Welcome to Ma221! (Mar 3)

Solving low rank LS using QR with pivoting

$A^{m\times n}$ rank $r<n$, first r cols independent

$$\Rightarrow A = QR \qquad R = \begin{array}{c c} & \begin{array}{cc} r & \end{array} \\ r & \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix} \end{array} \qquad R_{11} \text{ full rank}$$

$$Q = \begin{array}{c} \begin{array}{cc} r & \end{array} \\ \begin{bmatrix} Q_1, & Q_2 \end{bmatrix} \end{array}$$

$$\underset{x}{\operatorname{argmin}} \|Ax-b\|_2 \Rightarrow x = \begin{array}{c c} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} & \begin{array}{c} r \\ n-r \end{array} \end{array}$$

$$x_1 = R_{11}^{-1} \cdot Q_1^T b - R_{11}^{-1} \cdot R_{12} x_2 \quad \text{for any } x_2$$

how to pick x_2, say to minimize $\|x\|_2$

Ex $= A = \begin{bmatrix} e & 1 \\ 0 & 0 \end{bmatrix}$ e tiny $\quad R_{11} = e \quad R_{12} = 1$

$$\Rightarrow x = \begin{bmatrix} (b_1 - x_2)/e \\ x_2 \end{bmatrix}$$

e tiny $\Rightarrow$ very sensitive to small changes in b, x_2

$$AP = \begin{bmatrix} 1 & e \\ 0 & 0 \end{bmatrix} \Rightarrow x = \begin{bmatrix} b_1 - e x_2 \\ x_2 \end{bmatrix}$$

insensitive to changes in b, x_2

What would a "perfect" R factor look like?

Compare $\begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix}$ to $\begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}$

Def: Rank Revealing QR factorization (RRQR for short) is $A \cdot P = QR$

P permutation chosen so

R_{11} $k \times k$

(1) R_{22} is "small", ideally $\|R_{22}\|_2 = O(\sigma_{k+1})$

R_{22} "contains" smallest $n-k$ sing. vals.

(2) R_{11} is "large" ideally $\sigma_{min}(R_{11})$ not much smaller than σ_k

if in addition, we have

(3) $\|R_{11}^{-1} R_{12}\|$ not "too large"

then $AP = QR$ called strong RRQR

Thm (informal) if (1), (2), (3) hold

$$\sigma_i(A) \geq \sigma_i(R_{11}) \geq \frac{\sigma_i(A)}{\sqrt{1 + \|R_{11}^{-1} R_{12}\|_2^2}}$$

for $i = 1..k$

$$\sigma_i(A) \leq \sigma_{max}(R_{22}) \cdot \sqrt{1 + \|R_{11}^{-1} R_{12}\|_2^2}$$

$i = k+1, \ldots, n$

Leading k columns of $A \cdot P$ contain most information in range(A)

$$AP = [\overset{r}{A_1}, \overset{n-r}{A_2}] = QR = [\overset{r}{Q_1}, \overset{n-r}{Q_2}] \begin{array}{cc} & \begin{array}{cc} r & n-r \end{array} \\ r & \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \end{array}$$

$$\approx [Q_1, Q_2] \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix}$$

$$= Q_1 R_{11} [I, R_{11}^{-1} R_{12}] = A_1 [I, R_{11}^{-1} R_{12}]$$

How to compute P?

Algorithm: QR with column pivoting
QRCP for short

Analogous to partial pivoting:
simple greedy alg
often works, can fail up to a factor 2^n

First step: pick longest col of A, move to front

take one step of QR

among remaining columns pick one with longest orthogonal component to first column

repeat

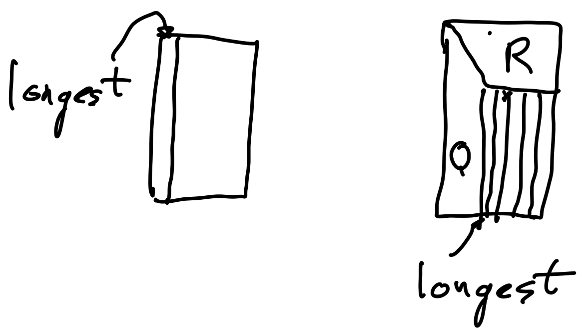
for $i = 1$ to $\min(m-1, n)$ or
R_{22} small enough

choose largest column (in norm)
in trailing matrix

$$\operatorname*{argmax}_{j \ge i} \| A(i:m, j) \|_2$$

if $i \neq j$, swap cols i & j

multiply $A(i:m, i:n)$ by
Householder matrix to
zero out $A(i+1:m, i)$



if stop after k steps because
$A(k+1:m, k+1:n)$ has no column
bigger than some threshold, then
cost $= 4 \, m \cdot n \cdot k$ versus $O(mn^2)$
for full QR, cheaper if $k < n$

Need to compute column norms
carefully to get this cost;
straightforward would cost mn^2

versus $m \cdot n$

Drawbacks: can fail in rare cases
maximizes communication

in LAPACK: geqpf
Matlab: $[Q, R, P] = qr(A)$

Matlab examples (code in typed notes)

How to fix failure mode, communication

Gu/Eisenstat Strong RRQR Algorithm
- avoids failures to pivot correctly
- more complicated pivoting,
  exchange a column in R_{11} and one not in R_{11} by maximizing $\det(R_{11})$
- guarantees strong RRQR, ie $\|R_{11}^{-1} R_{12}\|$ bounded
- more expensive, still $4mnk$ if stop after k steps

Avoiding Communication in RRQR

algorithm so far touches trailing matrix at each step $\Rightarrow O(mn^2)$ data movement instead of $O(mn^2/\sqrt{\text{cache size}})$

First attempt to fix, in LAPACK:

use matmul to update trailing submatrix, but only reduces communication 2x

Need different pivoting strategy:

Analogous to TSLU:

choose blocksize b

BestColumnsSoFar = (1:b)

for k = b+1 to n−b+1 step b ... assume b | n

form m × 2b matrix from columns in BestColumnsSoFar and columns k : k+b−1

choose best b columns from these 2b columns

update BestColumnsSoFar

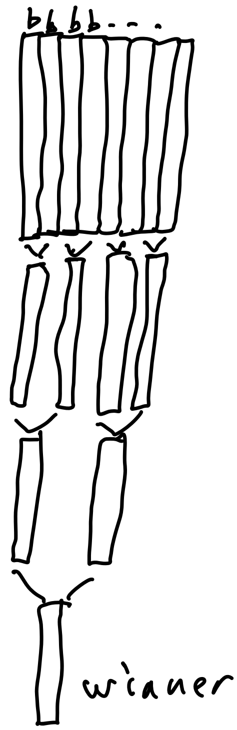
How to choose best b from 2b?

Factor $A_{2b} = QR =$ using TSQR

choose best b columns from R

(cheap because R just 2b × 2b)

Called "Tournament Pivoting"



Low Rank Factorizations without Orthogonal Factors

Why avoid orthogonal factor?
explainability and sparsity in data science

E: A: rows represent people
cols " characteristics
age, weight etc

Suppose a column of Q is
$.2 \cdot \text{age} - .3 \cdot \text{height} + .1 \cdot \text{income}$

Hard to interprest what this means
as predictor of another column
like "been treated for disease"

Instead want to approximate other
columns by linear combinations of
as few other columns as possible

Def: CUR decomposition of A:

C = subset of k columns of A

R = subset of k rows of A

U = $k \times k$ matrix

where $\|A - CUR\|$ is "small"
close to lower bound σ_{k+1}

lots of algorithms (webpage has more)

(1) Choose C: perform QR with some
kind of column pivoting to pick
"k most linearly independent columns"
$J = \{j_1, j_2, \ldots j_k\}$ be indices
of selected columns

(2) Perform GEPP, or TSLU, on C
to pick "k most linearly independent
rows of C"

$I = \{i_1, i_2, \dots, i_k\}$, R = these rows of A

Still need U

(1) HW 3.12 U that minimizes $\|A - CUR\|_F$ is $U = C^+ A R^+$

(2) Cheaper: choose U so that CUR matches A in rows I and columns J

$$C(I, 1:k) = R(1:k, J) = A(I, J)$$

$$\Rightarrow U = (A(I,J))^{-1}$$

· Randomized Linear Algebra

LS and SVD on $A^{m \times n}$ $\quad m \gg n$

Let Q be a random $m \times k$ orthogonal matrix $\quad k \ll n$

Approximate A by $Q(\underbrace{Q^T A}_{k \times n}) = (QQ^T)A$

Cost? $Q^T A$ costs $2mnk$

only 2x cheaper than QRCP for k steps

$\Rightarrow$ need cheap structured Q to make multiplication cheaper

Best result so far for $\underset{x}{\operatorname{argmin}} \|Ax-b\|_2$ is $O(nnz(A))$ for A sparse

Examples in low dimension of why random projection OK

Ex: $x \in \mathbb{R}^2$, $q \in \mathbb{R}^2$ random unit vector

$q = \begin{bmatrix} \sin t \\ \cos t \end{bmatrix}$ t uniform on $[0, 2\pi)$

how well does $|x^T q|$ approximate $\|x\|_2^2$?

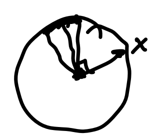
What is distribution of

$$|x^T q|^2 = \|x\|_2^2 \cdot \cos^2 \angle(x,q)$$

Easy to see $\angle(x,q)$ also uniform on $[0,2\pi)$

$E(|x^T q|^2) = .5 \|x\|_2^2$

What is Prob $(|x^T q|^2)$ underestimates $\|x\|_2^2$ by a factor $e << 1$?



$\text{Prob}(|\cos(t)|^2 < e)$

$\approx 2\sqrt{e}/\pi$ small

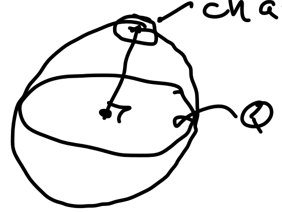
Ex: $x \in \mathbb{R}^3$, Q random plane

i.e. random 3×2 orthogonal matrix

$x^T Q$ = projection of x onto plane

How well does $\|x^T Q\|_2$ approx $\|x\|_2$?

$\|x^T Q\|_2^2 < e\|x\|_2^2$ depends on x being nearly parallel to perpendicular of Q



chance that x lies in this little circle?

proportional to e versus $\sqrt{e}$ before

Johnson–Lindenstrauss (JL) Lemma

$0 < \varepsilon < 1$, $x_1, \dots, x_n$ any n vectors in $\mathbb{R}^m$

$k \geq 8 \cdot \ln(n)/\varepsilon^2$

Let F be random $k \times m$ orthogonal matrix, multiplied by $\sqrt{m/k}$

Then with probability $\geq \frac{1}{n}$

forall $1 \leq i, j \leq n \quad i \neq j$

$$1 - \varepsilon \leq \frac{\|F(x_i - x_j)\|_2^2}{\|x_i - x_j\|_2^2} \leq 1 + \varepsilon$$

Probability $\frac{1}{n}$ seems small but being positive means F exists (original goal of JL)

Proof: think of F fixed

$x = x_i - x_j$ as random

$F = \begin{bmatrix} I \\ 0 \end{bmatrix}$, each entry of x i.i.d. $N(0,1)$

problem just reasoning about sums of squares of $N(0,1)$ variables

(Dasgupta + Gupta on web page)