QR Factorization and Singular Value Decomposition

COS 323

Last time

- Solving non-linear least squares
 - Newton, Gauss-Newton, Levenberg-Marquardt methods
 - Intro to logistic regresion
- Dealing with outliers and bad data:
 - Robust regression, least absolute deviation, and iteratively re-weighted least-squares
- Practical considerations
- Solving with Excel and Matlab

Today

- How do we solve least-squares...
 - without incurring condition-squaring effect of normal equations (A^TAx = A^Tb)
 - when A is singular, "fat", or otherwise poorly-specified?
- QR Factorization
 - Householder method
- Singular Value Decomposition
- Total least squares
- Practical notes

Review: Condition Number

- Cond(A) is function of A
- Cond(A) >= 1, bigger is bad
- Measures how change in input is propogated to change in output

$$\frac{\|\Delta x\|}{\|x\|} \leq cond(A) \frac{\|\Delta A\|}{\|A\|}$$

E.g., if cond(A) = 451 then can lose log(451)=
 2.65 digits of accuracy in x, compared to precision of A

Normal Equations are Bad

$$\frac{\|\Delta x\|}{\|x\|} \leq cond(A) \frac{\|\Delta A\|}{\|A\|}$$

- Normal equations involves solving A^TAx = A^Tb
- $\operatorname{cond}(A^T A) = [\operatorname{cond}(A)]^2$
- E.g., if cond(A) = 451 then can lose log(451²) = 5.3 digits of accuracy, compared to precision of A

QR Decomposition

What if we didn't have to use A^TA?

• Suppose we are "lucky":

#	#	•••	#]	[#]	
0	#		#	#	
0	0	•	•	#	[م]
0	• • •	0	# x ≘	≝ #	$\begin{bmatrix} R \\ O \end{bmatrix} x = b$
0	•••	•••	0	#	$\lfloor O \rfloor$
•			•	#	
0	0	•••	0	[#]	

• Upper triangular matrices are nice!

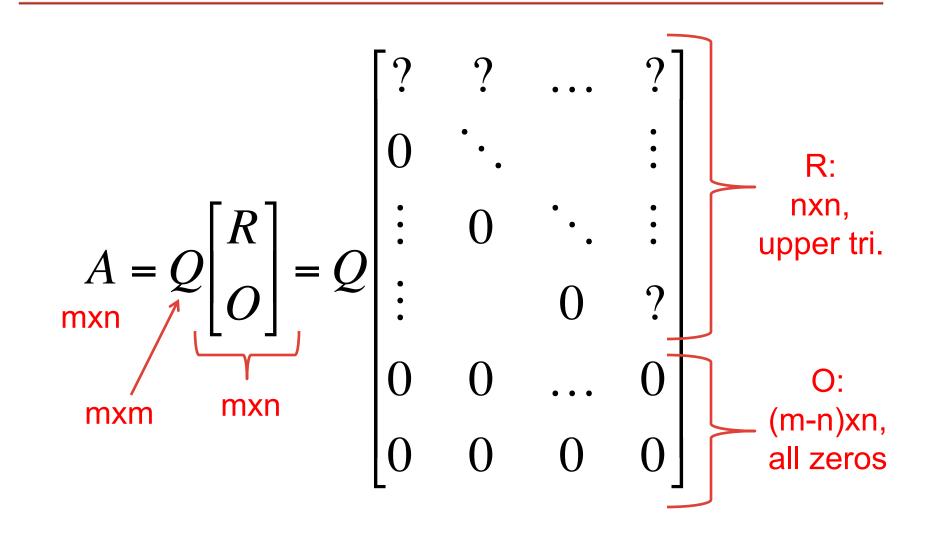
How to make A upper-triangular?

- Gaussian elimination?
 - Applying elimination yields MAx = Mb
 - Want to find x s.t. minimizes ||Mb-MAx||₂
 - Problem: ||Mv||₂!= ||v||₂ (i.e., M might "stretch" a vector v)
 - Another problem: M may stretch different vectors differently
 - i.e., M does not preserve Euclidean norm
 - i.e., x that minimizes ||Mb-MAx|| may not be same x that minimizes Ax=b

QR Factorization

- Can't usually find R such $A = \begin{bmatrix} R \\ O \end{bmatrix}$ • Can find Q, R such that $A = Q \begin{bmatrix} R \\ O \end{bmatrix}$, so $\begin{bmatrix} R \\ O \end{bmatrix} x = Q^T b$
- If Q orthogonal, doesn't change leastsquares solution
 - $Q^{T}Q=I$, columns of Q are orthonormal
 - i.e., Q preserves Euclidean norm: $||Qv||_2 = ||v||_2$

Goal of QR



Reformulating Least Squares using QR

$$\begin{aligned} \|r\|_{2}^{2} &= \|b - Ax\|_{2}^{2} \\ &= \left\|b - Q\begin{bmatrix}R\\O\end{bmatrix}x\right\|_{2}^{2} = \left\|Q^{T}b - Q^{T}Q\begin{bmatrix}R\\O\end{bmatrix}x\right\|_{2}^{2} \quad \text{because } A = Q\begin{bmatrix}R\\O\end{bmatrix} \\ &= \left\|Q^{T}b - \begin{bmatrix}R\\O\end{bmatrix}x\right\|_{2}^{2} \quad \text{because } Q \text{ is orthogonal } (Q^{T}Q=I) \\ &= \|c_{1} - Rx + c_{2}\|_{2}^{2} \quad \text{if we call } Q^{T}b = \begin{bmatrix}c_{1}\\c_{2}\end{bmatrix} \\ &= \|c_{1} - Rx\|_{2}^{2} + \|c_{2}\|_{2}^{2} \\ &= \|c_{2}\|_{2}^{2} \quad \text{if we choose x such that } Rx=c_{1} \end{aligned}$$

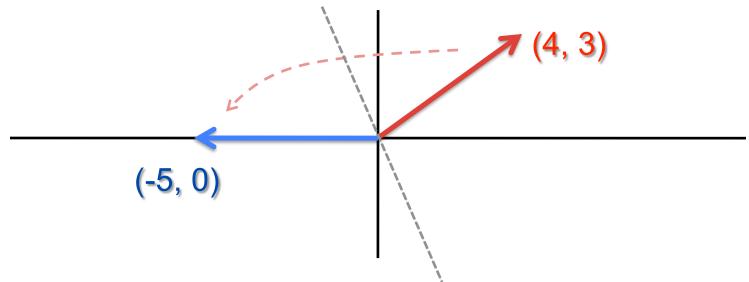
Householder Method for Computing QR Decomposition Orthogonalization for Factorization

$$A = Q\begin{bmatrix} R\\ O\end{bmatrix}$$

- Rough idea:
 - For each i-th column of A, "zero out" rows i+1 and lower
 - Accomplish this by multiplying A with an orthogonal matrix H_i
 - Equivalently, apply an orthogonal transformation to the i-th column (e.g., rotation, reflection)
 - Q becomes product $H_1^*...^*H_{n_i}R$ contains zero-ed out columns

Householder Transformation

- Accomplishes the critical sub-step of factorization:
 - Given any vector (e.g., a column of A), reflect it so that its last p elements become 0.
 - Reflection **preserves length** (Euclidean norm)



Computing Householder

• if *a* is the k-th column:

$$a = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$
$$v = \begin{bmatrix} 0 \\ a_2 \end{bmatrix} - \alpha e_k \text{ where } \alpha = -sign(a_k) ||a_2||_2$$

apply H to a and columns to the right :

$$Hu = u - \left(2\frac{v^T u}{v^T v}\right)v$$
 (*with some shortcuts - see p124)

Exercise: Show H is orthogonal (H^TH=I)

Outcome of Householder

$$H_{n} \dots H_{1}A = \begin{bmatrix} R \\ O \end{bmatrix}$$

where $Q^{T} = H_{n} \dots H_{1}$
so $Q = H_{1} \dots H_{n}$
so $A = Q \begin{bmatrix} R \\ O \end{bmatrix}$

Review: Least Squares using QR

$$\begin{aligned} \|r\|_{2}^{2} &= \|b - Ax\|_{2}^{2} \\ &= \left\|b - Q\begin{bmatrix}R\\O\end{bmatrix}x\right\|_{2}^{2} = \left\|Q^{T}b - Q^{T}Q\begin{bmatrix}R\\O\end{bmatrix}x\right\|_{2}^{2} \quad \text{because } A = Q\begin{bmatrix}R\\O\end{bmatrix} \\ &= \left\|Q^{T}b - \begin{bmatrix}R\\O\end{bmatrix}x\right\|_{2}^{2} \quad \text{because } Q \text{ is orthogonal } (Q^{T}Q=I) \\ &= \|c_{1} - Rx + c_{2}\|_{2}^{2} \quad \text{if we call } Q^{T}b = \begin{bmatrix}c_{1}\\c_{2}\end{bmatrix} \\ &= \|c_{1} - Rx\|_{2}^{2} + \|c_{2}\|_{2}^{2} \\ &= \|c_{2}\|_{2}^{2} \quad \text{if we choose x such that } Rx=c_{1} \end{aligned}$$

Using Householder

- Iteratively compute H₁, H₂, ... H_n and apply to A to get R
 - also apply to b to get

$$Q^T b = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

(

Solve for Rx=c₁ using back-substitution

Alternative Orthogonalization Methods

- Givens:
 - Don't reflect; rotate instead
 - Introduces zeroes into A one at a time
 - More complicated implementation than Householder
 - Useful when matrix is sparse
- Gram-Schmidt
 - Iteratively express each new column vector as a linear combination of previous columns, plus some (normalized) orthogonal component
 - Conceptually nice, but suffers from subtractive cancellation

Singular Value Decomposition

Motivation #1

 Diagonal matrices are even nicer than triangular ones:

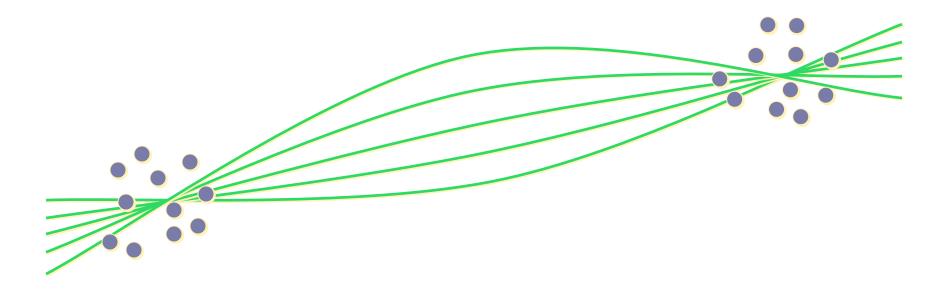
$$\begin{bmatrix} \# & 0 & 0 & 0 \\ 0 & \# & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \# \\ 0 & \cdots & 0 & \# \\ \vdots & & \vdots & & \# \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

Motivation #2

- What if you have fewer data points than parameters in your function?
 - i.e., A is "fat"
 - Intuitively, can't do standard least squares
 - Recall that solution takes the form $A^TAx = A^Tb$
 - When A has more columns than rows,
 A^TA is singular: can't take its inverse, etc.

Motivation #3

- What if your data poorly constrains the function?
- Example: fitting to y=ax²+bx+c



Underconstrained Least Squares

 Problem: if problem very close to singular, roundoff error can have a huge effect

– Even on "well-determined" values!

- Can detect this:
 - Uncertainty proportional to covariance $C = (A^T A)^{-1}$
 - In other words, unstable if $A^T A$ has small values
 - More precisely, care if $x^T(A^TA)x$ is small for any x
- Idea: if part of solution unstable, set answer to 0
 - Avoid corrupting good parts of answer

Singular Value Decomposition (SVD)

- Handy mathematical technique that has application to many problems
- Given any *m*×*n* matrix **A**, algorithm to find matrices **U**, **V**, and **W** such that

 $\mathbf{A} = \mathbf{U} \mathbf{W} \mathbf{V}^{\mathsf{T}}$

- **U** is *m*×*n* and **orthonormal**
- W is *n*×*n* and **diagonal**
- V is *n*×*n* and **orthonormal**

SVD

$$\begin{pmatrix} \mathbf{A} \\ \mathbf{A} \end{pmatrix} = \begin{pmatrix} \mathbf{U} \\ \mathbf{U} \end{pmatrix} \begin{pmatrix} w_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & w_n \end{pmatrix} \begin{pmatrix} \mathbf{V} \\ \mathbf{V} \end{pmatrix}^{\mathrm{T}}$$

Treat as black box: code widely available
 In Matlab: [U,W,V]=svd(A,0)

SVD

- The *w_i* are called the singular values of **A**
- If **A** is singular, some of the *w_i* will be 0
- In general $rank(\mathbf{A}) = number of nonzero w_i$
- SVD is mostly unique (up to permutation of singular values, or if some w_i are equal)

SVD and Inverses

- Why is SVD so useful?
- Application #1: inverses
- $A^{-1} = (V^T)^{-1} W^{-1} U^{-1} = V W^{-1} U^T$
 - Using fact that inverse = transpose
 for orthogonal matrices
 - Since W is diagonal, W⁻¹ also diagonal with reciprocals of entries of W

SVD and the Pseudoinverse

• $A^{-1} = (V^T)^{-1} W^{-1} U^{-1} = V W^{-1} U^T$

- This fails when some *w_i* are 0
 - It's *supposed* to fail singular matrix
 - Happens when rectangular A is rank deficient
- Pseudoinverse: if $w_i=0$, set $1/w_i$ to 0 (!)
 - "Closest" matrix to inverse
 - Defined for all (even non-square, singular, etc.) matrices
 - Equal to $(\mathbf{A}^{\mathsf{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathsf{T}}$ if $\mathbf{A}^{\mathsf{T}}\mathbf{A}$ invertible

SVD and Condition Number

 Singular values used to compute Euclidean (spectral) norm for a matrix:

$$\operatorname{cond}(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$$

SVD and Least Squares

- Solving **Ax=b** by least squares:
- $A^TAx = A^Tb \rightarrow x = (A^TA)^{-1}A^Tb$
- Replace with A⁺: x = A⁺b
- Compute pseudoinverse using SVD
 - Lets you see if data is singular (< n nonzero singular values)
 - Even if not singular, condition number tells you how stable the solution will be
 - Set $1/w_i$ to 0 if w_i is small (even if not exactly 0)

SVD and Matrix Similarity

- One common definition for the norm of a matrix is the Frobenius norm: $\|\mathbf{A}\|_{\mathrm{F}} = \sum_{i} \sum_{j} a_{ij}^{2}$
- Frobenius norm can be computed from SVD $\|\mathbf{A}\|_{\mathrm{F}} = \sum_{i} w_{i}^{2}$
- Euclidean (spectral) norm can also be computed: $\|\mathbf{A}\|_2 = \{\max[\lambda] : \lambda \in \sigma(\mathbf{A})\}$
- So changes to a matrix can be evaluated by looking at changes to singular values

SVD and Matrix Similarity

- Suppose you want to find best rank-k approximation to A
- Answer: set all but the largest k singular values to zero
- Can form compact representation by eliminating columns of U and V corresponding to zeroed w_i

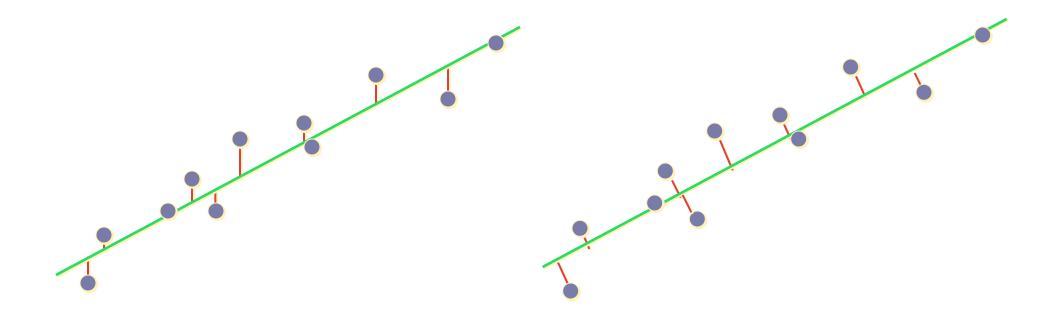
SVD and Eigenvectors

• Let $A = UWV^T$, and let x_i be *i*th column of V

• Consider
$$\mathbf{A}^{\mathsf{T}}\mathbf{A} x_{i}$$
:
 $\mathbf{A}^{\mathsf{T}}\mathbf{A} x_{i} = \mathbf{V}\mathbf{W}^{\mathsf{T}}\mathbf{U}^{\mathsf{T}}\mathbf{U}\mathbf{W}\mathbf{V}^{\mathsf{T}} x_{i} = \mathbf{V}\mathbf{W}^{2}\mathbf{V}^{\mathsf{T}} x_{i} = \mathbf{V}\mathbf{W}^{2}\begin{pmatrix}0\\\vdots\\1\\\vdots\\0\end{pmatrix} = \mathbf{V}\begin{pmatrix}0\\\vdots\\\\w_{i}^{2}\\\vdots\\0\end{pmatrix} = w_{i}^{2}x_{i}$

 So elements of W are sqrt(eigenvalues) and columns of V are eigenvectors of A^TA

- One final least squares application
- Fitting a line: vertical vs. perpendicular error



• Distance from point to line:

$$d_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} \cdot \vec{n} - a$$

where n is normal vector to line, a is a constant

• Minimize:

$$\chi^{2} = \sum_{i} d_{i}^{2} = \sum_{i} \left[\begin{pmatrix} x_{i} \\ y_{i} \end{pmatrix} \cdot \vec{n} - a \right]^{2}$$

• First, let's pretend we know n, solve for a

$$\chi^{2} = \sum_{i} \left[\begin{pmatrix} x_{i} \\ y_{i} \end{pmatrix} \cdot \vec{n} - a \right]$$
$$a = \frac{1}{m} \sum_{i} \begin{pmatrix} x_{i} \\ y_{i} \end{pmatrix} \cdot \vec{n}$$

2

• Then

$$d_{i} = \begin{pmatrix} x_{i} \\ y_{i} \end{pmatrix} \cdot \vec{n} - a = \begin{pmatrix} x_{i} - \frac{\Sigma x_{i}}{m} \\ y_{i} - \frac{\Sigma y_{i}}{m} \end{pmatrix} \cdot \vec{n}$$

• So, let's define

$$\begin{pmatrix} \widetilde{x}_i \\ \widetilde{y}_i \end{pmatrix} = \begin{pmatrix} x_i - \frac{\Sigma x_i}{m} \\ y_i - \frac{\Sigma y_i}{m} \end{pmatrix}$$

and minimize

$$\sum_{i} \left[\begin{pmatrix} \widetilde{x}_i \\ \widetilde{y}_i \end{pmatrix} \cdot \vec{n} \right]^2$$

• Write as linear system

$$\begin{pmatrix} \widetilde{x}_1 & \widetilde{y}_1 \\ \widetilde{x}_2 & \widetilde{y}_2 \\ \widetilde{x}_3 & \widetilde{y}_3 \\ \vdots \end{pmatrix} \begin{pmatrix} n_x \\ n_y \end{pmatrix} = \vec{0}$$

- Have An=0
 - Problem: lots of n are solutions, including n=0
 - Standard least squares will, in fact, return n=0

Constrained Optimization

- Solution: constrain n to be unit length
- So, try to minimize $|An|^2$ subject to $|n|^2=1$ $||A\vec{n}||^2 = (A\vec{n})^T (A\vec{n}) = \vec{n}^T A^T A \vec{n}$
- Expand in eigenvectors e_i of A^TA:

$$\vec{n} = \mu_1 \mathbf{e}_1 + \mu_2 \mathbf{e}_2$$
$$\vec{n}^{\mathrm{T}} \left(\mathbf{A}^{\mathrm{T}} \mathbf{A} \right) \vec{n} = \lambda_1 \mu_1^2 + \lambda_2 \mu_2^2$$
$$\left\| \vec{n} \right\|^2 = \mu_1^2 + \mu_2^2$$

where the λ_i are eigenvalues of A^TA

Constrained Optimization

- To minimize $\lambda_1 \mu_1^2 + \lambda_2 \mu_2^2$ subject to $\mu_1^2 + \mu_2^2 = 1$ set $\mu_{\min} = 1$, all other $\mu_i = 0$
- That is, n is eigenvector of A^TA with the smallest corresponding eigenvalue

Comparison of Least Squares Methods

- Normal equations (A^TAx = A^Tb)
 - O(mn²) (using Cholesky)
 - $\operatorname{cond}(A^{T}A) = [\operatorname{cond}(A)]^{2}$
 - Cholesky fails if cond(A)~1/sqrt(machine epsilon)

Householder

- Usually best orthogonalization method
- $O(mn^2 n^3/3)$ operations

- Relative error is best possible for least squares
- Breaks if cond(A) ~ 1/ (machine eps)

• SVD

- Expensive: mn² + n³ with bad constant factor
- Can handle rankdeficiency, near-singularity
- Handy for many different things

Matlab functions

- qr: explicit QR factorization
- svd
- A\b: ('\' operator)
 - Performs least-squares if A is m-by-n
 - Uses QR decomposition
- pinv: pseudoinverse
- rank: Uses SVD to compute rank of a matrix