Numerical Linear Algebra

Least squares problems

Gerard Sleijpen and Martin van Gijzen

9 februari 2012

TU Delft
Delft University of Technology

Program Day 9

- Least squares problems
- The SVD
- Regularisation
- CG for the normal equations
- LSQR and Bi-diagonalization

Least squares problems

In this lesson we consider the problem

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

with $\mathbf{A} \in \mathbb{C}^{n \times m}$, $\mathbf{x} \in \mathbb{C}^{m}$, and $\mathbf{b} \in \mathbb{C}^{n}$.
Furthermore,

- The system may be inconsistent $(\mathbf{b} \notin \mathcal{R}(\mathbf{A}))$.
- Usually $m \ll n$.
- The rank of $\mathbf{A}$ may be smaller than m.

Least squares problems (2)

The system $\mathbf{A}\mathbf{x} = \mathbf{b}$ may be inconsistent. We therefore solve it in the sense of **least squares**, meaning that we solve the minimisation problem $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$, i.e.,

$$\mathbf{x}_{LS} = \text{argmin}_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$$

Solutions $\mathbf{x}_{LS}$ to this problem satisfy the **normal equations**

$$\mathbf{A}^H \mathbf{A}\mathbf{x}_{LS} = \mathbf{A}^H \mathbf{b}$$

and hence

$$\mathbf{r}_{LS} = \mathbf{b} - \mathbf{A}\mathbf{x}_{LS} \perp \mathcal{R}(\mathbf{A})$$

If $\text{rank}(\mathbf{A}) < m$ the least squares solution is not unique.

Least squares problems (3)

Suppose rank$(\mathbf{A}) < m$ and $\mathbf{x}_{LS}$ is a least-squares solution. Then

$$\widehat{\mathbf{x}} = \mathbf{x}_{LS} + \mathbf{y} \quad \text{with} \quad \mathbf{y} \in \mathcal{N}(\mathbf{A})$$

is also a least squares solution.

The least square solution with minimum norm is unique. This **Least Square Minimal Norm** solution $\mathbf{x}_{LSMN}$ solves the constrained problem

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \quad \text{subject to} \quad \mathbf{x} \perp \mathcal{N}(\mathbf{A}) \ .$$

The Singular Value Decomposition

Let $\mathbf{A} \in \mathbb{C}^{n\times m}$ be a matrix of rank r. There exist unitary matrices $\mathbf{U} \in \mathbb{C}^{n\times n}$ and $\mathbf{V} \in \mathbb{C}^{m\times m}$ such that

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^H, \qquad \Sigma = \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix}$$

where $\Sigma \in \mathbb{R}^{n\times m}$ and $\Sigma_r = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_r),$ and

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0.$$

The σ_i are called the **singular values** of $\mathbf{A}$.
$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^H$ is the **SVD** of $\mathbf{A}$.

The SVD and the LSMN solution

The least-squares minimum norm solution can be computed using the SVD by

$$\mathbf{x}_{LSMN} = \mathbf{V} \begin{pmatrix} \Sigma_r^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \mathbf{U}^H \mathbf{b}$$

The matrix

$$\mathbf{A}^{\dagger} \equiv \mathbf{V} \begin{pmatrix} \Sigma_r^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \mathbf{U}^H$$

is called the **Moore-Penrose pseudoinverse** of $\mathbf{A}$.

The SVD and the LSMN solution (2)

We prove that $\mathbf{x}_{LSMN} = \mathbf{A}^{\dagger}\mathbf{b}$.

$$\mathbf{z} = \mathbf{V}^H\mathbf{x} = \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{pmatrix} \quad \mathbf{c} = \mathbf{U}^H\mathbf{b} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix},$$

where $\mathbf{z}_1, \mathbf{c}_1 \in \mathbb{C}^r$. Then,

$$\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 = \|\mathbf{U}^H(\mathbf{b} - \mathbf{A}\mathbf{V}\mathbf{V}^H\mathbf{x})\|_2 =$$

$$= \left\| \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix} - \begin{pmatrix} \Sigma_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} \mathbf{c}_1 - \Sigma_r\mathbf{z}_1 \\ \mathbf{c}_2 \end{pmatrix} \right\|_2.$$

Hence, $\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2$ is minimized by $\mathbf{z}_1 = \Sigma_r^{-1}\mathbf{c}_1$ and $\|\mathbf{x}\|$ by $\mathbf{z}_2 = \mathbf{0}$.

Noisy problems

In least-squares problems $\mathbf{b}$ often corresponds to measured date, which means that we are actually solving the **noisy problem**

$$\mathbf{A}\mathbf{x} = \mathbf{b} + \delta b \; .$$

Moreover, small singular values typically correspond to the noise.

These small singular values have a dramatic effect on the LSMN-solution (why?)!!!

This is an example of a so-called **ill-posed problem**: small perturbations in the data give a large perturbation in the solution.

Regularization

Limiting this effect is called **regularization.**

Several regularization methods have been proposed:

• Set small singular values to 0. This requires the explicit calculation of the SVD, which is not possible for large scale problems.

• **Tykhonov regularisation.** Solve the damped least squares problem:

$$\min_{\mathbf{x}} \left\| \begin{pmatrix} \mathbf{A} \\ \tau \mathbf{I} \end{pmatrix} \mathbf{x} - \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \right\|_2$$

• Use an iterative method (reason: convergence to small singular values is slow)

CGLS (1)

CG can always be applied to the normal equations

$$\mathbf{A}^H \mathbf{A}\mathbf{x} = \mathbf{A}^H \mathbf{b}$$

since $\mathbf{A}^H \mathbf{A}$ is Hermitian positive semi-definite.

The stability can be improved by replacing inner products

$$\mathbf{u}^H (\mathbf{A}^H \mathbf{A}\mathbf{u})$$

by inner products

$$(\mathbf{A}\mathbf{u})^H \mathbf{A}\mathbf{u}$$

which leads to the algorithm CGLS.

CGLS (2)

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0,\ \mathbf{s}_0 = \mathbf{A}^H \mathbf{r}_0,\ \mathbf{u}_0 = \mathbf{s}_0,\ \rho_0 = \mathbf{s}_k^H \mathbf{s}_k$ *Initialization*

FOR $k = 0, 1, \cdots,$ DO

$\mathbf{c}_k = \mathbf{A}\mathbf{u}_k$

$\sigma_k = \mathbf{c}_k^H \mathbf{c}_k, \quad \alpha_k = \frac{\rho_k}{\sigma_k}$

$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{u}_k$ *update iterate*

$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{c}_k$ *update residual*

$\mathbf{s}_{k+1} = \mathbf{A}^H \mathbf{r}_{k+1}$ *residual normal equations*

$\rho_{k+1} = \mathbf{s}_{k+1}^H \mathbf{s}_{k+1}, \quad \beta_k = \frac{\rho_{k+1}}{\rho_k}$

$\mathbf{u}_{k+1} = \mathbf{s}_{k+1} + \beta_k \mathbf{u}_k$ *update direction vector*

END FOR

CGLS (3)

CGLS can also be used for solving nonsymmetric square systems. However, this has two important disadvantages:

- The work per iteration is twice as much as in CG;
- $K_2(\mathbf{A}^H \mathbf{A}) = K_2(\mathbf{A})^2$, which means that convergence is often very slow.

CGLS (4), Assignment

Assuming that $K_2(\mathbf{A}) = 100$ and $\mathbf{A}$ is Hermitian:

1. Give an upper bound on the number of CG iterations

   required to satisfy $\frac{\|\mathbf{x}-\mathbf{x}_k\|_A}{\|\mathbf{x}-\mathbf{x}_0\|_A} < 10^{-6}$.

   Hint: use the upper bound

$$\|\mathbf{x} - \mathbf{x}_k\|_A \leq 2 \left(\frac{\sqrt{K_2(\mathbf{A})} - 1}{\sqrt{K_2(\mathbf{A})} + 1} \right)^k \|\mathbf{x} - \mathbf{x}_0\|_A .$$

2. Answer the same question for CGLS.

CGLS (4), Assignment

Assuming that $K_2(\mathbf{A}) = 100$ and $\mathbf{A}$ is Hermitian:

1. Give an upper bound on the number of CG iterations

   required to satisfy $\frac{\|\mathbf{x}-\mathbf{x}_k\|_A}{\|\mathbf{x}-\mathbf{x}_0\|_A} < 10^{-6}$.

   Hint: use the upper bound

$$\|\mathbf{x} - \mathbf{x}_k\|_A \leq 2\left(\frac{\sqrt{K_2(\mathbf{A})} - 1}{\sqrt{K_2(\mathbf{A})} + 1}\right)^k \|\mathbf{x} - \mathbf{x}_0\|_A.$$

2. Answer the same question for CGLS.

Answer: CG: 73, CGLS 726

LSQR

LSQR (Paige and Saunders) is derived by applying Lanczos to

$$\begin{pmatrix} \mathbf{I} & \mathbf{A} \\ \mathbf{A}^H & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{r} \\ \mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} .$$

with starting vector $\mathbf{u}_1 = \frac{1}{\|\mathbf{b}\|} \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix}$

LSQR (2)

The second vector in the Krylov subspace becomes

$$\frac{1}{\|\mathbf{b}\|}\begin{pmatrix} \mathbf{b} \\ \mathbf{A}^H\mathbf{b} \end{pmatrix}$$

After orthonormalisation we obtain

$$\frac{1}{\|\mathbf{A}^H\mathbf{b}\|}\begin{pmatrix} \mathbf{0} \\ \mathbf{A}^H\mathbf{b} \end{pmatrix}$$

Repeating this procedure shows that we get alternatingly

orthogonal vectors $\begin{pmatrix} \mathbf{u} \\ \mathbf{0} \end{pmatrix}$ and $\begin{pmatrix} \mathbf{0} \\ \mathbf{v} \end{pmatrix}$

LSQR (3)

This observation leads to the following

Bidiagonalisation algorithm (Golub and Kahan)

$\beta_1 \mathbf{u}_1 = \mathbf{b}, \quad \alpha_1 \mathbf{v}_1 = \mathbf{A}^H \mathbf{u}_1$

FOR $\quad i = 2, 3, \ldots \quad$ DO

$$\beta_i \mathbf{u}_i = \mathbf{A}\mathbf{v}_{i-1} - \alpha_{i-1}\mathbf{u}_{i-1}$$

$$\alpha_i \mathbf{v}_i = \mathbf{A}^H \mathbf{u}_i - \beta_i \mathbf{v}_{i-1}$$

END FOR

with $\alpha_i > 0$ and $\beta_i > 0$ such that $\|\mathbf{u}_i\| = \|\mathbf{v}_i\| = 1$.

See also Exercises 5.8 and 6.8

LSQR (4)

With $\quad \mathbf{U}_k \equiv [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k], \quad \mathbf{V}_k \equiv [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k]$ and

$$\underline{B}_k \equiv \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_k \end{bmatrix},$$

we have

$$\begin{aligned} \beta_1 \mathbf{U}_{k+1} e_1 &= \mathbf{b} \\ \mathbf{A}\mathbf{V}_k &= \mathbf{U}_{k+1} \underline{B}_k \\ \mathbf{A}^H \mathbf{U}_{k+1} &= \mathbf{V}_k (\underline{B}_k)^H + \alpha_{k+1} \mathbf{v}_{k+1} e_{k+1}^T = \mathbf{V}_{k+1} B_{k+1}^H. \end{aligned}$$

LSQR (5)

Now construct solution vectors $\mathbf{x}_k = \mathbf{V}_k \, y_k$.

Then we get for $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$:

$$\begin{aligned} \mathbf{r}_k &= \beta_1 \mathbf{U}_{k+1} \, e_1 - \mathbf{A}\mathbf{V}_k \, y_k \\ &= \beta_1 \mathbf{U}_{k+1} e_1 - \mathbf{U}_{k+1} \, \underline{B}_k \, y_k \\ &= \mathbf{U}_{k+1} (\beta_1 e_1 - \underline{B}_k \, y_k) \\ &= \mathbf{U}_{k+1} \, t_k \end{aligned}$$

LSQR (6)

Substitution in the augmented system and using the Gallerkin condition gives

$$\begin{pmatrix} \mathbf{U}_{k+1}^H & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_k^H \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{A} \\ \mathbf{A}^H & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U}_{k+1}\, t_{k+1} \\ \mathbf{V}_k\, y_k \end{pmatrix} = \begin{pmatrix} \mathbf{U}_{k+1}^H \mathbf{b} \\ \mathbf{0} \end{pmatrix},$$

which leads to the lower dimensional system

$$\begin{pmatrix} I & \underline{B}_k \\ (\underline{B}_k)^H & 0 \end{pmatrix} \begin{pmatrix} t_{k+1} \\ y_k \end{pmatrix} = \begin{pmatrix} \beta_1 e_1 \\ 0 \end{pmatrix}.$$

LSQR (7)

This last equation is equivalent to the least squares problem

$$y_k = \text{argmin}_y \|\beta_1 e_1 - \underline{B}_k y\|_2.$$

In LSQR this problem is solved using the QR-decomposition

$$\mathbf{x}_k = (\mathbf{V}_k R_k^{-1})(\underline{Q}_k^H(\beta_1 e_1)), \quad \text{where} \quad \underline{B}_k = \underline{Q}_k R_k.$$

The QR-decomposition is based on Givens rotations.

LSQR is famous for its robustness.

Final remarks

Today we have seen CG-type methods for the normal equations.

These methods can also be applied to nonsymmetric systems.

The disadvantage of this approach is that the condition number may be squared compared to the original system. This may lead to slow convergence and/or an inaccurate solution.

However, there are also classes of problems for which the normal equations approach works quite well, in particular if $\mathbf{A}$ is close to an orthogonal matrix.