Engs 104 Lecture 10

W 2010

Optimization

George Cybenko

Today

Algorithms for linear algebra computations

Matlab examples

(LDU, QR, SVD, Eigenvalues)

Last time, we saw how to solve

$$\min_x \|Ax-b\|_2^2 \qquad \text{(linear least squares)}$$

using two techniques:

A is $m \times n$

1. Normal equations

$$A^T A x = A^T b \qquad n \times n$$

2. QR decomposition

$$A = QR \qquad Q^T Q = I, \; R \text{ upper } \nabla$$

$$Rx = Q^T b$$

Simple observation:

Row reduction operation on matrix A is matrix multiplization of A by a specially structured matrix on the left.

Eg add $\alpha_{ij} \times$ row j to row i

$$\begin{bmatrix} 1 & & \alpha_{1i} & & \\ & \ddots & \alpha_{2i} & & 0 \\ & & 1 & & \\ & & 1+\alpha_{ii} & & \\ 0 & & & \ddots & \\ & & \alpha_{ni} & & 1 \end{bmatrix} A = \Big(\underbrace{I}_{\text{Identity}} + \underbrace{\alpha_i}_{\text{coefficients}} \overbrace{e_i}^{i^{th}\text{ column of } I}{}^{T}\Big) A$$

<u>Example</u>

$$\begin{bmatrix} 1 & & & \\ \alpha_{21} & 1 & & 0 \\ \vdots & & \ddots & \\ \alpha_{n1} & 0 & & 1 \end{bmatrix} A = \begin{bmatrix} a_{11} & a_{12} & & a_{1n} \\ 0 & a_{22}^{(2)} & \cdots & \cdots \\ \vdots & & & \\ 0 & a_{n2}^{(2)} & \cdots & \cdots \end{bmatrix}$$

$$\boxed{\alpha_{j1} = -\frac{a_{j1}}{a_{11}}} \qquad = \left(I + \alpha_1 \cdot e_1^T\right) A = A^{(1)}$$

Row reduce second column, using second row.

$$\alpha_{j2} = \frac{-a_{j2}^{(2)}}{a_{22}^{(2)}}$$

$$A^{(2)} = \left(I + \alpha_2 e_2^T\right)\left(I + \alpha_1 e_1^T\right) A^{(0)}$$

$$= \left(I + \alpha_2 e_2^T\right) A^{(1)}$$

Proceed with row reduction

Use $i+1$ row of $A^{(i)}$ to introduce 0's below diagonal of the $i+1$ column.

$$A^{(n-1)} = \left(I + \alpha_{n-1} e_{n-1}^T\right) \cdots\cdots \left(I + \alpha_1 e_1^T\right) A^{(0)}$$

$$= \begin{bmatrix} a_{11}^{(n-1)} & & \sim \\ & \ddots & \\ 0 & & a_{nn}^{(n-1)} \end{bmatrix} = \begin{bmatrix} a_{11}^{(n-1)} & & 0 \\ & \ddots & \\ 0 & & a_{nn}^{(n-1)} \end{bmatrix} U$$

E104, L10

$$A^{(n-1)} = \prod_{j=1}^{n-1} \left(I + \alpha_j e_j^T\right) A = D \cdot U$$

$$= F_{n-1} F_{n-2} \quad F_1 A$$

$$\left(F_{n-1} \cdots F_1\right)^{-1} = F_1^{-1} F_2^{-1} \cdots F_{n-1}^{-1}$$

$$= \left(I + \alpha_1 e_1^T\right)^{-1} \left(I + \alpha_2 e_2^T\right)^{-1} \left(\cdots \right) \left(I + \alpha_{n-1} e_{n-1}^T\right)^{-1}$$

$$= \left(I - \alpha_1 e_1^T\right)\left(I - \alpha_2 e_2^T\right) \cdots \left(I - \alpha_{n-1} e_{n-1}^T\right)$$

$$= I - \alpha_1 e_1^T - \alpha_2 e_2^T \cdots - \alpha_{n-1} e_{n-1}^T$$

~~upper tri~~

because

$$\left(I + \alpha_i e_i^T\right)\left(I - \alpha_i e_i^T\right)$$

$$= I + \alpha_i e_i^T - \alpha_i e_i^T - \alpha_i \underbrace{e_i^T \alpha_i}_{0} e_i^T$$

$$= I$$

and

$$\alpha_i \underbrace{e_i^T \alpha_j}_{=\,0} e_j^T = 0 \quad \text{for} \quad i < j$$

Gaussian elimination (row reduction)

$$A = L \cdot D \cdot U$$

where L is lower ◺ triangular with 1's on diagonal,

D is diagonal

U is upper ∇ with 1's on diagonal

LDU factorization!

This works if $a_{ii}^{(i-1)} \neq 0$ or not small.

Fact if $A^T = A$ is positive definite, $a_{ii}^{(i-1)}$ is always the largest element in row & column i (and is >0).

Otherwise, may have to permute rows of A etc

Matlab

$$[L, U] = \text{lu}(A)$$

produces L, U lower/upper ∇
with $L \cdot U = A$.

Row/column reduction of A
implicitly produces a
factorization of A!

$$A x = b$$

If $A = L \cdot U$ then

$$L \cdot U x = b$$

Solve $L \cdot y = b$ (back solve)

then $U x = y$ (back solve)

gives a solution to $Ax = b$.

Fact Suppose $A^T A = B$

so B is symmetric & pos def.

$B = LDU$ ($= LU$ matlab difference)

not 1's on diagonal)

$B^T = U^T D^T L^T = B = LDU$

$$\underbrace{L^{-1} U^T D^T}_{\text{lower } \Delta} = \underbrace{D U (L^T)^{-1}}_{\text{upper } \Delta}$$

$L^{-1} = (U^T)^{-1}$ so $L = U^T$

So for B pos def & symmetric

$$B = L \cdot D \cdot L^T$$

$$D = \begin{bmatrix} d_{11} & & 0 \\ & \ddots & \\ 0 & & d_{nn} \end{bmatrix} \qquad d_{ii} > 0$$

$$B = \left(L \cdot D^{1/2}\right)\left(D^{1/2} L^T\right)$$

$$= C^T \cdot C$$

Cholesky Decomposition

Matlab chol(A)

Let's do some examples.

Now suppose

$$B = A^T A = C^T C$$

so

$$(C^T)^{-1} A^T A C^{-1} = I$$

$$(AC^{-1})^T = (C^T)^{-1} A^T = (C^{-1})^T A^T$$

So let

$$Q = AC^{-1}, \quad Q^T Q = I$$

Q is orthogonal!

$$A = QC$$

Q orthogonal

C upper Δ

$$A^T A = C^T C = C^T (Q^T Q) C$$

Can we compute C without computing $B = A^T A$?

<u>Yes</u> this is the "QR" decomposition.

Methods for computing Q,R

1. Gram-Schmidt
2. Givens Rotations
3. Householder Transformations

Gram-Schmidt

$x, y \in E^n$ two vectors

let $\tilde{y} = y - \left(\frac{x^T y}{\|x\|^2}\right) x$

$x^T \tilde{y} = x^T \cdot y - \frac{x^T x}{\|x\|_2^2} x^T y = 0$

x is first column of A
y is another column of A

$$A^{(1)} = A \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \cdots \\ 0 & 1 & & \\ \vdots & & \ddots & \\ & 0 & & 1 \\ 0 & & & \end{bmatrix}$$

where $\alpha_{1i} = -\dfrac{a_1^T a_i}{\|a_1\|_2^2}$

a_i = i^{th} column of A

Repeat to get $A^{(2)}$ by subtracting appropriate ~~colums~~ multiples of second column from columns 3, 4, ..., n

Resulting $A^{(n-1)}$ has orthogonal columns &

$$Q = A^{(n-1)} = A \cdot \underbrace{U_1 \cdot U_2 \cdots U_n}_{\text{upper } \Delta}$$

E104, L10

QR factorization is a very useful tool!

examples

1. solve $Ax=b$ ✓

2. solve $\min_x \|Ax-b\|_2^2$ ✓

3. compute eigenvalues of A ✓

QR algorithm for finding eigenvalues of A $(n \times n)$

$$A^{(0)} = Q^{(0)} R^{(0)}$$

$$A^{(1)} = R^{(0)} Q^{(0)} = Q^{(1)} R^{(1)}$$

$$A^{(2)} = R^{(1)} Q^{(1)} = Q^{(2)} R^{(2)}$$

$A^{(k)}$ converges & can read off eigenvalues!

In practice this is tooo expensive!

If A is symmetric, can "reduce" A as follows

$$A = Q^T T Q \qquad Q \text{ orthogonal}$$

T is "~~tridiagonal~~"
"tridiagonal"

Doing QR iteration on T is cheap.
n ops per iteration

E104, L10

A is $m \times n$ general matrix

$A^T A = B$ is $n \times n$ pos def & symmetric

$B = Q^T D Q$ $\quad$ Q are orthogonal eigenvectors of B

D are eigenvalues > 0

So $\left(D^{-1/2} Q A^T\right) A Q^T D^{-1/2}$ = Identity

So $AQ^T D^{-1/2} = W$ is orthogonal

$$A = W D^{1/2} Q$$

diagonal ≥ 0

orthogonal

is Singular Value Decomposition.