

AMS526: Numerical Analysis I (Numerical Linear Algebra)

Lecture 9: Least Squares Problems

Xiangmin Jiao

SUNY Stony Brook

October 2, 2008

Outline

1 Linear Least Squares Problems

2 Floating Point Arithmetic

Linear Least Squares Problems

- Overdetermined system of equations $\mathbf{Ax} = \mathbf{b}$, where \mathbf{A} has more rows than columns and has full rank, in general has no solutions
- Example application: Polynomial least squares fitting
- In general, minimize the residual $\mathbf{r} = \mathbf{b} - \mathbf{Ax}$
- In terms of 2-norm, we obtain linear least squares problem: Given $\mathbf{A} \in \mathbb{C}^{m \times n}$, $m \geq n$, and $\mathbf{b} \in \mathbb{C}^m$, find $\mathbf{x} \in \mathbb{C}^n$ such that $\|\mathbf{b} - \mathbf{Ax}\|_2$ is minimized
- If \mathbf{A} has full rank, the minimizer \mathbf{x} is the solution to the normal equation

$$\mathbf{A}^* \mathbf{Ax} = \mathbf{A}^* \mathbf{b}$$

or in terms of the *pseudoinverse* \mathbf{A}^+ ,

$$\mathbf{x} = \mathbf{A}^+ \mathbf{b}, \quad \text{where } \mathbf{A}^+ = (\mathbf{A}^* \mathbf{A})^{-1} \mathbf{A}^* \in \mathbb{C}^{n \times m}$$

Geometric Interpretation

- \mathbf{Ax} is in $\text{range}(\mathbf{A})$, and the point in $\text{range}(\mathbf{A})$ closest to \mathbf{b} is its orthogonal projection onto $\text{range}(\mathbf{A})$
- Residual \mathbf{r} is then orthogonal to $\text{range}(\mathbf{A})$, and hence $\mathbf{A}^* \mathbf{r} = \mathbf{A}^*(\mathbf{b} - \mathbf{Ax}) = \mathbf{0}$
- \mathbf{Ax} is orthogonal projection of \mathbf{b} , where $\mathbf{x} = \mathbf{A}^+ \mathbf{b}$, so $\mathbf{P} = \mathbf{AA}^+ = \mathbf{A}(\mathbf{A}^* \mathbf{A})^{-1} \mathbf{A}^*$ is orthogonal projection (recall lecture 6)

Solution of Least Squares Problems

- One approach is to solve normal equation $\mathbf{A}^* \mathbf{A} \mathbf{x} = \mathbf{A}^* \mathbf{b}$ directly using Cholesky factorization
 - ▶ Is unstable, but is very efficient if $m \gg n$ ($mn^2 + \frac{1}{3}n^3$)
- More robust approach is to use QR factorization $\mathbf{A} = \hat{\mathbf{Q}} \hat{\mathbf{R}}$
 - ▶ \mathbf{b} can be projected onto $\text{range}(\mathbf{A})$ by $\mathbf{P} = \hat{\mathbf{Q}} \hat{\mathbf{Q}}^*$, and therefore $\hat{\mathbf{Q}} \hat{\mathbf{R}} \mathbf{x} = \hat{\mathbf{Q}} \hat{\mathbf{Q}}^* \mathbf{b}$
 - ▶ Left-multiply by $\hat{\mathbf{Q}}^*$ and we get $\hat{\mathbf{R}} \mathbf{x} = \hat{\mathbf{Q}}^* \mathbf{b}$ (note $\mathbf{A}^+ = \hat{\mathbf{R}}^{-1} \hat{\mathbf{Q}}^*$)

Least squares via QR Factorization

Compute reduced QR factorization $\mathbf{A} = \hat{\mathbf{Q}} \hat{\mathbf{R}}$
Compute vector $\mathbf{c} = \hat{\mathbf{Q}}^* \mathbf{b}$
Solve upper-triangular system $\hat{\mathbf{R}} \mathbf{x} = \mathbf{c}$ for \mathbf{x}

- Computation is dominated by QR factorization ($2mn^2 - \frac{2}{3}n^3$)
- Question: If Householder QR is used, how to compute $\hat{\mathbf{Q}}^* \mathbf{b}$?

Solution of Least Squares Problems

- One approach is to solve normal equation $\mathbf{A}^* \mathbf{A} \mathbf{x} = \mathbf{A}^* \mathbf{b}$ directly using Cholesky factorization
 - ▶ Is unstable, but is very efficient if $m \gg n$ ($mn^2 + \frac{1}{3}n^3$)
- More robust approach is to use QR factorization $\mathbf{A} = \hat{\mathbf{Q}} \hat{\mathbf{R}}$
 - ▶ \mathbf{b} can be projected onto $\text{range}(\mathbf{A})$ by $\mathbf{P} = \hat{\mathbf{Q}} \hat{\mathbf{Q}}^*$, and therefore $\hat{\mathbf{Q}} \hat{\mathbf{R}} \mathbf{x} = \hat{\mathbf{Q}} \hat{\mathbf{Q}}^* \mathbf{b}$
 - ▶ Left-multiply by $\hat{\mathbf{Q}}^*$ and we get $\hat{\mathbf{R}} \mathbf{x} = \hat{\mathbf{Q}}^* \mathbf{b}$ (note $\mathbf{A}^+ = \hat{\mathbf{R}}^{-1} \hat{\mathbf{Q}}^*$)

Least squares via QR Factorization

Compute reduced QR factorization $\mathbf{A} = \hat{\mathbf{Q}} \hat{\mathbf{R}}$
Compute vector $\mathbf{c} = \hat{\mathbf{Q}}^* \mathbf{b}$
Solve upper-triangular system $\hat{\mathbf{R}} \mathbf{x} = \mathbf{c}$ for \mathbf{x}

- Computation is dominated by QR factorization ($2mn^2 - \frac{2}{3}n^3$)
- Question: If Householder QR is used, how to compute $\hat{\mathbf{Q}}^* \mathbf{b}$?
- Answer: Compute $\mathbf{Q}^* \mathbf{b}$ (where \mathbf{Q} is from full QR factorization) and then take $\mathbf{b}_{1:n}$

Solution by SVD

- Using $\mathbf{A} = \hat{\mathbf{U}}\hat{\Sigma}\mathbf{V}^*$, \mathbf{b} can be projected onto $\text{range}(\mathbf{A})$ by $\mathbf{P} = \hat{\mathbf{U}}\hat{\mathbf{U}}^*$, and therefore $\hat{\mathbf{U}}\hat{\Sigma}\mathbf{V}^*\mathbf{x} = \hat{\mathbf{U}}\hat{\mathbf{U}}^*\mathbf{b}$
- Left-multiply by $\hat{\mathbf{U}}^*$ and we get $\hat{\Sigma}\mathbf{V}^*\mathbf{x} = \hat{\mathbf{U}}^*\mathbf{b}$

Least squares via SVD

Compute reduced SVD factorization $\mathbf{A} = \hat{\mathbf{U}}\hat{\Sigma}\mathbf{V}^*$

Compute vector $\mathbf{c} = \hat{\mathbf{U}}^*\mathbf{b}$

Solve diagonal system $\hat{\Sigma}\mathbf{w} = \mathbf{c}$ for \mathbf{w}

Set $\mathbf{x} = \mathbf{V}\mathbf{w}$

- Work is dominated by SVD, which is $\sim 2mn^2 + 11n^3$ flops, very expensive if $m \approx n$
- Best numerical stability
- Question: If \mathbf{A} is rank deficient, how to solve $\mathbf{Ax} = \mathbf{b}$?

Solution by SVD

- Using $\mathbf{A} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\mathbf{V}^*$, \mathbf{b} can be projected onto $\text{range}(\mathbf{A})$ by $\mathbf{P} = \hat{\mathbf{U}}\hat{\mathbf{U}}^*$, and therefore $\hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\mathbf{V}^*\mathbf{x} = \hat{\mathbf{U}}\hat{\mathbf{U}}^*\mathbf{b}$
- Left-multiply by $\hat{\mathbf{U}}^*$ and we get $\hat{\mathbf{\Sigma}}\mathbf{V}^*\mathbf{x} = \hat{\mathbf{U}}^*\mathbf{b}$

Least squares via SVD

Compute reduced SVD factorization $\mathbf{A} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\mathbf{V}^*$

Compute vector $\mathbf{c} = \hat{\mathbf{U}}^*\mathbf{b}$

Solve diagonal system $\hat{\mathbf{\Sigma}}\mathbf{w} = \mathbf{c}$ for \mathbf{w}

Set $\mathbf{x} = \mathbf{V}\mathbf{w}$

- Work is dominated by SVD, which is $\sim 2mn^2 + 11n^3$ flops, very expensive if $m \approx n$
- Best numerical stability
- Question: If \mathbf{A} is rank deficient, how to solve $\mathbf{Ax} = \mathbf{b}$?
- Answer: \mathbf{x} is no longer unique. Constrain \mathbf{x} to be orthogonal to null space of \mathbf{A} .

Outline

1 Linear Least Squares Problems

2 Floating Point Arithmetic

Floating Point Representations

- Computers can only use finite number of bits to represent a real number
 - ▶ Numbers cannot be arbitrarily large or small (associated risks of *overflow* and *underflow*)
 - ▶ There must be gaps between representable numbers (potential round-off errors)
- Commonly used computer-representations are floating point representations, which resemble scientific notation

$$\pm(d_0 + d_1\beta^{-1} + \cdots + d_{p-1}\beta^{-p+1})\beta^e, \quad 0 \leq d_i \leq \beta$$

where β is base, p is digits of precision, and e is exponent between e_{min} and e_{max}

- Normalize if $d_0 \neq 0$ (except for 0)
- Gaps between adjacent numbers scale with size of numbers
- Relative resolution given by *machine epsilon* $\epsilon_{\text{machine}} = 0.5\beta^{1-p}$
- For all x , there exists a floating point x' such that

$$|x - x'| \leq \epsilon_{\text{machine}}|x|$$

IEEE Floating Point Representations

- Single precision: 32 bit
 - ▶ 1 sign bit (S), 8 exponent bits (E), 23 significand bits (M),
 $(-1)^S \times 1.M \times 2^{E-127}$
 - ▶ $\epsilon_{\text{machine}}$ is $2^{-24} \approx 6e - 8$
- Double precision: 64 bits
 - ▶ 1 sign bit (S), 11 exponent bits (E), 52 significand bits (M),
 $(-1)^S \times 1.M \times 2^{E-1023}$
 - ▶ $\epsilon_{\text{machine}}$ is $2^{-53} \approx e - 16$
- Special quantities
 - ▶ $+\infty$ and $-\infty$ when operation overflows; e.g., $x/0$ for nonzero x
 - ▶ NaN (Not a Number) is returned when an operation has no well-defined result; e.g., $0/0$, $\sqrt{-1}$, $\arcsin(2)$, NaN

Machine Epsilon

- Define $\text{fl}(x)$ as closest floating point approximation to x
- By definition of $\epsilon_{\text{machine}}$, we have:

For all $x \in \mathbb{R}$, there exists ϵ with $|\epsilon| \leq \epsilon_{\text{machine}}$
such that $\text{fl}(x) = x(1 + \epsilon)$

- Given operation $+$, $-$, \times , and $/$ (denoted by $*$), floating point numbers x and y , and corresponding floating point arithmetic (denoted by \circledast), we require that $x \circledast y = \text{fl}(x * y)$
- This is guaranteed by IEEE floating point arithmetic
- Fundamental axiom of floating point arithmetic:

For all $x, y \in \mathbb{F}$, there exists ϵ with $|\epsilon| \leq \epsilon_{\text{machine}}$
such that $x \circledast y = (x * y)(1 + \epsilon)$

- These properties will be the basis of error analysis with rounding errors