Welcome back to Ma221! Lecture 23, Oct 16

Randomized Low Rank Factorizations

Given $A^{m\times n}$ $m \gg n$, target rank $k \ll n$.
Usually don't know k accurately, in practice pick $k+p$, p extra columns for "safety", get better error bounds as function of p

Given $A^{m\times n}$, choose $F^{n\times(k+p)}$, random tall+skinny, form $A\cdot F$ to get randomized linear combinations of columns of A, i.e. sample column space of A

1) choose random $n\times(k+p)$ F
2) form $Y = A\cdot F$ $\quad m\times(k+p)$
3) factor $Y = QR$, Q also spans same sample of range of A as Y
4) $B = Q^T A$ $\quad (k+p)\times n$

Answer: approximate A by $Q\cdot B = Q(Q^T A)$
QQ^T = orthogonal projection, rank $k+p$ onto space approximating column space of A

Best possible Q: first $k+p$ left singular vectors of $A = U_A \Sigma_A V_A^T$

$$QQ^T A = U(1:m, 1:k+p) \cdot \Sigma_A(1:k+p, 1:k+p) \cdot (V(1:n, 1:k+p))^T$$

$= k+p$ truncated SVD

$$\|A - QQ^T A\|_2 = \sigma_{k+p+1}$$

Our goal is just error proportional to σ_{k+1}

Thm: If each $F(i,j)$ is i.i.d. $N(0,1)$

$$E(\|A - QQ^T A\|_2) = \sigma_{k+1}\left(1 + \frac{4\sqrt{k+p}}{p-1} \min(m,n)\right)$$

$$\text{Prob}\left(\|A - QQ^T A\| \le \sigma_{k+1}\left(1 + 11 \cdot \sqrt{k+p}\sqrt{\min(m,n)}\right)\right) \ge 1 - \frac{6}{p^p}$$

$p = 6 \Rightarrow$ prob $\sim .9999$

When is Randomized low rank approximation cheaper than QRCP, which costs $O(m \cdot n \cdot (k+p))$?

If A sparse, steps 2), 3) 4) cost

(2) $Y = A \cdot F$ costs $2 \cdot nnz(A) \cdot (k+p)$
(3) $Y = QR$ costs $2m(k+p)^2$
(4) $B = Q^T A$ costs $2 \cdot nnz(A)(k+p)$

each of these can be much smaller than $QRCP = O(m \cdot n \cdot (k+p))$

When does 3) dominate, vs. 2) & 4)?
Depends on how dense A is: if A has at least $k+p$ nonzeros per row, 2) and 4) dominate 3).

Chap 7 has more algorithms for SVD of sparse matrices

What about dense A?
2): $A \cdot F$ costs $2mn(k+p)$, same as QRCP

need cheaper F
If we use SRTT or SRHT for F
cost $(A \cdot F)$ drops to $O(m \cdot n \cdot \log m)$
Factoring $Y = QR$ still $O(m(k+p)^2)$
better than QRCP if $k+p \ll n$
But $B = Q^T A$ still costs $O(m \cdot n \cdot (k+p))$
like QRCP, need new idea

Randomized Low-Rank Factorization via Row Extraction

(1) choose random $n \times (k+p)$ F

(2) $Y = A \cdot F$ $\quad m \times (k+p)$

(3) $Y = QR$

(4) Choose $k+p$ "most linearly independent" $k+p$ rows of Q:

write $\underset{\text{permutation}}{P} \cdot Q = \begin{array}{c} k+p \\ \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \end{array} \begin{array}{l} \\ k+p \\ m-(k+p) \end{array}$

P from GEPP, or tournament pivoting, or QRCP on Q^T

(5) $X = P \cdot Q \cdot Q_1^{-1} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} Q_1^{-1} = \begin{bmatrix} I \\ Q_2 Q_1^{-1} \end{bmatrix}$

we expect $\|X\| = O(1)$, true if we QRCP with strong rank revealing factorization (Gu & Eisenstat)

(6) $PA = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \begin{array}{l} k+p \\ m-(k+p) \end{array}$ $\qquad A \sim P^T X \cdot A_1$

$$= P^T \begin{bmatrix} I \\ Q_2 Q_1^{-1} \end{bmatrix} A_1$$

$$= P^T \begin{bmatrix} A_1 \\ Q_2 Q_1^{-1} A_1 \end{bmatrix}$$

Cost on a dense A

(2) $O(m \cdot n \cdot \log n)$ or $O(m \cdot n \cdot \log k+p)$
if use SRFT or SRHT

(3) $Y = QR \qquad 2m(k+p)^2$

(4) GEPP on Q or QRCP on Q^T
$\cdot\ 2m(k+p)^2$

(5) $Q_2 \cdot Q_1^{-1} \qquad O(m(k+p)^2)$

much better than previous cost
$O(m \cdot n \cdot (k+p))$ when $k+p \ll n$

Thm $\|A - P^T X \cdot A_1\|_2 \leq (1 + \|X\|_2) \|A - QQ^T A\|_2$

proof: assume $P = I$

$$\|A - XA_1\|_2 = \|A - QQ^TA + QQ^TA - XA_1\|_2$$
$$\leq \|A - QQ^TA\|_2 + \|QQ^TA - XA_1\|_2$$
$$= \|A - QQ^TA\|_2 + \|XQ_1 \cdot Q^TA - XA_1\|_2$$
$$\leq \|A - QQ^TA\|_2 + \|X\|_2 \cdot \|Q_1 \cdot Q^TA - A_1\|_2$$
$$\leq \|A - QQ^TA\|_2 + \|X\|_2 \cdot \|\underbrace{\begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}}_{Q} Q^TA - \underbrace{\begin{bmatrix} A_1 \\ A_2 \end{bmatrix}}_{A}\|_2$$
$$= \|A - QQ^TA\|_2 + \|X\|_2 \cdot \|QQ^TA - A\|_2$$

Chap 4: Eigenvalue Problems

Goals:

- Canonical Forms (recall Jordan, why Schur form better)
- Variations on eigenproblems (not just one square matrix)
- Perturbation theory (can I trust my answer?)
- Algorithms (for a single nonsymmetric A, $A = A^T$ is Chap 5)

Webpage: Templates for solution of Algebraic Eigenproblems

Recall Definitions

Def: $p(\lambda) = \det(A^{n \times n} - \lambda I)$ is characteristic polynomial

n roots are eigenvalues (evals)

Def: if λ eval $\exists$ nonzero null vector x of $A - \lambda I$: $(A - \lambda I)x = 0$ or $Ax = \lambda x$

x right eigenvector (evec)

Analagously $\exists\, y^*$ s.t. $y^*(A - \lambda I) = 0$

$y^* A = \lambda y^*$ $\quad$ y^* left evec

Def: S nonsingular, $B = SAS^{-1}$

S similarity transform

A and B similar

Lemma: A and B similar $\Rightarrow$

have same evals,

evecs related by multiplying by S.

pf: $Ax = \lambda x \Rightarrow SAS^{-1}Sx = \lambda Sx$

$= (B)Sx = \lambda Sx$

$y^H A = \lambda y^H \Rightarrow$

$(y^H S^{-1}) SAS^{-1} = \lambda y^H S^{-1}$

$(y^H S^{-1}) B = \lambda y^H S^{-1}$

Goal: transform A to a similar and simpler B whose evals and evecs "easy" to compute

Ideal: B diagonal $\Rightarrow$

evals = diagonals of B

evecs $= \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

Lemma: if $Ax_i = \lambda_i x_i$ $i = 1:n$

and $S = [x_1, \ldots, x_n]$ nonsingular

i.e. $\exists$ n linearly independent evecs

Then $A = S \cdot \operatorname{diag}\begin{bmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_n \end{bmatrix} \cdot S^{-1}$

But can't always diagonalize A:
recall Jordan form, which
we will <u>not</u> compute