Methods for solving Linear Least Squares problems

Anibal Sosa

IPM for Linear Programming,
September 2009

Outline

1. The Least Square Problem (LSQ)
   - Linear Least Square Problems
2. Methods for solving Linear LSQ
   - Normal Equations
   - QR Factorization
   - Singular Value Decomposition (SVD)
3. Comments on the three methods
4. Regularization techniques
   - Tikhonov regularization and Damped SVD
   - Tikhonov regularization order one and two

Outline

1. The Least Square Problem (LSQ)
   - Linear Least Square Problems
2. Methods for solving Linear LSQ
   - Normal Equations
   - QR Factorization
   - Singular Value Decomposition (SVD)
3. Comments on the three methods
4. Regularization techniques
   - Tikhonov regularization and Damped SVD
   - Tikhonov regularization order one and two

Outline

1. The Least Square Problem (LSQ)
   - Linear Least Square Problems
2. Methods for solving Linear LSQ
   - Normal Equations
   - QR Factorization
   - Singular Value Decomposition (SVD)
3. Comments on the three methods
4. Regularization techniques
   - Tikhonov regularization and Damped SVD
   - Tikhonov regularization order one and two

Outline

Computational Science

The Least Square Problem (LSQ)

- The objective function has the following special form

  $f(x) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x)$, where $r_j : \mathbb{R}^n \to \mathbb{R}$ are the *residuals* , i. e.,

$$\min_{x\in\mathbb{R}^n} f(x) = \min_{x\in\mathbb{R}^n} \frac{1}{2} r^T(x) r(x) = \min_{x\in\mathbb{R}^n} \frac{1}{2} ||r(x)||_2^2$$

  $r : \mathbb{R}^n \to \mathbb{R}^m$ is called the residual vector, i.e., $r = \begin{bmatrix} r_1(x) \\ r_2(x) \\ \vdots \\ r_m(x) \end{bmatrix}$

- Least square problems arise in many areas of applications
- Largest source of unconstrained optimization problems

The Least Square Problem (LSQ)

- The objective function has the following special form $f(x) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x)$, where $r_j : \mathbb{R}^n \rightarrow \mathbb{R}$ are the *residuals* , i. e.,

$$\min_{x\in\mathbb{R}^n} f(x) = \min_{x\in\mathbb{R}^n} \frac{1}{2} r^T(x) r(x) = \min_{x\in\mathbb{R}^n} \frac{1}{2} ||r(x)||_2^2$$

$r : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is called the residual vector, i.e., $r = \begin{bmatrix} r_1(x) \\ r_2(x) \\ \vdots \\ r_m(x) \end{bmatrix}$

- Least square problems arise in many areas of applications
- Largest source of unconstrained optimization problems

The Least Square Problem (LSQ)

- The objective function has the following special form

  $f(x) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x)$, where $r_j : \mathbb{R}^n \to \mathbb{R}$ are the *residuals* , i. e.,

$$\min_{x\in\mathbb{R}^n} f(x) = \min_{x\in\mathbb{R}^n} \frac{1}{2} r^T(x) r(x) = \min_{x\in\mathbb{R}^n} \frac{1}{2} ||r(x)||_2^2$$

  $r : \mathbb{R}^n \to \mathbb{R}^m$ is called the residual vector, i.e., $r = \begin{bmatrix} r_1(x) \\ r_2(x) \\ \vdots \\ r_m(x) \end{bmatrix}$

- Least square problems arise in many areas of applications

- Largest source of unconstrained optimization problems

Linear Least Square Problems

- Let $\phi(x;\rho)$ be a model function that predict experimental values, for some fix parameters ρ.
- Usually we want to minimize the differences between the observed values $y \in \mathbb{R}^m$(data) and the predicted values $\phi(x;\rho) \in \mathbb{R}^m$.
- We can use LSQ setting $r(x) = \phi(x;\rho) - y$

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}||\phi(x;\rho) - y||_2^2 \quad (1)$$

- If ϕ in (2) is nonlinear then we have a nonlinear LSQ problem
- In our case $\phi(x) = Ax$, thus we say this is a linear LSQ problem

Linear Least Square Problems

- Let $\phi(x;\rho)$ be a model function that predict experimental values, for some fix parameters ρ.
- Usually we want to minimize the differences between the observed values $y \in \mathbb{R}^m$(data) and the predicted values $\phi(x;\rho) \in \mathbb{R}^m$.
- We can use LSQ setting $r(x) = \phi(x;\rho) - y$

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} ||\phi(x;\rho) - y||_2^2 \tag{1}$$

- If ϕ in (2) is nonlinear then we have a nonlinear LSQ problem
- In our case $\phi(x) = Ax$, thus we say this is a linear LSQ problem

Linear Least Square Problems

- Let $\phi(x;\rho)$ be a model function that predict experimental values, for some fix parameters ρ.
- Usually we want to minimize the differences between the observed values $y \in \mathbb{R}^m$(data) and the predicted values $\phi(x;\rho) \in \mathbb{R}^m$.
- We can use LSQ setting $r(x) = \phi(x;\rho) - y$

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} ||\phi(x;\rho) - y||_2^2 \qquad (1)$$

- If ϕ in (2) is nonlinear then we have a nonlinear LSQ problem
- In our case $\phi(x) = Ax$, thus we say this is a linear LSQ problem

Linear Least Square Problems

- Let $\phi(x;\rho)$ be a model function that predict experimental values, for some fix parameters ρ.
- Usually we want to minimize the differences between the observed values $y \in \mathbb{R}^m$(data) and the predicted values $\phi(x;\rho) \in \mathbb{R}^m$.
- We can use LSQ setting $r(x) = \phi(x;\rho) - y$

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} ||\phi(x;\rho) - y||_2^2 \tag{1}$$

- If ϕ in (2) is nonlinear then we have a nonlinear LSQ problem
- In our case $\phi(x) = Ax$, thus we say this is a linear LSQ problem

Preliminaries for solving the LSQ problem

Observe that

$$f(x) = \frac{1}{2}||Ax-y||_2^2 = \frac{1}{2}(Ax-y)^T(Ax-y) = \frac{1}{2}x^TA^TAx - x^TA^Ty + \frac{1}{2}y^Ty$$

is easy to prove that

$$\nabla f(x) = A^T(Ax-y) \quad \nabla^2 f(x) = A^TA$$

Since f is a convex function is well known that any x^* such that $\nabla f(x^*) = 0$ is a global minimizer of f, therefore x^* satisfy the normal equations

$$A^TAx = A^Ty$$

Next we discuss three major algorithms for solving Linear LSQ problems, assuming: i) $m \geq n$ and ii) A is full rank

Preliminaries for solving the LSQ problem

Observe that

$$f(x) = \frac{1}{2}||Ax-y||_2^2 = \frac{1}{2}(Ax-y)^T(Ax-y) = \frac{1}{2}x^TA^TAx - x^TA^Ty + \frac{1}{2}y^Ty$$

is easy to prove that

$$\nabla f(x) = A^T(Ax-y) \quad \nabla^2 f(x) = A^TA$$

Since f is a convex function is well known that any x^* such that $\nabla f(x^*) = 0$ is a global minimizer of f, therefore x^* satisfy the normal equations

$$A^TAx = A^Ty$$

Next we discuss three major algorithms for solving Linear LSQ problems, assuming: i) $m \geq n$ and ii) A is full rank

Preliminaries for solving the LSQ problem

Observe that

$$f(x) = \frac{1}{2}||Ax-y||_2^2 = \frac{1}{2}(Ax-y)^T(Ax-y) = \frac{1}{2}x^TA^TAx - x^TA^Ty + \frac{1}{2}y^Ty$$

is easy to prove that

$$\nabla f(x) = A^T(Ax-y) \quad \nabla^2 f(x) = A^TA$$

Since f is a convex function is well known that any x^* such that $\nabla f(x^*) = 0$ is a global minimizer of f, therefore x^* satisfy the normal equations

$$A^TAx = A^Ty$$

Next we discuss three major algorithms for solving Linear LSQ problems, assuming: i) $m \geq n$ and ii) A is full rank

Normal Equations

Step1 Compute A^TA and A^Ty

Step2 Compute Cholesky factorization of $A^TA > 0$

$$A^TA = R^TR, \quad R \text{ is an upper triangular matrix}(R_{ii} > 0)$$

Step3 Perform two triangular substitutions

$$R^Tz = R^Ty \Longrightarrow Rx^* = z$$

Disadvantages:

- Relative error of $x^* \approx \kappa(A)^2$[1]
- Sensitive to ill-conditioned matrices

[1] $\kappa(A) = ||A||\,||A^{-1}|| \approx \frac{\sigma_1}{\sigma_n} = \kappa_2(A)$

Normal Equations

Step1 Compute $A^T A$ and $A^T y$

Step2 Compute Cholesky factorization of $A^T A > 0$

$$A^T A = R^T R, \quad R \text{ is an upper triangular matrix} (R_{ii} > 0)$$

Step3 Perform two triangular substitutions

$$R^T z = R^T y \Longrightarrow Rx^* = z$$

Disadvantages:

- Relative error of $x^* \approx \kappa(A)^2$[1]
- Sensitive to ill-conditioned matrices

[1] $\kappa(A) = ||A||\,||A^{-1}|| \approx \frac{\sigma_1}{\sigma_n} = \kappa_2(A)$

Normal Equations

Step1 Compute $A^T A$ and $A^T y$

Step2 Compute Cholesky factorization of $A^T A > 0$

$$A^T A = R^T R, \quad R \text{ is an upper triangular matrix}(R_{ii} > 0)$$

Step3 Perform two triangular substitutions

$$R^T z = R^T y \Longrightarrow Rx^* = z$$

Disadvantages:

- Relative error of $x^* \approx \kappa(A)^2$[1]
- Sensitive to ill-conditioned matrices

[1] $\kappa(A) = ||A|| \, ||A^{-1}|| \approx \frac{\sigma_1}{\sigma_n} = \kappa_2(A)$

QR Factorization

Notice that $||\cdot||$ is invariant under orthogonal transformations

$$||Ax - y||_2^2 = ||Q^T(Ax - y)||_2^2$$

where $Q_{m\times m}$ is orthogonal

- The QR factorization is done as follows

$$A\Pi = Q\begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1 \quad Q_2]\begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R \qquad (2)$$

  where $\Pi_{n\times n}$ is a permutation matrix, Q_1 is the first n columns of Q and $R_{n\times n}$ is upper triangular with $R_{ii} > 0$

- Using 2 we have

$$||Ax - y||_2^2 = \left|\begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}(A\Pi\Pi^T x - y)\right|_2^2$$

Computational Science

QR Factorization

Notice that $||\cdot||$ is invariant under orthogonal transformations

$$||Ax-y||_2^2 = ||Q^T(Ax-y)||_2^2$$

where $Q_{m\times m}$ is orthogonal

- The QR factorization is done as follows

$$A\Pi = Q\begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1 \quad Q_2]\begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R \tag{2}$$

  where $\Pi_{n\times n}$ is a permutation matrix, Q_1 is the first n columns of Q and $R_{n\times n}$ is upper triangular with $R_{ii} > 0$

- Using 2 we have

$$||Ax-y||_2^2 = \left\| \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} (A\Pi\Pi^T x - y) \right\|_2^2$$

QR Factorization

Notice that $||\cdot||$ is invariant under orthogonal transformations

$$||Ax - y||_2^2 = ||Q^T(Ax - y)||_2^2$$

where $Q_{m\times m}$ is orthogonal

- The QR factorization is done as follows

$$A\Pi = Q\begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1 \quad Q_2]\begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R \tag{2}$$

  where $\Pi_{n\times n}$ is a permutation matrix, Q_1 is the first n columns of Q and $R_{n\times n}$ is upper triangular with $R_{ii} > 0$

- Using 2 we have

$$||Ax - y||_2^2 = \left|\left| \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} (A\Pi\Pi^T x - y) \right|\right|_2^2$$

QR Factorization(2)

$$\left\| \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} \left(\underbrace{[Q_1 \quad Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix}}_{A\Pi} \Pi^T x - y \right) \right\|_2^2 = \left\| \begin{bmatrix} R \\ 0 \end{bmatrix} \Pi^T x - \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} y \right\|_2^2$$

$$= ||R\Pi^T x - Q_1^T y||_2^2 + ||Q_2^T y||_2^2$$

Notice that from the last equation:

- The last term does not depend on x
- The minimum value is reached when $R\Pi^T x - Q_1^T y = 0$, therefore

$$x^* = \Pi R^{-1} Q_1^T y$$

QR Factorization(2)

$$\left\| \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} \left(\underbrace{[Q_1 \quad Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix}}_{A\Pi} \Pi^T x - y \right) \right\|_2^2 = \left\| \begin{bmatrix} R \\ 0 \end{bmatrix} \Pi^T x - \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} y \right\|_2^2$$

$$= ||R\Pi^T x - Q_1^T y||_2^2 + ||Q_2^T y||_2^2$$

Notice that from the last equation:

- The last term does not depend on x
- The minimum value is reached when $R\Pi^T x - Q_1^T y = 0$, therefore

$$x^* = \Pi R^{-1} Q_1^T y$$

QR Factorization Algorithm

Step1 Compute QR factorization of A

Step2 Extract Q_1, identify Π and R

Step3 Perform one triangular substitution and one permutation

$$Rz = Q_1^T y \Longrightarrow x^* = \Pi z$$

Advantage:

- Relative error of $x^* \approx \kappa(A)$

Disadvantage:

- Sometimes is necessary more information about data sensitivity

QR Factorization Algorithm

Step1 Compute QR factorization of A

Step2 Extract Q_1, identify Π and R

Step3 Perform one triangular substitution and one permutation

$$Rz = Q_1^T y \Longrightarrow x^* = \Pi z$$

Advantage:

- Relative error of $x^* \approx \kappa(A)$

Disadvantage:

- Sometimes is necessary more information about data sensitivity

QR Factorization Algorithm

Step1 Compute QR factorization of A

Step2 Extract Q_1, identify Π and R

Step3 Perform one triangular substitution and one permutation

$$Rz = Q_1^T y \Longrightarrow x^* = \Pi z$$

Advantage:

- Relative error of $x^* \approx \kappa(A)$

Disadvantage:

- Sometimes is necessary more information about data sensitivity

Singular Value Decomposition (SVD)

Theorem

If $A_{m\times n}$ is real then there exist orthogonal matrices

$$U = [u_1 \dots u_m] \in \mathbb{R}^{m\times m} \text{ and } V = [v_1 \dots v_n] \in \mathbb{R}^{n\times n}$$

such that $A = U\Sigma V^T$, where
$\Sigma = diag(\sigma_1, \dots, \sigma_p) \in \mathbb{R}^{m\times n}$, $p = \min\{m, n\}$ *and* $\sigma_1 \geq \sigma_2 \dots \geq \sigma_p \geq 0$

- In our case $\sigma_1 \geq \sigma_2 \dots \geq \sigma_n > 0$ since A is full rank and $m \gg n$ thus

$$A = U \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} V^T = [U_1 \quad U_2] \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} V^T = U_1\Sigma_1V^T \qquad (3)$$

where U_1 has the first n columns of U and $\Sigma_1 = diag(\sigma_1, \dots, \sigma_n)$.

Computational Science

Singular Value Decomposition (SVD)

Theorem

If $A_{m\times n}$ is real then there exist orthogonal matrices

$$U = [u_1 \dots u_m] \in \mathbb{R}^{m\times m} \textit{ and } V = [v_1 \dots v_n] \in \mathbb{R}^{n\times n}$$

such that $A = U\Sigma V^T$, where
$\Sigma = diag(\sigma_1, \dots, \sigma_p) \in \mathbb{R}^{m\times n}$, $p = \min\{m, n\}$ *and* $\sigma_1 \geq \sigma_2 \dots \geq \sigma_p \geq 0$

- In our case $\sigma_1 \geq \sigma_2 \dots \geq \sigma_n > 0$ since A is full rank and $m \gg n$ thus

$$A = U \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} V^T = [U_1 \quad U_2] \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} V^T = U_1 \Sigma_1 V^T \qquad (3)$$

where U_1 has the first n columns of U and $\Sigma_1 = diag(\sigma_1, \dots, \sigma_n)$.

The thin SVD

- Using (3) and similar ideas from QR

$$||Ax - y||_2^2 = \left|\left| \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} \left(V^T x\right) - \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} y \right|\right|_2^2$$
$$= ||\Sigma_1 \left(V^T x\right) - U_1^T y||_2^2 + ||U_2^T y||_2^2$$

Again from the last equation:

- The last term does not depend on x
- The minimum value is reached when $\Sigma \left(V^T x\right) - U_1^T y = 0$, therefore

$$x^* = V\Sigma^{-1} U_1^T y$$

or equivalently

$$x^* = \sum_{i=1}^{n} \left(\frac{u_i^T y}{\sigma_i}\right) v_i \tag{4}$$

The thin SVD

- Using (3) and similar ideas from QR

$$\begin{aligned} ||Ax - y||_2^2 &= \left|\left| \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} \left(V^T x\right) - \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} y \right|\right|_2^2 \\ &= ||\Sigma_1 \left(V^T x\right) - U_1^T y||_2^2 + ||U_2^T y||_2^2 \end{aligned}$$

Again from the last equation:

- The last term does not depend on x
- The minimum value is reached when $\Sigma \left(V^T x\right) - U_1^T y = 0$, therefore

$$x^* = V\Sigma^{-1} U_1^T y$$

or equivalently

$$x^* = \sum_{i=1}^{n} \left(\frac{u_i^T y}{\sigma_i} \right) v_i \qquad (4)$$

Computational Science

SVD

- Equation (4) gives useful information about x^* sensitivity
  - Small changes in A or y can induce large changes in x^* if σ_i is small
  - A is rank defficient when $\frac{\sigma_n}{\sigma_1} \ll 1$. ($\sigma_n$ is the distance from A to the set of singular matrices)
- x^*calculated as in (4) has the smallest 2-norm of all minimizers

Advantage:

- Most robust and reliable

Disadvantage:

- Most expensive

SVD

- Equation (4) gives useful information about x^* sensitivity
  - Small changes in A or y can induce large changes in x^* if σ_i is small
  - A is rank defficient when $\frac{\sigma_n}{\sigma_1} \ll 1$. ($\sigma_n$ is the distance from A to the set of singular matrices)
- x^*calculated as in (4) has the smallest 2-norm of all minimizers

Advantage:

- Most robust and reliable

Disadvantage:

- Most expensive

Normal Eq. vs QR vs SVD

- The Cholesky-based algorithm is practical if $m \gg n$ (is easier store $A^T A$), even if A is sparse
- The QR algorithm avoid squaring $\kappa(A)$
- When A is rank-deficient, some $\sigma_i \approx 0$ thus any vector

$$x^* = \sum_{\sigma_i \neq 0} \left(\frac{u_i^T y}{\sigma_i} \right) v_i + \sum_{\sigma_i = 0} \tau v_i$$

is also a minimizer of $||Ax - y||$, for τ such that $\sigma_i \geq \tau$,. Thus setting $\tau_i = 0$ we get the minimum norm solution[2]

Remark: For very large problems is recommended to use iterative methods as *Conjugate Gradient*

[2] This is a type of filter by doing truncation

Normal Eq. vs QR vs SVD

- The Cholesky-based algorithm is practical if $m \gg n$ (is easier store $A^T A$), even if A is sparse
- The QR algorithm avoid squaring $\kappa(A)$
- When A is rank-deficient, some $\sigma_i \approx 0$ thus any vector

$$x^* = \sum_{\sigma_i \neq 0} \left(\frac{u_i^T y}{\sigma_i} \right) v_i + \sum_{\sigma_i = 0} \tau v_i$$

is also a minimizer of $||Ax - y||$, for τ such that $\sigma_i \geq \tau$,. Thus setting $\tau_i = 0$ we get the minimum norm solution[2]

Remark: For very large problems is recommended to use iterative methods as *Conjugate Gradient*

[2]This is a type of filter by doing truncation

Normal Eq. vs QR vs SVD

- The Cholesky-based algorithm is practical if $m \gg n$ (is easier store $A^T A$), even if A is sparse
- The QR algorithm avoid squaring $\kappa(A)$
- When A is rank-deficient, some $\sigma_i \approx 0$ thus any vector

$$x^* = \sum_{\sigma_i \neq 0} \left(\frac{u_i^T y}{\sigma_i} \right) v_i + \sum_{\sigma_i = 0} \tau v_i$$

is also a minimizer of $||Ax - y||$, for τ such that $\sigma_i \geq \tau$,. Thus setting $\tau_i = 0$ we get the minimum norm solution[2]

Remark: For very large problems is recommended to use iterative methods as *Conjugate Gradient*

[2] This is a type of filter by doing truncation

Normal Eq. vs QR vs SVD

- The Cholesky-based algorithm is practical if $m \gg n$ (is easier store $A^T A$), even if A is sparse
- The QR algorithm avoid squaring $\kappa(A)$
- When A is rank-deficient, some $\sigma_i \approx 0$ thus any vector

$$x^* = \sum_{\sigma_i \neq 0} \left(\frac{u_i^T y}{\sigma_i} \right) v_i + \sum_{\sigma_i = 0} \tau v_i$$

is also a minimizer of $||Ax - y||$, for τ such that $\sigma_i \geq \tau$,. Thus setting $\tau_i = 0$ we get the minimum norm solution[2]

Remark: For very large problems is recommended to use iterative methods as *Conjugate Gradient*

[2] This is a type of filter by doing truncation

Tikhonov regularization[a]

[a]Ridge regression

- Most commonly used method for ill-posed problems
- The ill-conditioned problem 1 is posed as

$$\min \frac{1}{2}||Ax - y||_2^2 + \frac{1}{2}\alpha^2||x||_2^2 \tag{5}$$

for some suitable *regularization parameter* $\alpha > 0$

- This improves the problem condition, even is A is rank-deficient, shifting the small singular values

$$\left(A^T A + \alpha I_n\right) x = \underbrace{A^T A x}_{\lambda x} + \alpha x = (\lambda + \alpha) x$$

for any eigenvalue λ and eigenvector x of $A^T A$

Tikhonov regularization[a]

[a] Ridge regression

- Most commonly used method for ill-posed problems
- The ill-conditioned problem 1 is posed as

$$\min \frac{1}{2}||Ax - y||_2^2 + \frac{1}{2}\alpha^2||x||_2^2 \tag{5}$$

for some suitable *regularization parameter* $\alpha > 0$

- This improves the problem condition, even is A is rank-deficient, shifting the small singular values

$$\left(A^T A + \alpha I_n\right) x = \underbrace{A^T A x}_{\lambda x} + \alpha x = (\lambda + \alpha) x$$

for any eigenvalue λ and eigenvector x of $A^T A$

Tikhonov regularization[a]

[a] Ridge regression

- Most commonly used method for ill-posed problems
- The ill-conditioned problem 1 is posed as

$$\min \frac{1}{2}||Ax - y||_2^2 + \frac{1}{2}\alpha^2||x||_2^2 \tag{5}$$

for some suitable *regularization parameter* $\alpha > 0$

- This improves the problem condition, even is A is rank-deficient, shifting the small singular values

$$\left(A^T A + \alpha I_n\right) x = \underbrace{A^T Ax}_{\lambda x} + \alpha x = (\lambda + \alpha)\, x$$

for any eigenvalue λ and eigenvector x of $A^T A$

Tikhonov regularization and Damped SVD

- A little algebra shows that the minimum solution of (5) is given by the nonsingular system

$$\left(A^T A + \alpha^2 I_n\right) x = A^T y$$

and from (4) we can show that

$$x^* = \sum_{i=1}^{n} f_i \left(\frac{u_i^T y}{\sigma_i} \right) v_i$$

where $f_i = \frac{\sigma_i^2}{\sigma_i^2+\alpha^2}$ are known as *filter factors*[3]

- The impact of an small α in the filter factors is:
  - None for large $\sigma_i (\alpha \ll \sigma_i)$,i.e. $\frac{\sigma_i^2}{\sigma_i^2+\alpha^2} \approx 1$
  - Reduce the magnification of $\frac{1}{\sigma_i}$ since $\frac{\sigma_i^2}{\sigma_i^2+\alpha^2} \approx \frac{\sigma_i^2}{\alpha^2} \ll 1$
- A "good" choice of α may provide enough numerical stability to expect a good approximate solution

[3] *In signal processing are known as Wiener filters*

Tikhonov regularization and Damped SVD

- A little algebra shows that the minimum solution of (5) is given by the nonsingular system

$$\left(A^T A + \alpha^2 I_n\right) x = A^T y$$

and from (4) we can show that

$$x^* = \sum_{i=1}^{n} f_i \left(\frac{u_i^T y}{\sigma_i} \right) v_i$$

where $f_i = \frac{\sigma_i^2}{\sigma_i^2+\alpha^2}$ are known as *filter factors*[3]

- The impact of an small α in the filter factors is:
  - None for large $\sigma_i (\alpha \ll \sigma_i)$,i.e. $\frac{\sigma_i^2}{\sigma_i^2+\alpha^2} \approx 1$
  - Reduce the magnification of $\frac{1}{\sigma_i}$ since $\frac{\sigma_i^2}{\sigma_i^2+\alpha^2} \approx \frac{\sigma_i^2}{\alpha^2} \ll 1$
- A "good" choice of α may provide enough numerical stability to expect a good approximate solution

[3] In signal processing are known as Wiener filters

Tikhonov regularization and Damped SVD

- A little algebra shows that the minimum solution of (5) is given by the nonsingular system

$$\left(A^T A + \alpha^2 I_n\right) x = A^T y$$

and from (4) we can show that

$$x^* = \sum_{i=1}^{n} f_i \left(\frac{u_i^T y}{\sigma_i} \right) v_i$$

where $f_i = \frac{\sigma_i^2}{\sigma_i^2 + \alpha^2}$ are known as *filter factors*[3]

- The impact of an small α in the filter factors is:
  - None for large $\sigma_i (\alpha \ll \sigma_i)$,i.e. $\frac{\sigma_i^2}{\sigma_i^2+\alpha^2} \approx 1$
  - Reduce the magnification of $\frac{1}{\sigma_i}$ since $\frac{\sigma_i^2}{\sigma_i^2+\alpha^2} \approx \frac{\sigma_i^2}{\alpha^2} \ll 1$
- A "good" choice of α may provide enough numerical stability to expect a good approximate solution

[3] *In signal processing are known as Wiener filters*

Tikhonov regularization order one

- Damping the large components in magnitude may not inhibit undesirable behavior of the singular values.
- Strong regularization is needed, penalizaing rapid changes of x_i (4)

$$\min \frac{1}{2}||Ax - y||_2^2 + \frac{1}{2}\alpha^2 \sum_{i=2}^{n-1} (x_i - x_{i-1})^2$$

- Again this expression is minimized by the solution of

$$\left(A^T A + \alpha^2 B_1^T B_1\right) x = A^T y$$

where

$$B_1 = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ \vdots & \ddots & 1 & \ddots & \ddots \\ \vdots & \vdots & \vdots & \ddots & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{(n-1)\times n}$$

Tikhonov regularization order one

- Damping the large components in magnitude may not inhibit undesirable behavior of the singular values.
- Strong regularization is needed, penalizaing rapid changes of x_i (4)

$$\min \frac{1}{2}||Ax - y||_2^2 + \frac{1}{2}\alpha^2 \sum_{i=2}^{n-1} (x_i - x_{i-1})^2$$

- Again this expression is minimized by the solution of

$$\left(A^T A + \alpha^2 B_1^T B_1\right) x = A^T y$$

where

$$B_1 = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ \vdots & \ddots & 1 & \ddots & \ddots \\ \vdots & \vdots & \vdots & \ddots & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{(n-1)\times n}$$

Tikhonov regularization order one

- Damping the large components in magnitude may not inhibit undesirable behavior of the singular values.
- Strong regularization is needed, penalizaing rapid changes of x_i (4)

$$\min \frac{1}{2}||Ax - y||_2^2 + \frac{1}{2}\alpha^2 \sum_{i=2}^{n-1} (x_i - x_{i-1})^2$$

- Again this expression is minimized by the solution of

$$\left(A^T A + \alpha^2 B_1^T B_1\right) x = A^T y$$

where

$$B_1 = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ \vdots & \ddots & 1 & \ddots & \ddots \\ \vdots & \vdots & \vdots & \ddots & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{(n-1)\times n}$$

Tikhonov regularization order two

- An even stronger regularization is

$$\min \frac{1}{2}||Ax - y||_2^2 + \frac{1}{2}\alpha^2 \sum_{i=2}^{n-1} (x_{i+1} - 2x_i + x_{i-1})^2$$

- Again this expression is minimized by the solution of

$$\left(A^T A + \alpha^2 B_2^T B_2\right) x = A^T y$$

where

$$B_2 = \begin{bmatrix} -2 & 1 & 0 & 0 & \cdots \\ 1 & -2 & 1 & 0 & \cdots \\ \vdots & 1 & -2 & 1 & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 1 & -2 \end{bmatrix}_{(n-2)\times n}$$

Tikhonov regularization order two

- An even stronger regularization is

$$\min \frac{1}{2}||Ax-y||_2^2 + \frac{1}{2}\alpha^2 \sum_{i=2}^{n-1} (x_{i+1} - 2x_i + x_{i-1})^2$$

- Again this expression is minimized by the solution of

$$\left(A^T A + \alpha^2 B_2^T B_2\right) x = A^T y$$

where

$$B_2 = \begin{bmatrix} -2 & 1 & 0 & 0 & \cdots \\ 1 & -2 & 1 & 0 & \cdots \\ \vdots & 1 & -2 & 1 & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 1 & -2 \end{bmatrix}_{(n-2)\times n}$$

References

- Numerical Optimization. J. Nocedal, S. Wright. Second Edition. Springer. 2006
- Matrix Computations. G. Golub, Van Loan. Third Edition. Jhons Hopkins University Press. 1996