt{\sum_{i=n}^{\hat{m}} t_i^2 + \mu}}.$$

If F has low displacement rank, it is always possible to constuct a fast GSA.

Outline

- The described techniques are applied to the robust regression problem.

$$\min_x \sum_{i=1}^{m} \rho(r_i(x))$$

subject to $r = b - Ax$, where ρ is a given function.

- The described techniques are applied to the robust regression problem.

$$\min_x \sum_{i=1}^{m} \rho(r_i(x))$$

  subject to $r = b - Ax$, where ρ is a given function.
- It is possible to add a regularization term.

- We consider solving our minimization problem by Newton's method as in (O'Leary, '90).

- We consider solving our minimization problem by Newton's method as in (O'Leary, '90).
- We compute the gradient vector z defined by

$$z_j = \rho'(r_j),$$

and define $D(r)$ to be a diagonal matrix with entries $d_{jj} = \rho''(r_j)$.

- The function

$$f(x) = \sum_{i=1}^{m} \rho(r_i(x))$$

has gradient $g = -A^T z$ and Hessian matrix $H = A^T DA$, so the Newton step for solving (1) is

$$s = -H^{-1} g = (A^T DA)^{-1} A^T z,$$

which causes a change

$$\Delta r = As = -A(A^T DA)^{-1} A^T z$$

in the residual vector r.

- Observe now that we can determine s by solving the linear system

$$\begin{bmatrix} D^{-1} & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} y \\ s \end{bmatrix} = \begin{bmatrix} D^{-1}z \\ 0 \end{bmatrix},$$

which is the same problem considered before except for the scaling by D^{-1} on the right-hand side.

Outline

- We performed some numerical experiments with the algorithms designed in this paper, considering an example on Toeplitz least squares problems with no outliers (Ng, ’96).

- We performed some numerical experiments with the algorithms designed in this paper, considering an example on Toeplitz least squares problems with no outliers (Ng, ’96).
- The entries t_k of the 100×30 rectangular Toeplitz matrices are generated from the second order autoregressive process given by

$$t_k - 1.4t_{k-1} + 0.5t_{k-2} = y_k,$$

where t_{-1}, t_0 and $y_k, k = 1, 2, \ldots, m+n-1$, are sampled from the Gaussian distribution with mean 0 and standard deviation 1.

We employ this input process to generate the Toeplitz data matrices. In the tests, we choose the solution x to be the vector with all entries equal to 1. The right–hand–side b in (2) is computed from perturbing Tx by Gaussian noise:

$$b = T_{100,30}x + e,$$

with e_k generated from the Gaussian distribution with 0 mean and standard deviation $1.0e-4,\ k = 1, 2, \ldots, 100.$

- Results obtained by computing a solution from robust regression using the Talwar function.

$$\rho(z) = \begin{cases} z^2/2, & |z| \leq \beta, \\ \beta^2/2, & |z| > \beta, \end{cases} \quad , \qquad \beta = 0.5 \times 10^{-4}$$

- Results obtained by computing a solution from robust regression using the Talwar function.

$$\rho(z) = \begin{cases} z^2/2, & |z| \leq \beta, \\ \beta^2/2, & |z| > \beta, \end{cases}, \qquad \beta = 0.5 \times 10^{-4}$$

- For each case, we performed 100 simulations replacing respectively 2%, 5% and 10% of the entries of the vector b by values from the Gaussian distribution with mean 0 and standard deviation $1.2, 1.5, 2.0, 2.5, 5, 10$, respectively.

- Results obtained by computing a solution from robust regression using the Talwar function.

$$\rho(z) = \begin{cases} z^2/2, & |z| \leq \beta, \\ \beta^2/2, & |z| > \beta, \end{cases}, \qquad \beta = 0.5 \times 10^{-4}$$

- For each case, we performed 100 simulations replacing respectively 2%, 5% and 10% of the entries of the vector b by values from the Gaussian distribution with mean 0 and standard deviation $1.2, 1.5, 2.0, 2.5, 5, 10$, respectively.
- For each value of the standard deviation and for each level of changed observations, 100 simulations with Toeplitz matrices of size 100×30 are considered.

The values reported in the table are the percentage of times that the relative error was larger than 10^{-3}:

$$\frac{\|x - x_c\|_2}{\|x\|_2} > 10^{-3},$$

with x_c the solution computed by solving the robust regression problem.

σ	1.2	1.5	2.0	2.5	5.0	10.0
2%	1	0	1	2	1	2
5%	10	22	9	13	11	16
10%	42	44	41	40	49	43

Table: Percentage of problems for which the relative error is greater than 10^{-3} on 100 simulations with different percentages of outliers (rows) and different values of the standard deviation (columns).

Example 2

- To show the computational efficiency of our algorithm, we used `Matlab 5.3` to compare the number of operations with that of a standard algorithm.

Example 2

- To show the computational efficiency of our algorithm, we used `Matlab 5.3` to compare the number of operations with that of a standard algorithm.
- Let $T \in \mathbb{R}^{1000\times 400}$ be a Toeplitz matrix generated as in the previous example. We computed 16 symmetric positive definite matrices

$$T^T(I - S_p^2)T + \mu I, \quad p = 0, 1, \ldots, 15, \tag{9}$$

where S_p is a diagonal matrix with only $p-1$ entries different from zero, corresponding to the indices of the outliers.

Example 2

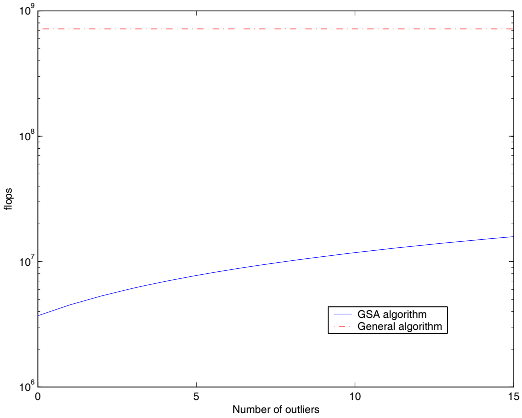
- To show the computational efficiency of our algorithm, we used `Matlab 5.3` to compare the number of operations with that of a standard algorithm.
- Let $T \in \mathbb{R}^{1000\times 400}$ be a Toeplitz matrix generated as in the previous example. We computed 16 symmetric positive definite matrices

$$T^T(I - S_p^2)T + \mu I, \quad p = 0, 1, \ldots, 15, \tag{9}$$

  where S_p is a diagonal matrix with only $p - 1$ entries different from zero, corresponding to the indices of the outliers.
- Regularizing factor: $\mu = 10^{-4}$.

- We compare the work of computing the Cholesky factorization of the Toeplitz matrices using the generalized Schur algorithm and the following one, which does not exploit the Toeplitz structure:
  1. Compute $\hat{T} = \sqrt{I - S_p^2}T.\ (O(pn)\text{operations})$.
  2. Compute R_M, the R factor of the QR–factorization of $\hat{T}$, via the `Matlab` function `qr`. $\left(O(mn^2)\text{operations}\right)$.
  3. Compute the matrix $\hat{A} = R_M^T R_M + \mu I.\ \left(O(n^3)\text{operations}\right)$.
  4. Compute the Cholesky factorization of A via the `Matlab` function `chol`. $\left(O(n^3)\text{operations}\right)$.

  Since $m \gg n$, the most expensive step is the second one. At the same order of complexity, we could also compute the Cholesky factor of by first computing the product $T^T(I - S_p^2)T$.



Computation of the Cholesky factorization of $T^T(I - S_p^2)T + \mu I$.

Conclusions

We have shown how to efficiently compute the solution to robust regression problems when the data matrix has low displacement rank, and have also explained how regularization can be included in the problem formulation.
Thus, the possibility of having outliers in the data is no longer an impediment to using fast methods for structured problems.

N. MASTRONARDI, D.P. O'LEARY,
Fast Robust Regression Algorithms for Problems with Toeplitz Structure,
Computational Statistics & Data Analysis,
accepted for publication.