# Linear least squares problems

Guangning (Gary) Tan

tgnteach@gmail.com

CAS708/CSE700

McMaster University

October 1, 2015

# Contents



---

This PDF can be accessed at

tgn3000.com ⇒ Teaching ⇒ CAS708/CSE700 Week 4b

# Review

$$\min_{\mathbf{x}} ||A\mathbf{x} - \mathbf{b}||_2^2$$

- $\mathbf{x} \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ with $m > n$, hence tall
- Has full column rank, $\text{rank}(A) = n$
- $A\mathbf{x} = \mathbf{b}$ is an overdetermined system, no exact solution
- Arises in data fitting problems, e.g.,
  - $y = ax^3 + bx^2 + cx + d$
  - $y = ae^{bx}$
  - $y = a + b \log x$

# Unconstrained optimization problem

Objective function

$$f(\mathbf{x}) = ||A\mathbf{x} - \mathbf{b}||_2^2 = \sum_{i=1}^{m} \left( b_i - \sum_{j=1}^{n} a_{ij} x_j \right)^2$$

- Convex problem, optimal value (minimum) exists
- Necessary condition

$$\nabla f = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right) = \mathbf{0}^T$$

# Normal equations

- For $j = 1 : n$,

$$\frac{\partial f}{\partial x_j} = \sum_{i=1}^{m} \left[ \left( b_i - \sum_{k=1}^{n} a_{ik} x_j \right) (-a_{ij}) \right] = 0$$

- Rearranging terms gives normal equations

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

- Since $A$ has full column rank, $A^T A$ is SPD; can do Cholesky
- Solution is $\mathbf{x} = (A^T A)^{-1} A^T\mathbf{b}$

# Normal equations (cont.)

$$A^T A\mathbf{x} = A^T\mathbf{b}, \qquad \mathbf{x} = (A^T A)^{-1} A^T\mathbf{b}$$

- $A^\dagger = (A^T A)^{-1} A^T$ is Moore-Penrose inverse of $A$
- Assume $A$ has singular values $\sigma_1 \geq \ldots \geq \sigma_n > 0$
- Condition number $\kappa(A) = \sigma_1/\sigma_n$
- $\kappa(A^T A) = \kappa(A)^2$ can be large

# Polynomial data fitting

$$b \approx x_0 + x_1 v + x_2 v^2 + \ldots, x_{n-1} v^{n-1}$$

- Sample points $(t_i, y_i)$, $i = 1 : m$, $m > n$
- Find undetermined coefficients $x_0, \ldots, x_{n-1}$ to fit data

$$\begin{bmatrix} 1 & v_1 & v_1^2 & \ldots & v_1^{n-1} \\ 1 & v_2 & v_2^2 & \ldots & v_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & v_m & v_m^2 & \ldots & v_m^{n-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} \approx \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix}$$

  written $V\mathbf{x} \approx \mathbf{b}$
- Wish to find $\min_{\mathbf{x}} ||V\mathbf{x} - \mathbf{b}||_2$
- $V$ is Vandermonde matrix, with nodes or knots $v_1, \ldots, v_m$, assumed real and distinct
- Will see Vandermonde again in polynomial interpolation

# Orthogonal transformations

- Singular value decomposition (SVD)
  - `[U,S,V] = svd(A)`
  - Used when $A$ is (nearly) rank deficient
- QR
  - `[Q, R] = qr(A)`
  - “Standard” approach
  - Expensive than normal equations when $m \gg n$ but more robust
  - Gram-Schmidt
  - Householder (elementary reflectors)
  - Givens (plane rotations)

# Singular value decomposition (SVD)

$$A = U\Sigma V^T$$

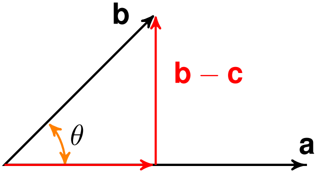
- $A \in \mathbb{R}^{m\times n}$
- $U$, $V$ are orthogonal matrices, of size $m$, $n$, resp.
- $\Sigma$ is a diagonal matrix, with singular values $\sigma_i$ on the diagonal, $i = 1 : \min\{m, n\}$
- Generalized inverse, or Moore-Penrose pseudoinverse of $A$ is $A^\dagger = V\Sigma^\dagger U^T$
- $\Sigma^\dagger$ is pseudoinverse of $\Sigma$, formed by replacing every nonzero diagonal entry by its reciprocal and transposing it

# SVD (cont.)

$$A_{m\times n} = U \begin{bmatrix} \begin{matrix} \sigma_1 & & & & & & \\ & \sigma_2 & & & & & \\ & & \ddots & & & & \\ & & & \sigma_k & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{matrix} \\ \hline \mathbf{0}_{(m-n)\times n} \end{bmatrix} V^T \qquad \text{where } \sigma_1 \geq \ldots \geq \sigma_k > 0$$

$$A^{\dagger}_{n\times m} = V \left[ \begin{array}{ccccccc|c} \sigma_1^{-1} & & & & & & & \\ & \sigma_2^{-1} & & & & & & \\ & & \ddots & & & & & \\ & & & \sigma_k^{-1} & & & & \mathbf{0}_{n\times(m-n)} \\ & & & & 0 & & & \\ & & & & & \ddots & & \\ & & & & & & 0 & \end{array} \right] U^T$$

# Gram-Schmidt orthogonalization




Basic idea

- $\mathbf{a}^T\mathbf{b} = |\mathbf{a}||\mathbf{b}|\cos\theta$
- $(\mathbf{b}-\mathbf{c}) \perp \mathbf{a}$

$\mathbf{c} = |\mathbf{b}|\cos\theta \cdot \mathbf{q}_1$

- Let $\mathbf{q}_1 = \mathbf{a}/|\mathbf{a}|$
- $\mathbf{c}$ is the projection of $\mathbf{b}$ onto $\mathbf{q}_1$
- $\mathbf{b}-\mathbf{c} = \mathbf{b} - |\mathbf{b}| \cdot \frac{\mathbf{q}_1^T\mathbf{b}}{|\mathbf{q}_1||\mathbf{b}|} \cdot \mathbf{q}_1 = \mathbf{b} - \left(\mathbf{q}_1^T\mathbf{b}\right) \cdot \mathbf{q}_1$
- An orthogonal basis consists of

$$\mathbf{q}_1 = \mathbf{a}/|\mathbf{a}| \quad \text{and} \quad \mathbf{q}_2 = (\mathbf{b}-\mathbf{c})/|\mathbf{b}-\mathbf{c}|$$

# Gram-Schmidt (cont.)

**Algorithm (Gram-Schmidt).**

for $j = 1 : n$

  $\mathbf{q}_j \leftarrow \mathbf{a}_j$

  for $i = 1 : j - 1$

    $r_{ij} \leftarrow \mathbf{q}_i^T \mathbf{a}_j$

    $\mathbf{q}_j \leftarrow \mathbf{q}_j - r_{ij}\mathbf{q}_i$

  end

  $r_{jj} \leftarrow ||\mathbf{q}_j||_2$

  $\mathbf{q}_j \leftarrow \mathbf{q}_j / r_{jj}$    % normalization

end

- Need separate storage for $A$, $Q$

# Gram-Schmidt (cont.)

**Algorithm (Modified Gram-Schmidt).**

for $j = 1 : n$

$\quad r_{jj} \leftarrow ||\mathbf{q}_j||_2$

$\quad \mathbf{q}_j \leftarrow \mathbf{q}_j / r_{jj}$ % normalization

$\quad$ for $i = j + 1 : n$

$\quad\quad r_{ij} \leftarrow \mathbf{q}_i^T \mathbf{a}_j$

$\quad\quad \mathbf{a}_j \leftarrow \mathbf{a}_j - r_{ij}\mathbf{q}_i$

$\quad$ end

end

- Can update $A$ to $Q$ when iterating $j$

# Example

$$A_0 = \begin{bmatrix} 1 & 2 & 3 \\ 1 & -1 & 2 \\ -1 & 2 & -1 \\ -2 & 0 & 1 \\ 1 & 2 & -1 \end{bmatrix} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3)$$

$$r_{11} = ||\mathbf{a}_1|| = 2\sqrt{2}$$

$$\mathbf{q}_1 = \mathbf{a}_1/r_{11} = (1, 1, -1, -2, 1)^T/2\sqrt{2}$$

$$r_{12} = \mathbf{q}_1^T \mathbf{a}_2 = 1/2\sqrt{2}$$

$$\mathbf{a}_2 \leftarrow \mathbf{a}_2 - r_{12}\mathbf{q}_1 = (15, -9, 17, 2, 15)^T/8$$

$$r_{13} = \mathbf{q}_1^T \mathbf{a}_3 = 3/2\sqrt{2}, \quad \mathbf{a}_3 \leftarrow \mathbf{a}_3 - r_{13}\mathbf{q}_1 = \ldots$$

$$A_1 = \begin{bmatrix} 1/2\sqrt{2} & 15/8 & 3 \\ 1/2\sqrt{2} & -9/8 & 2 \\ -1/2\sqrt{2} & 17/8 & -1 \\ -2/2\sqrt{2} & 2/8 & 1 \\ 1/2\sqrt{2} & 15/8 & -1 \end{bmatrix} = (\mathbf{q}_1, \mathbf{a}_2, \mathbf{a}_3)$$

$$r_{22} = ||\mathbf{a}_2|| = \sqrt{206}/4$$

$$\mathbf{q}_2 = \mathbf{a}_2/r_{22} = (15, -9, 17, 2, 15)^T/2\sqrt{206}$$

$$r_{23} = \mathbf{q}_2^T\mathbf{a}_3 = \ldots, \quad \mathbf{a}_3 \leftarrow \mathbf{a}_3 - r_{23}\mathbf{q}_2 = \ldots$$

$$r_{33} = ||\mathbf{a}_3||, \quad \mathbf{q}_3 = \mathbf{a}_3/r_{33}$$

$$Q = (\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)$$

$$R = \left[\begin{array}{ccc} r_{11} & r_{12} & r_{13} \\ & r_{22} & r_{23} \\ & & r_{33} \\ \hline & \mathbf{0}_{2\times 3} & \end{array}\right]$$

$$A = QR$$

# Householder orthogonalization

$$H = I - 2\mathbf{u}\mathbf{u}^T$$

- Householder transformation of a unit vector $\mathbf{u}$
- Orthogonal and symmetric
- Given vector $\mathbf{a}$, find $\mathbf{u}$ s.t. $\alpha\mathbf{e}_1 = H\mathbf{a}$

$$\mathbf{u} = (\mathbf{a} - \alpha\mathbf{e}_1)/(2\mathbf{u}^T\mathbf{a})$$

- Can pick $\mathbf{v} = \mathbf{a} - \alpha\mathbf{e}_1$ and then normalize
- To avoid cancellation, choose $\alpha = -\text{sgn}(\mathbf{a}(1))||\mathbf{a}||_2$

$$\mathbf{v} = \mathbf{a} + \text{sgn}(\mathbf{a}(1))||\mathbf{a}||_2\mathbf{e}_1, \qquad \mathbf{u} = \mathbf{v}/||\mathbf{v}||_2$$

# Example

$$A_0 = \begin{bmatrix} 1 & 2 & 3 \\ 1 & -1 & 2 \\ -1 & 2 & -1 \\ -2 & 0 & 1 \\ 1 & 2 & -1 \end{bmatrix}$$

$$\mathbf{v} = \mathbf{a} + \operatorname{sgn}(\mathbf{a}(1))||\mathbf{a}||_2\mathbf{e}_1$$

$$= \begin{bmatrix} 1 \\ 1 \\ -1 \\ -2 \\ 1 \end{bmatrix} + 2\sqrt{2}\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1+2\sqrt{2} \\ 1 \\ -1 \\ -2 \\ 1 \end{bmatrix}$$

$$\mathbf{u} = \mathbf{v}/||\mathbf{v}||_2, \quad H_1 = I - 2\mathbf{u}\mathbf{u}^T, \quad A_1 = H_1 A_0$$

```
>> v = A(:,1); v(1) = v(1)+norm(A(:,1))*sign(v(1));
>> u = v/norm(v)
u =
     0.8227
     0.2149
    -0.2149
    -0.4298
     0.2149

>> H1 = eye(5)-2*u*u'; A = H1*A
A =
    -2.8284   -0.3536   -1.0607
     0.0000   -1.6148    0.9393
    -0.0000    2.6148    0.0607
    -0.0000    1.2295    3.1213
     0.0000    1.3852   -2.0607
```

```
>> v = A(2:end,2);
>> v(1) = v(1)+norm(A(2:end,2))*sign(v(1));
>> u = v/norm(v)
u =
   -0.8515
    0.4279
    0.2012
    0.2267

>> H2 = blkdiag(eye(1),eye(4)-2*u*u');
>> A = H2*A
A =
   -2.8284   -0.3536   -1.0607
   -0.0000    3.5882   -0.1045
    0.0000    0.0000    0.5853
   -0.0000   -0.0000    3.3680
    0.0000         0   -1.7827
```

```
>> v = A(3:end,3);
>> v(1) = v(1)+norm(A(3:end,3))*sign(v(1));
>> u = v/norm(v)
u =
    0.7589
    0.5756
   -0.3047

>> H3 = blkdiag(eye(2),eye(3)-2*u*u');
>> A = H3*A
A =
   -2.8284   -0.3536   -1.0607
   -0.0000    3.5882   -0.1045
    0.0000    0.0000   -3.8554
   -0.0000   -0.0000    0.0000
    0.0000    0.0000   -0.0000
```

# Householder orthogonalization (cont.)

- Recall that $H_1, H_2, H_3$ are orthogonal matrices

$$R = H_3 H_2 H_1 A = Q^{-1} A$$
$$QR = A$$

where

$$Q = H_1^{-1} H_2^{-1} H_3^{-1} = H_1^T H_2^T H_3^T$$

# Householder orthogonalization (cont.)

**Algorithm (Householder).**

```
[m,n]=size(A); p=zeros(1,n);
for j=1:n
    a ←A(j:m,j);
    e1 ←[1; zeros(m-j,1)];
    u ←a+sign(a(1))*norm(a)*e1;
    u ←u/norm(u);    % normalization
    A(j:m,j:n)-=2*u*(u'*A(j:m,j:n));    % A_j ← H_j A_{j-1}
    % Store u
    p(k)=u(1);
    A(k+1:m,k)=u(2:m-k+1);
end
```

# Givens rotation

$$GA_0 = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \cos\theta & \sin\theta \\ & & & -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & \times & \times \\ 1 & \times & \times \\ -1 & \times & \times \\ -2 & \times & \times \\ 1 & \times & \times \end{bmatrix} = \begin{bmatrix} 1 & \times & \times \\ 1 & \times & \times \\ -1 & \times & \times \\ a & \times & \times \\ 0 & \times & \times \end{bmatrix}$$

- Choose $\theta$ s.t. $2\sin\theta + \cos\theta = 0$
- $\sin\theta = 1/\sqrt{5}, \quad \cos\theta = -2/\sqrt{5}$
- $a = \|(-2, 1)^T\|_2 = \sqrt{5}$