ORF307 – Optimization

3. Least squares

Bartolomeo Stellato — Spring 2023

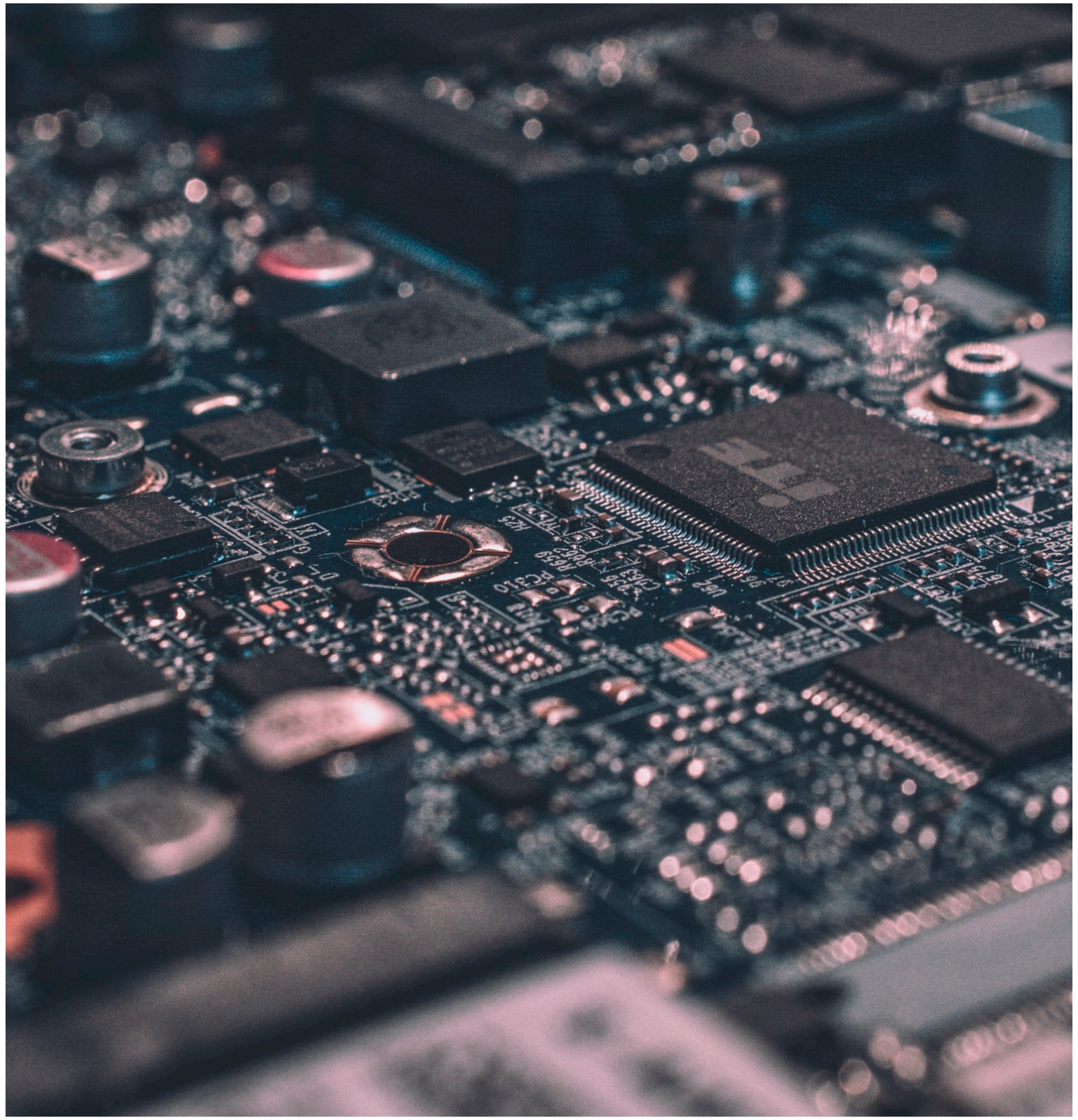
Ed Forum

- Why swapping components of vectors takes significantly less time than performing floating points operations?
- When are sparse matrices actually sparse? How many entries are 0?
- What are “gains” in between “just solve” and “factor-solve”?
- Are we supposed to be able to do the different factorizations (LU, LLT) by hand?

Recap

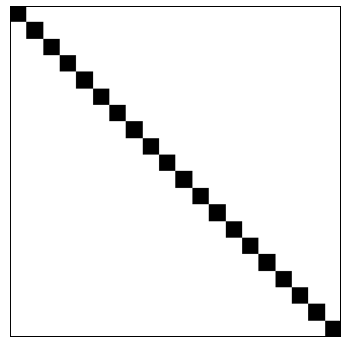
Flop counts

- Computers store real numbers in **floating-point format**
- Basic arithmetic operations (addition, multiplication, etc…) are called **floating point operations (flops)**
- **Algorithm complexity**: total number of flops needed as function of dimensions
- **Execution time** $\approx$ (flops)/(computer speed) *[Very grossly approximated]*
- Modern computers can go at 1 Gflop/sec (10^9 flops/sec)

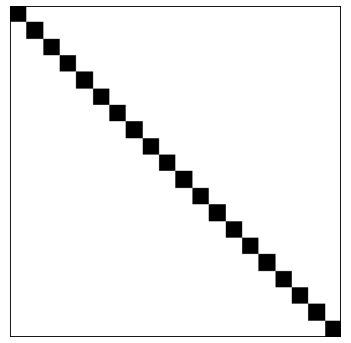
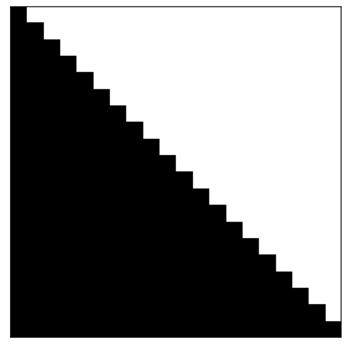


$Ax = b$

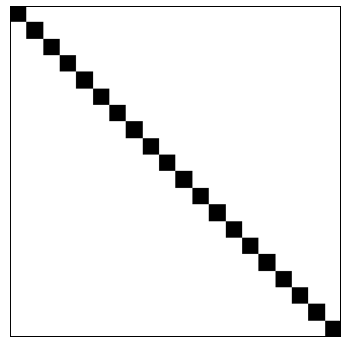
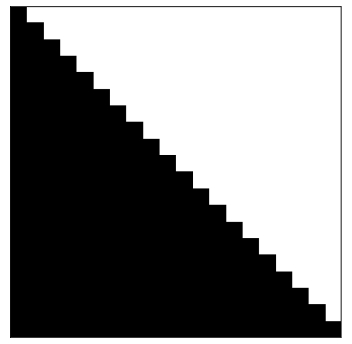
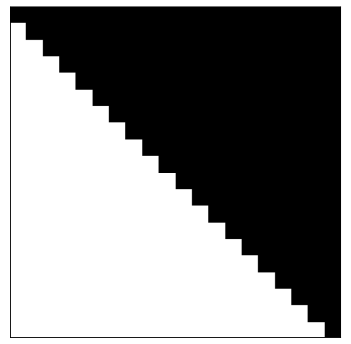
Summary of easy linear systems

		method	flops
	**diagonal** $A = \mathbf{diag}(a_1, \ldots, a_n)$	$x_i = b_i / a_i$	n

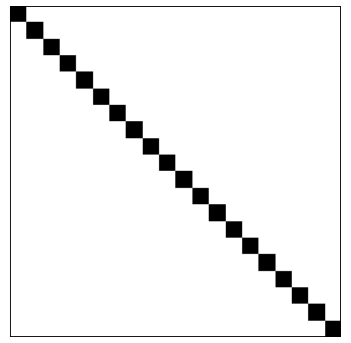
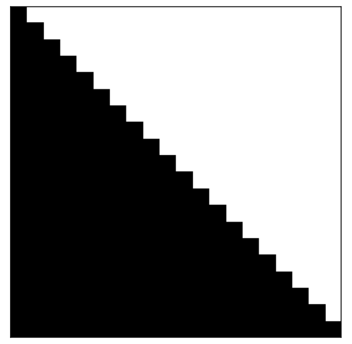
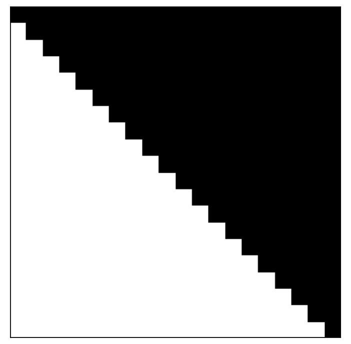
Summary of easy linear systems

		method	flops
	**diagonal** $A = \mathbf{diag}(a_1, \ldots, a_n)$	$x_i = b_i / a_i$	n
	**lower triangular** $A_{ij} = 0$ for $i < j$	forward substitution	n^2

Summary of easy linear systems

		method	flops
	**diagonal** $A = \mathbf{diag}(a_1, \ldots, a_n)$	$x_i = b_i / a_i$	n
	**lower triangular** $A_{ij} = 0$ for $i < j$	forward substitution	n^2
	**upper triangular** $A_{ij} = 0$ for $i > j$	backward substitution	n^2

Summary of easy linear systems

		method	flops
	**diagonal** $A = \mathbf{diag}(a_1, \ldots, a_n)$	$x_i = b_i / a_i$	n
	**lower triangular** $A_{ij} = 0$ for $i < j$	forward substitution	n^2
	**upper triangular** $A_{ij} = 0$ for $i > j$	backward substitution	n^2
	permutation $P_{ij} = 1$ if $j = \pi_i$ else 0	inverse permutation	0

The factor-solve method for solving $Ax = b$

1. **Factor** A as a product of simple matrices:

$$A = A_1 A_2 \cdots A_k, \longrightarrow A_1 A_2, \ldots A_k x = b$$

(A_i diagonal, upper/lower triangular, permutation, etc)

The factor-solve method for solving $Ax = b$

1. **Factor** A as a product of simple matrices:

$$A = A_1 A_2 \cdots A_k, \longrightarrow A_1 A_2, \dots A_k x = b$$

(A_i diagonal, upper/lower triangular, permutation, etc)

2. **Compute** $x = A^{-1}b = A_k^{-1} \cdots A_1^{-1} b$
   by solving k "easy" systems

$$\longrightarrow \quad \begin{aligned} A_1 x_1 &= b \\ A_2 x_2 &= x_1 \\ &\vdots \\ A_k x &= x_{k-1} \end{aligned}$$

The factor-solve method for solving $Ax = b$

1. **Factor** A as a product of simple matrices:

$$A = A_1 A_2 \cdots A_k, \longrightarrow A_1 A_2, \dots A_k x = b$$

(A_i diagonal, upper/lower triangular, permutation, etc)

2. **Compute** $x = A^{-1}b = A_k^{-1} \cdots A_1^{-1} b$
   by solving k "easy" systems $\longrightarrow$

$$\begin{aligned} A_1 x_1 &= b \\ A_2 x_2 &= x_1 \\ &\vdots \\ A_k x &= x_{k-1} \end{aligned}$$

Note: step 2 is much cheaper than step 1

Multiple right-hand sides

You now have factored A and you want to solve d linear systems with different righ-hand side m-vectors b_i

$$Ax = b_1 \qquad Ax = b_2 \qquad \dots \quad Ax = b_d$$

Multiple right-hand sides

You now have factored A and you want to solve d linear systems with different righ-hand side m-vectors b_i

$$Ax = b_1 \qquad Ax = b_2 \qquad \ldots \quad Ax = b_d$$

Factorization-caching procedure

1. Factor $A = A_1, \ldots, A_k$ **only once** (expensive)
2. Solve all linear systems using **the same factorization** (cheap)

Multiple right-hand sides

You now have factored A and you want to solve d linear systems with different righ-hand side m-vectors b_i

$$Ax = b_1 \qquad Ax = b_2 \qquad \dots \quad Ax = b_d$$

Factorization-caching procedure

1. Factor $A = A_1, \dots, A_k$ **only once** (expensive)
2. Solve all linear systems using **the same factorization** (cheap)

Solve many “at the price of one”

LL^T (Cholesky) Factorization

Every positive definite matrix A can be factored as

$$A = LL^T$$

L lower triangular

LL^T (Cholesky) Factorization

Every positive definite matrix A can be factored as

$$A = LL^T$$

L lower triangular

Procedure

- Works only on **symmetric with positive definite** matrices
- No need to permute as in LU
- One of infinite possible choices of L

LL^T (Cholesky) Factorization

Every positive definite matrix A can be factored as

$$A = LL^T$$

L lower triangular

Procedure

- Works only on **symmetric with positive definite** matrices
- No need to permute as in LU
- One of infinite possible choices of L

Complexity

- $(1/3)n^3$ flops (half of LU decomposition)
- Less if A has special structure (sparse, diagonal, etc)

LL^T (Cholesky) Solution

$$Ax = b, \quad \Rightarrow \quad LL^T x = b$$

Iterations

1. *Forward substitution*: Solve $Lx_1 = b$ (n^2 flops)
2. *Backward substitution*: Solve $L^T x = x$ (n^2 flops)

LL^T (Cholesky) Solution

$$Ax = b, \quad \Rightarrow \quad LL^T x = b$$

Iterations

1. *Forward substitution*: Solve $Lx_1 = b$ (n^2 flops)
2. *Backward substitution*: Solve $L^T x = x$ (n^2 flops)

Complexity

- Factor + solve: $(1/3)n^3 + 2n^2 \approx (1/3)n^3$ (for large n)
- Just solve (prefactored): $2n^2$

Today's lecture

Least squares

- Least squares optimization
- Gram matrix
- Solving least squares
- Example

Least squares optimization

Solving overdetermined linear systems

You have an *overdetermined* $m \times n$ linear system (m > n)

$$Ax = b$$

(with tall A)

Solving overdetermined linear systems

You have an *overdetermined* $m \times n$ linear system (m > n)

$$Ax = b$$

(with tall A)

example

Typically no solution

$$\begin{bmatrix} 2 & 0 \\ -1 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

Least squares problem

residual vector

$$r = Ax - b$$

$r_i = (Ax - b)_i$

→

Goal: make it as small as possible

minimize $\|r\|$

Least squares problem

residual vector

$r = Ax - b$ → **Goal:** make it as small as possible

minimize $\|r\|$

Least squares problem

$$\text{minimize} \quad \|Ax - b\|_2^2$$

- x is the *decision variable*
- $\|Ax - b\|_2^2$ is the *objective function*

Least squares solution

minimize $\|Ax - b\|_2^2$

Least squares solution

$\|r\|_2^2 = r^T r$

$$\text{minimize} \quad \|\overbrace{Ax - b}^{r}\|_2^2$$

$x^\star$ is a *solution* of least squares problem if

optimality condition $\|Ax^\star - b\|^2 \leq \|Ax - b\|^2$, for any n-vector x

Least squares solution

minimize $\quad \|Ax - b\|_2^2$

$x^\star$ is a *solution* of least squares problem if

optimality condition $\quad \|Ax^\star - b\|^2 \leq \|Ax - b\|^2, \quad$ for any n-vector x

$x^\star$ need not (and usually does not) satisfy $Ax^\star = b$

What happens if $x^\star$ does satisfy $Ax^\star = b$?

Column interpretation

$$A = \left[a_1, \ldots, a_n\right], \qquad a_1, \ldots, a_n \text{ are columns of } A$$

Goal: find a linear combination of the columns of A that is closest to b

$$\|Ax - b\|^2 = \|(x_1 a_1 + \cdots + x_n a_n) - b\|^2$$

Column interpretation

$$A = \begin{bmatrix} a_1, \ldots, a_n \end{bmatrix}, \qquad a_1, \ldots, a_n \text{ are columns of } A$$

Goal: find a linear combination of the columns of A that is closest to b

$$\|Ax - b\|^2 = \|(x_1 a_1 + \cdots + x_n a_n) - b\|^2$$

If $x^\star$ is a solution of the least squares problem, the m-vector

$$Ax^\star = x_1^\star a_1 + \cdots + x_n^\star a_n$$

is the closest to b among all linear combinations of the columns of A

Row interpretation

$$A = \begin{bmatrix} \tilde{a}_1^T \\ \vdots \\ \tilde{a}_m^T \end{bmatrix}, \qquad \tilde{a}_1^T, \ldots, \tilde{a}_m^T \text{ are rows of } A$$

The residual components are $r_i = \tilde{a}_i^T x - b_i$

Row interpretation

$$A = \begin{bmatrix} \tilde{a}_1^T \\ \vdots \\ \tilde{a}_m^T \end{bmatrix}, \qquad \tilde{a}_1^T, \ldots, \tilde{a}_m^T \text{ are rows of } A$$

The residual components are $r_i = \tilde{a}_i^T x - b_i$

Goal minimize sum of squares of the residuals

$$\|Ax - b\|^2 = (\tilde{a}_1^T x - b_1)^2 + \cdots + (\tilde{a}_m^T x - b_m)^2$$

Row interpretation

$$A = \begin{bmatrix} \tilde{a}_1^T \\ \vdots \\ \tilde{a}_m^T \end{bmatrix}, \qquad \tilde{a}_1^T, \ldots, \tilde{a}_m^T \text{ are rows of } A$$

The residual components are $r_i = \tilde{a}_i^T x - b_i$

Goal minimize sum of squares of the residuals

$$\|Ax - b\|^2 = (\tilde{a}_1^T x - b_1)^2 + \cdots + (\tilde{a}_m^T x - b_m)^2$$

Comparison

- Solving $Ax = b$ forces all residuals to be zero
- Least squares attempts to make them small

Example

$$\overset{A}{\begin{bmatrix} 2 & 0 \\ -1 & 1 \\ 0 & 2 \end{bmatrix}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \overset{b}{\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}}$$

Least squares problem

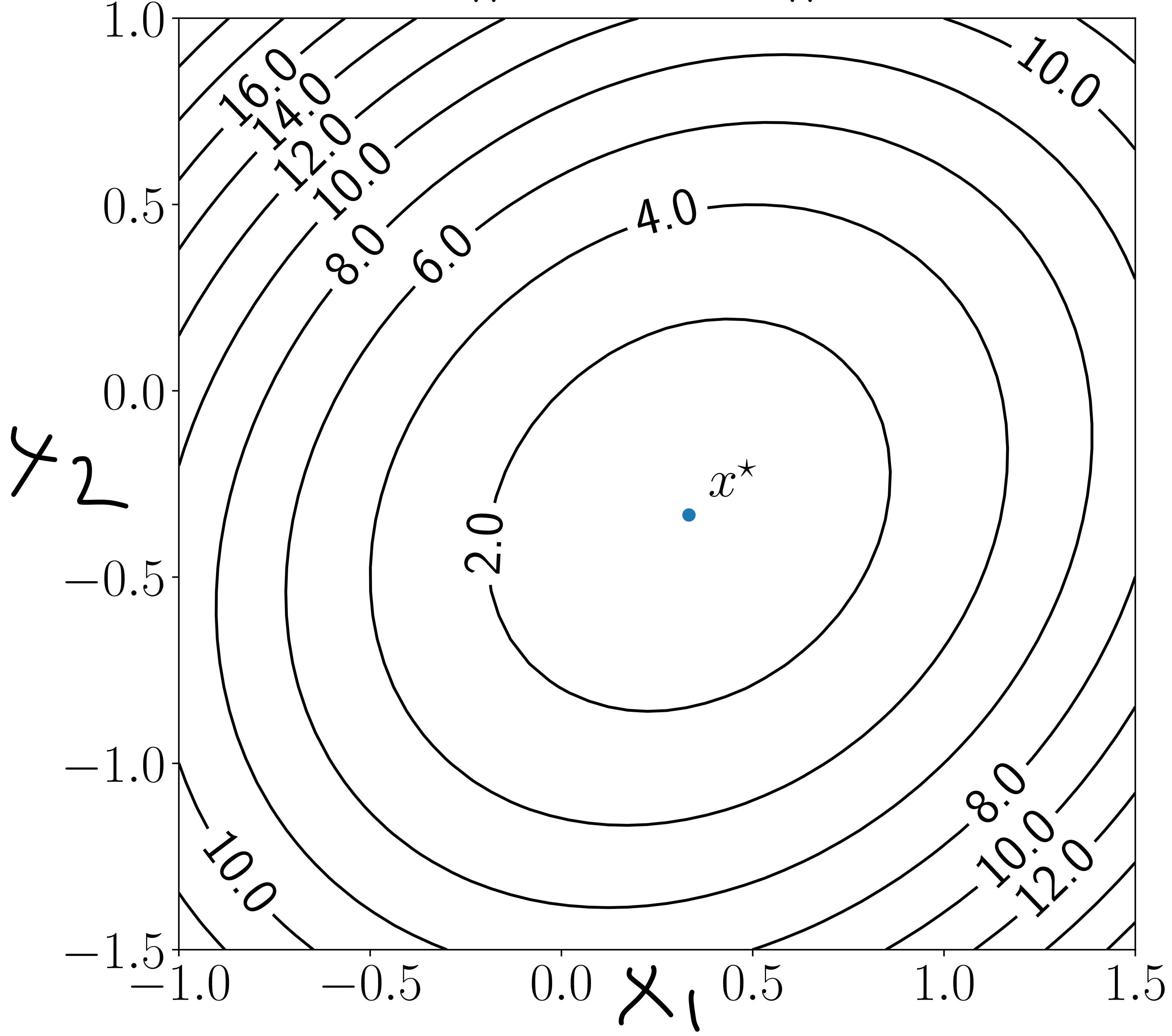
Compute x to minimize

$$\|Ax - b\|^2 = (2x_1 - 1)^2 + (-x_1 + x_2)^2 + (2x_2 + 1)^2$$

Example

$$\overset{A}{\begin{bmatrix} 2 & 0 \\ -1 & 1 \\ 0 & 2 \end{bmatrix}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \overset{b}{\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}}$$

$\|Ax - b\|^2$



Least squares problem

Compute x to minimize

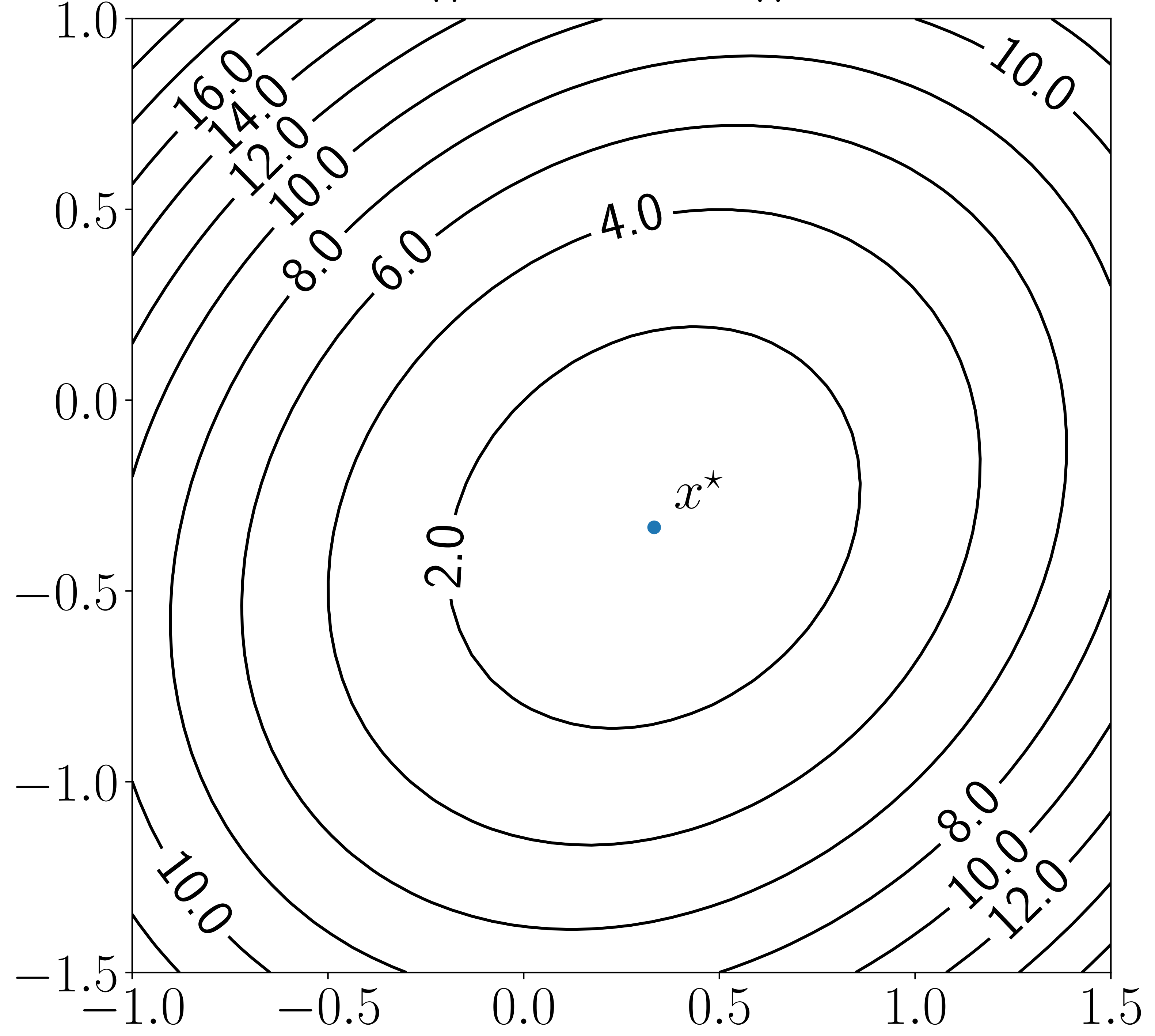
$$\|Ax - b\|^2 = (2x_1 - 1)^2 + (-x_1 + x_2)^2 + (2x_2 + 1)^2$$

Solution $x^\star = (1/3, -1/3)$ (via calculus)

Example

$$\overset{A}{\begin{bmatrix} 2 & 0 \\ -1 & 1 \\ 0 & 2 \end{bmatrix}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \overset{b}{\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}}$$

$\|Ax - b\|^2$



Least squares problem

Compute x to minimize

$$\|Ax - b\|^2 = (2x_1 - 1)^2 + (-x_1 + x_2)^2 + (2x_2 + 1)^2$$

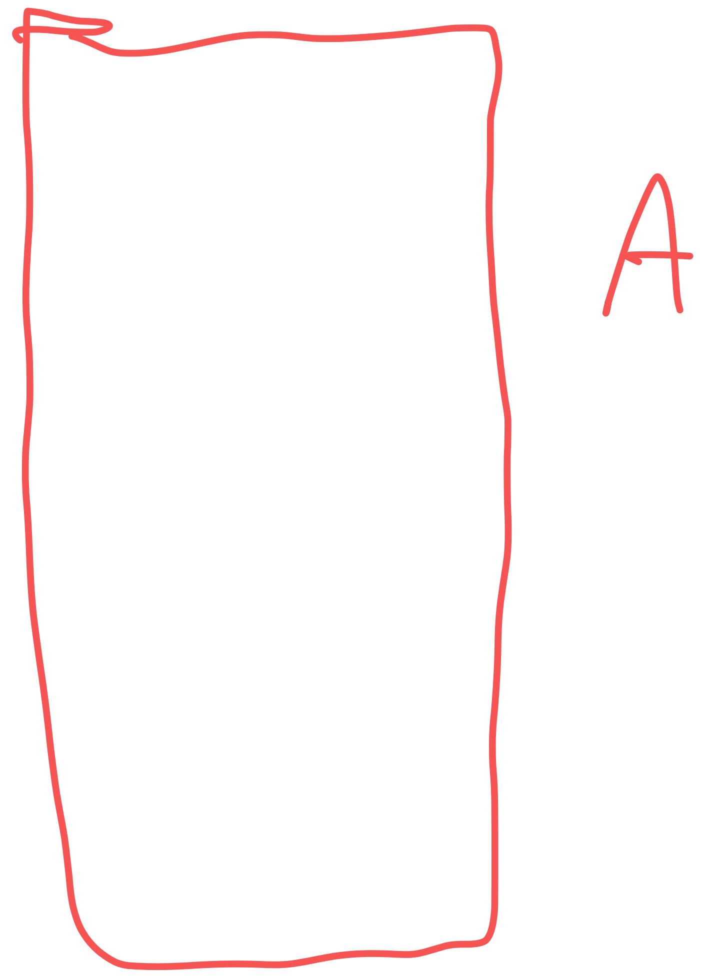
Solution $x^\star = (1/3, -1/3)$ (via calculus)

Interpretations

- $\|Ax^\star - b\|^2 = 2/3$ smallest possible value of $\|Ax - b\|^2$
- $Ax^\star = (2/3, -2/3, -2/3)$ is the linear combination of columns of A closest to b

Gram matrix

Gram matrix



Given an $m \times n$ matrix A with columns $a_1, \ldots, a_n$

the **Gram matrix** of A is

$$A^T A = \begin{bmatrix} a_1^T a_1 & a_1^T a_2 & \ldots & a_1^T a_n \\ a_2^T a_1 & a_2^T a_2 & \ldots & a_2^T a_n \\ \vdots & \vdots & \ddots & \vdots \\ a_n^T a_1 & a_n^T a_2 & \ldots & a_n^T a_n \end{bmatrix}$$

Very useful in least squares problems

Gram matrix

Invertibility

A has linearly independent columns if and only if $A^T A$ is invertible

Gram matrix

Invertibility

A has linearly independent columns if and only if $A^T A$ is invertible

Proof

We show that $Ax = 0 \iff A^T Ax = 0$

Gram matrix
Invertibility

A has linearly independent columns if and only if $A^T A$ is invertible

Proof

We show that $Ax = 0 \iff A^T Ax = 0$

$\Rightarrow$ if $Ax = 0$ then we can write

$$A^T Ax = A^T(Ax) = A^T 0 = 0$$

Gram matrix

Invertibility

A has linearly independent columns if and only if $A^T A$ is invertible

Proof

We show that $Ax = 0 \iff A^T A x = 0$

$\Rightarrow$ if $Ax = 0$ then we can write

$$A^T A x = A^T (Ax) = A^T 0 = 0$$

$\Leftarrow$ if $A^T A x = 0$ then we can write

$$0 = x^T 0 = x^T (A^T A x) = x^T A^T A x = \|Ax\|^2$$

Gram matrix
Invertibility

A has linearly independent columns if and only if $A^T A$ is invertible

Proof

We show that $Ax = 0 \iff A^T A x = 0$

$\Rightarrow$ if $Ax = 0$ then we can write

$$A^T A x = A^T (Ax) = A^T 0 = 0$$

$\Leftarrow$ if $A^T A x = 0$ then we can write

$$0 = x^T 0 = x^T (A^T A x) = x^T A^T A x = \|Ax\|^2$$

which implies that $Ax = 0$ (definition of norm) $\blacksquare$

$(CD)^T = D^T C^T$

Positive (semi)definiteness of Gram matrix

Positive semidefinite (always)

$$x^T A^T A x = (Ax)^T (Ax) = \|Ax\|^2 \geq 0, \qquad \text{for any } n\text{-vector } x$$

Positive (semi)definiteness of Gram matrix

Positive semidefinite (always)

$$x^T A^T A x = (Ax)^T (Ax) = \|Ax\|^2 \geq 0, \qquad \text{for any } n\text{-vector } x$$

Positive definite

$A^T A$ is positive definite if and only if A has linearly independent columns

Positive (semi)definiteness of Gram matrix

Positive semidefinite (always)

$$x^T A^T A x = (Ax)^T (Ax) = \|Ax\|^2 \geq 0, \qquad \text{for any } n\text{-vector } x$$

Positive definite

$A^T A$ is positive definite if and only if A has linearly independent columns

Proof

If the columns of A are linearly independent, then

$Ax \neq 0$ for any $x \neq 0$

Positive (semi)definiteness of Gram matrix

Positive semidefinite (always)

$$x^T A^T A x = (Ax)^T (Ax) = \|Ax\|^2 \geq 0, \qquad \text{for any } n\text{-vector } x$$

Positive definite

$A^T A$ is positive definite if and only if A has linearly independent columns

Proof

If the columns of A are linearly independent, then

$Ax \neq 0$ for any $x \neq 0$

Therefore, $x^T A^T A x = \|Ax\|^2 > 0$ (definition of norm) ■

Solving least squares problems

Main assumption

Least squares problem

$$\text{minimize} \quad \|Ax - b\|_2^2$$

A has **linearly independent columns**

True in most practical examples such as data fitting (next lecture)

Calculus derivation

$\tilde{a}_i^T x - b_i = r_i$

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

Calculus derivation

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

The solution $x^\star$ satisfies

$$\nabla f(x^\star)_k = \frac{\partial f}{\partial x_k}(x^\star) = 0,$$

for $k = 1, \ldots, n$

$y^T z = \sum y_i z_i$

$A_{ik} = (A^T)_{ki}$

Calculus derivation

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

The solution $x^\star$ satisfies

$$\nabla f(x^\star)_k = \frac{\partial f}{\partial x_k}(x^\star) = 0,$$

for $k = 1, \ldots, n$

$$\longrightarrow$$

$$\frac{\partial f}{\partial x_k}(x) = 2 \sum_{i=1}^{m} \left(\sum_{j=1}^{n} A_{ij} x_j - b_i \right) (A_{ik})$$

$$= 2 \sum_{i=1}^{m} (A^T)_{ki} (Ax - b)_i$$

$$= 2 (A^T (Ax - b))_k$$

Calculus derivation in vector form

$$-b^T(Ax) + (Ax)^T(-b) =$$
$$= -b^T A x - x^T A^T b$$
$$= -x^T A^T b - x^T A^T b$$
$$= -2x^T(A^T b)$$

$$f(x) = \|Ax - b\|^2 = (Ax - b)^T(Ax - b) = x^T A^T A x - 2(A^T b)^T x + b^T b$$

Calculus derivation in vector form

$$f(x) = \|Ax - b\|^2 = (Ax-b)^T(Ax-b) = x^T A^T A x - 2(A^T b)^T x + b^T b$$

$$\nabla f(x^\star) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x^\star) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x^\star) \end{bmatrix} = 2A^T A x^\star - 2A^T b = 2A^T(Ax^\star - b) = 0$$

Calculus derivation in vector form

$$f(x) = \|Ax - b\|^2 = (Ax - b)^T(Ax - b) = x^T A^T A x - 2(A^T b)^T x + b^T b$$

$$\nabla f(x^\star) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x^\star) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x^\star) \end{bmatrix} = 2A^T A x^\star - 2A^T b = 2A^T(Ax^\star - b) = 0$$

normal equations

$$(A^T A)x^\star = A^T b$$

Calculus derivation in vector form

$$f(x) = \|Ax - b\|^2 = (Ax - b)^T(Ax - b) = x^T A^T A x - 2(A^T b)^T x + b^T b$$

$$\nabla f(x^\star) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x^\star) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x^\star) \end{bmatrix} = 2A^T A x^\star - 2A^T b = 2A^T(Ax^\star - b) = 0$$

$n \times n$
square
linear system → **normal equations**

$$(A^T A) x^\star = A^T b$$

Optimality

For $x^\star$ such that $A^T A x^\star = A^T b$, we have

Optimality

For $x^\star$ such that $A^T A x^\star = A^T b$, we have

$$\begin{aligned}
\|Ax - b\|^2 &= \|(Ax - Ax^\star) + (Ax^\star - b)\|^2 \\
&= \|A(x - x^\star)\|^2 + \|Ax^\star - b\|^2 + 2(A(x - x^\star))^T (Ax^\star - b) \\
&= \|A(x - x^\star)\|^2 + \|Ax^\star - b\|^2 + 2(x - x^\star)^T A^T (Ax^\star - b) \\
&= \|A(x - x^\star)\|^2 + \|Ax^\star - b\|^2
\end{aligned}$$

Optimality

For $x^\star$ such that $A^T A x^\star = A^T b$, we have

$$
\begin{aligned}
\|Ax - b\|^2 &= \|(Ax - Ax^\star) + (Ax^\star - b)\|^2 \\
&= \|A(x - x^\star)\|^2 + \|Ax^\star - b\|^2 + 2(A(x - x^\star))^T (Ax^\star - b) \\
&= \|A(x - x^\star)\|^2 + \|Ax^\star - b\|^2 + 2(x - x^\star)^T A^T (Ax^\star - b) \\
&= \|A(x - x^\star)\|^2 + \|Ax^\star - b\|^2
\end{aligned}
$$

$(A^T(Ax^\star - b) = 0)$

Optimality

For $x^\star$ such that $A^T A x^\star = A^T b$, we have

$$\begin{aligned} \|Ax - b\|^2 &= \|(Ax - Ax^\star) + (Ax^\star - b)\|^2 \\ &= \|A(x - x^\star)\|^2 + \|Ax^\star - b\|^2 + 2(A(x - x^\star))^T (Ax^\star - b) \\ &= \|A(x - x^\star)\|^2 + \|Ax^\star - b\|^2 + 2(x - x^\star)^T A^T (Ax^\star - b) \\ &= \|A(x - x^\star)\|^2 + \|Ax^\star - b\|^2 \end{aligned}$$

$(A^T(Ax^\star - b) = 0)$

≥ 0 BECAUSE IT'S A NORM

Therefore, for any x, we have

$$\|Ax - b\|^2 \geq \|Ax^\star - b\|^2$$

Optimality

For $x^\star$ such that $A^T A x^\star = A^T b$, we have

$$\|Ax - b\|^2 = \|(Ax - Ax^\star) + (Ax^\star - b)\|^2$$
$$= \|A(x - x^\star)\|^2 + \|Ax^\star - b\|^2 + 2(A(x - x^\star))^T (Ax^\star - b)$$
$$= \|A(x - x^\star)\|^2 + \|Ax^\star - b\|^2 + 2(x - x^\star)^T A^T (Ax^\star - b)$$
$$= \|A(x - x^\star)\|^2 + \|Ax^\star - b\|^2$$

$$(A^T (Ax^\star - b) = 0)$$

Therefore, for any x, we have

$$\|Ax - b\|^2 \geq \|Ax^\star - b\|^2$$

If equality holds, $A(x - x^\star) = 0 \Rightarrow x = x^\star$
since columns of A are linearly independent

Solving normal equations

$$(A^T A)x^\star = A^T b$$

Solving normal equations

$$(A^T A)x^\star = A^T b$$

Inversion

$$x^\star = (A^T A)^{-1} A^T b$$

Solving normal equations

$$(A^T A)x^\star = A^T b$$

Inversion

$$x^\star = (A^T A)^{-1} A^T b$$

→

Pseudo-inverse

$$A^\dagger = (A^T A)^{-1} A^T$$

Solving normal equations

$$(A^T A)x^\star = A^T b$$

Inversion

$$x^\star = (A^T A)^{-1} A^T b$$

→

Pseudo-inverse

$$A^\dagger = (A^T A)^{-1} A^T$$

Factor-solve method

A has linearly independent columns

↓

$A^T A$ is **symmetric positive-definite**

Solving normal equations

$$(A^T A)x^\star = A^T b$$

Inversion

$$x^\star = (A^T A)^{-1} A^T b$$

→

Pseudo-inverse

$$A^\dagger = (A^T A)^{-1} A^T$$

Factor-solve method

A has linearly independent columns

↓

$A^T A$ is **symmetric positive-definite**

→

Cholesky factorization

$$A^T A = LL^T$$

Solving normal equations

$$(A^T A)x^\star = A^T b$$

Inversion → **Pseudo-inverse**

$$x^\star = (A^T A)^{-1} A^T b \longrightarrow A^\dagger = (A^T A)^{-1} A^T$$

Factor-solve method → **Cholesky factorization**

A has linearly independent columns

↓

$A^T A$ is **symmetric positive-definite**

$$\longrightarrow \quad A^T A = L L^T$$

Which method is faster?

Solving normal equations with Cholesky

1. Form linear system $A^T Ax = A^T b$
   - Form $M = A^T A$ ($2mn^2$ flops)
   - Form $q = A^T b$: ($2mn$ flops)

Solving normal equations with Cholesky

1. Form linear system $A^T Ax = A^T b$
    - Form $M = A^T A$ ($2mn^2$ flops)
    - Form $q = A^T b$: ($2mn$ flops)
2. Factor $M = LL^T$ ($(1/3)n^3$ flops)

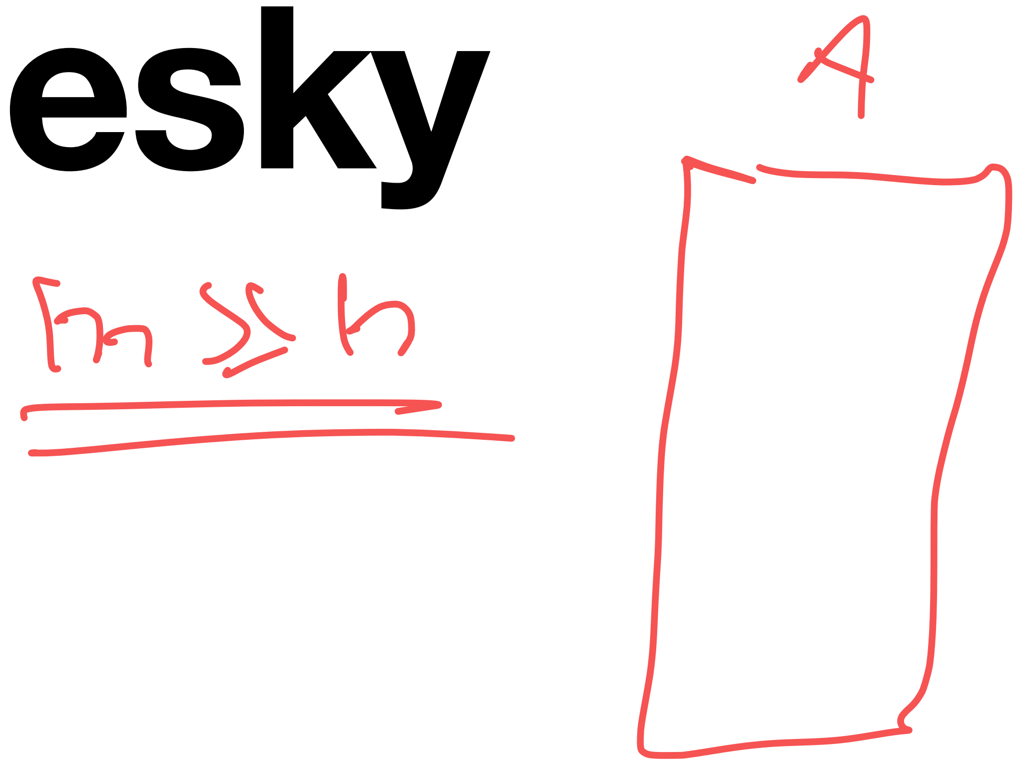
Solving normal equations with Cholesky

1. Form linear system $A^T Ax = A^T b$
   - Form $M = A^T A$ ($2mn^2$ flops)
   - Form $q = A^T b$: ($2mn$ flops)

2. Factor $M = LL^T$ ($(1/3)n^3$ flops)

3. Solve $LL^T x = q$ ($2n^2$ flops)
   (with forward/backward substitution)

Solving normal equations with Cholesky



1. Form linear system $A^T A x = A^T b$
   - Form $M = A^T A$ ($2mn^2$ flops)
   - Form $q = A^T b$: ($2mn$ flops)

2. Factor $M = LL^T$ ($(1/3)n^3$ flops)

3. Solve $LL^T x = q$ ($2n^2$ flops)
   (with forward/backward substitution)

Complexity

- Factor + solve: $2mn^2 + 2mn + (1/3)n^3 + 2n^2 \approx 2mn^2$
- Solve given a new b (prefactored): $2mn + 2n^2 \approx 2mn$

Example

Optimal advertising

m demographic groups we want to advertise to → v^{des} is the m-vector of desired views/impressions

Optimal advertising

m demographic groups
we want to advertise to → v^{des} is the m-vector
of desired views/impressions

n advertising channels
(web publishers, radio, print, etc.) → s is the n-vector
of purchases

Optimal advertising

m demographic groups we want to advertise to → v^{des} is the m-vector of desired views/impressions

n advertising channels (web publishers, radio, print, etc.) → s is the n-vector of purchases

$m \times n$ matrix A gives demographic reach of channels → A_{ij} is the number of views for group i and dollar spent on channel j $(1000/\$)$

Optimal advertising

m demographic groups
we want to advertise to → v^{des} is the m-vector of desired views/impressions

n advertising channels
(web publishers, radio, print, etc.) → s is the n-vector of purchases

$m \times n$ matrix A gives
demographic reach of channels → A_{ij} is the number of views for group i and dollar spent on channel j $(1000/\$)$

Views across demographic groups

$$v = As$$

Optimal advertising

m demographic groups we want to advertise to → v^{des} is the m-vector of desired views/impressions

n advertising channels (web publishers, radio, print, etc.) → s is the n-vector of purchases

$m \times n$ matrix A gives demographic reach of channels → A_{ij} is the number of views for group i and dollar spent on channel j $(1000/\$)$

Views across demographic groups

$$v = As$$

Goal

minimize $\|As - v^{\text{des}}\|^2$

Optimal advertising

Results

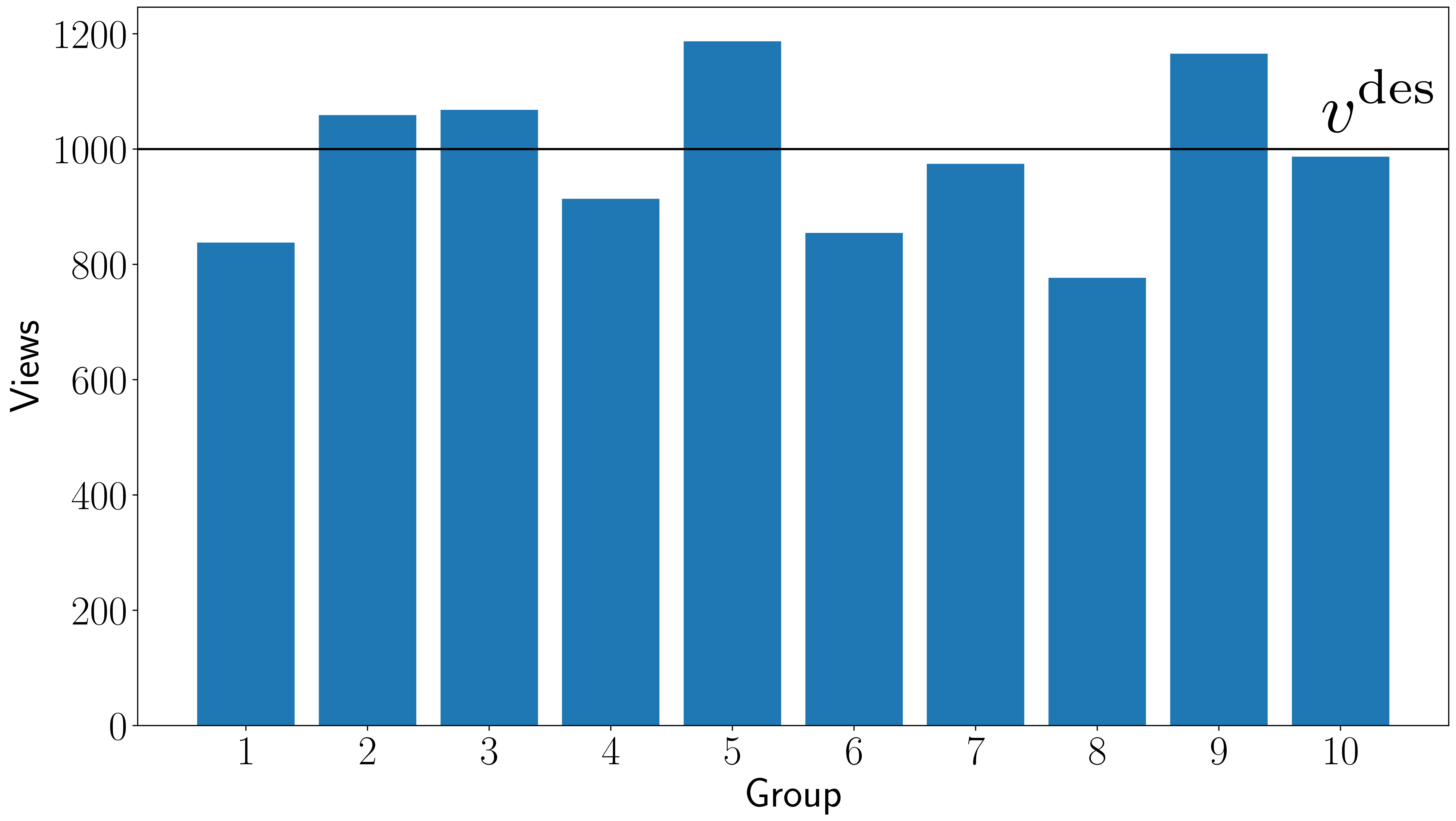
$m = 10$ groups, $n = 3$ channels

desired views vector $v^{\text{des}} = (10^3)\mathbf{1}$

minimize $\quad \|As - v^{\text{des}}\|^2$

↓

optimal spending $s^\star = (62, 100, 1443)$



Optimal advertising

Reusing factorization on large example

$m = 100,000$ groups, $n = 5,000$ channels

minimize $\quad \|As - v^{\text{des}}\|^2$

Optimal advertising

Reusing factorization on large example

$m = 100,000$ groups, $n = 5,000$ channels

minimize $\quad \|As - v^{\mathrm{des}}\|^2$

Pseudoinverse

Time: 263 sec

Optimal advertising

Reusing factorization on large example

$m = 100,000$ groups, $n = 5,000$ channels

minimize $\|As - v^{\text{des}}\|^2$

First solve

desired views $v^{\text{des},1} = (10^3)\mathbf{1}$

1. Form linear system $Mx = q$
   where $M = A^T A, q = A^T b$
2. Factor $M = LL^T$
3. Solve $LL^T x = q$

Pseudoinverse

Time: 263 sec

Optimal advertising

Reusing factorization on large example

$m = 100,000$ groups, $n = 5,000$ channels

minimize $\quad \|As - v^{\text{des}}\|^2$

First solve

desired views $v^{\text{des},1} = (10^3)\mathbf{1}$

1. Form linear system $Mx = q$
   where $M = A^T A, q = A^T b$
2. Factor $M = LL^T$
3. Solve $LL^T x = q$

Pseudoinverse

Time: 263 sec

Complexity

$2mn^2$

Time: 9 sec

Optimal advertising

Reusing factorization on large example

$m = 100,000$ groups, $n = 5,000$ channels

minimize $\quad \|As - v^{\mathrm{des}}\|^2$

Pseudoinverse

Time: 263 sec

First solve

desired views $v^{\mathrm{des},1} = (10^3)\mathbf{1}$

1. Form linear system $Mx = q$
   where $M = A^TA, q = A^Tb$
2. Factor $M = LL^T$
3. Solve $LL^Tx = q$

Complexity

$2mn^2$

Time: 9 sec

Second solve

desired views $v^{\mathrm{des},2} = 500\mathbf{1}$

1. Form $q = A^Tb$
2. Solve $LL^Tx = q$

Optimal advertising

Reusing factorization on large example

$m = 100,000$ groups, $n = 5,000$ channels

minimize $\quad \|As - v^{\mathrm{des}}\|^2$

Pseudoinverse

Time: 263 sec

First solve

desired views $v^{\mathrm{des},1} = (10^3)\mathbf{1}$

1. Form linear system $Mx = q$ where $M = A^T A, q = A^T b$
2. Factor $M = LL^T$
3. Solve $LL^T x = q$

Complexity

$2mn^2$

Time: 9 sec

Second solve

desired views $v^{\mathrm{des},2} = 500\mathbf{1}$

1. Form $q = A^T b$
2. Solve $LL^T x = q$

Complexity

$2mn$

Time: 0.37 sec

Least squares

Today, we learned to:

- **Define and recognize** least squares problems
- **Solve** least squares problems using Cholesky factorization
- **Understand** the benefits of reusing factorizations

References

- S. Boyd, L. Vandenberghe: Introduction to Applied Linear Algebra – Vectors, Matrices, and Least Squares
  - Chapter 12: least squares

Next lecture

- Least squares and data fitting