# Welcome to Math 221! Lecture 15, Fall 24

## Randomized Low Rank Factorizations

Goal: Beat QRCP (and SVD)

Given $A^{m \times n}$, $m > n$, and target rank $k \ll n$, usually don't know $k$ accurately, so in practice use $k+p$, $p = \#$ extra columns in sampling matrix, to get good rank $k$ approx

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

1) choose random $n \times (k+p)$ $F$
2) form $Y = A \cdot F$ $\quad m \times (k+p)$
   expect $Y$ to accurately sample column space of $A$
3) $Y = QR$, columns of $Q$ also sample column space of $A$
4) $B = Q^T A$

Answer: approximate $A$ by $Q \cdot B = Q \cdot Q^T \cdot A$

$Q \cdot Q^T$ = orthogonal projection onto a space approximating column space of $A$

If we compute SVD of $B = U \Sigma V^T$

then $Q \cdot B = \underbrace{(Q \cdot U)}_{\text{orthogonal}} \cdot \Sigma \cdot V^T$ = approx SVD of $A$

Best possible $Q$: first $k+p$ left singular vectors, for $\sigma_1, \ldots, \sigma_{k+p}$

if $A = U_A \cdot \Sigma_A \cdot V_A^T$, then

$$QQ^TA = U_A(1:m, 1:k+p)\Sigma_A(1:k+p, 1:k+p)(V_A(1:n, k+p))^T$$

= $k+p$ truncated SVD of $A$

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

Our goal is just to get error proportional to $\sigma_{k+1}$

Thm: if $F(i,j)$ is i.i.d $N(0,1)$, then

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

$$\text{prob}\left( \|A - QQ^T A\|_2 \le \left(1 + 11 \cdot \sqrt{k+p}\,\sqrt{\min(m,n)}\right)\sigma_{k+1} \right) \ge 1 - \frac{6}{p^p}$$

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

When is Randomized Low Rank Approximation cheaper than QRCP, which costs $O(m \cdot n \cdot (k+p))$?

If $A$ sparse, last 3 steps of alg cost:

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

each of which can be much smaller than cost of QRCP: $O(m \cdot n (k+p))$

since $nnz(A) \ll m \cdot n$;

$m(k+p)^2 \ll m \cdot n \cdot k+p$ since $k+p \ll n$

Does (3) dominate, or (2) + (4)?

if $A$ has at least $k+p$ nonzeros per row

then $nnz(A) \ge m \cdot (k+p)$ and (2)+(4) dominate

(Chap 7 has more approximate SVD algs, not randomized, can combine with randomization for more speedups)

What about dense $A$?

If we use explicit dense $F$

$\mathrm{cost}(A\cdot F) = 2\cdot m\cdot n\cdot(k+p)$, comparable to QRCP

Factoring $Y=QR$ still costs $O(m(k+p)^2)$

potentially much less than QRCP

But $B=Q^T A$ still costs $O(m\cdot n\cdot(k+p))$

comparable to QRCP

# 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) Find "most linearly independent" $k+p$ rows of $Q$:

write $P\cdot Q = \begin{array}{cc} k+p & \\ \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} & \begin{array}{l} k+p \\ m-(k+p) \end{array} \end{array}$, can use GEPP or QRCP on $Q^T$

permutation

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

we expect $\|X\| \sim O(1)$

(6) use $P$ to select same rows of $A$

$$PA = \begin{array}{c} n \\ \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \end{array} \begin{array}{l} \\ k+p \\ m-(k+p) \end{array}$$

, final approx of $A$ is $A \sim P^T X \cdot A_1$

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

Costs for a dense $A$:

(2) $O(m \cdot n \cdot \log n)$ or $O(m \cdot n \cdot \log(k+p))$
using SRTT or SRHT

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

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

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

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

If GEPP or QRCP in step (4) work well,
expect $\|X\| = \|Q_2 \cdot Q_1^{-1}\| = O(1)$

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

Proof: Assume $P = I$ for simplicity

$$\| A - X \cdot A_1 \|_2 = \| A - QQ^T A + QQ^T A - X A_1 \|_2$$
$$\leq \| A - QQ^T A \|_2 + \| QQ^T A - X A_1 \|_2$$
$$= \| A - QQ^T A \|_2 + \| X \cdot Q_1 \cdot Q^T A - X A_1 \|_2$$
$$= \| A - QQ^T A \|_2 + \| X \|_2 \cdot \| Q_1 Q^T A - A_1 \|_2$$
$$\leq \| A - QQ^T A \|_2 + \| X \|_2 \cdot \left\| \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} Q^T A - \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \right\|_2$$
$$= \| A - QQ^T A \| + \| X \|_2 \cdot \| QQ^T A - A \|_2$$
$$= (1 + \| X \|_2) \| A - QQ^T A \|_2 \qquad \text{QED}$$

---

Eigen value Problems:

Goals:

Canonical Forms (recall Jordan form, why Schur form better)

Variations on eigen problems (not just one square matrix)

Perturbation theory (can I trust my answer?)

Algorithms (for a single, nonsymmetric A, Chap 5 for $A = A^T$)

Webpage: Templates for Solution of Algebraic Eigenvalue Problems

Recall Definitions:

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

n roots $\rightarrow$ eigenvalues

Def: if $\lambda$ eigenvalue, $\exists$ nonzero null vector $x$ of $A - \lambda I \Rightarrow$

$Ax - \lambda I x = 0 \Rightarrow Ax = \lambda x$

$x$ right eigenvector (evec for short)

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

$\Rightarrow y^H A = \lambda y^H$, $y$ left evec

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

$S$ similarity transform

$A$ and $B$ are similar

Lemma: $A$ and $B$ similar $\Rightarrow$ have same evals, and evecs related by multiplying by $S$ or $S^{-1}$

pf: $Ax = \lambda x$ iff $\underbrace{SAS^{-1}}_{B} Sx = \lambda Sx$

iff $B(Sx) = \lambda (Sx)$

$y^H A = \lambda y^H$ iff $y^H S^{-1} \underbrace{S A S^{-1}}_{B} = \lambda y^H S^{-1}$

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

Goal: Transform $A$ to a similar and "simpler" $B$ whose evals and evecs are "easy" to compute

Simplest: $B$ diagonal

$\Rightarrow$ evals are $B(i,i)$, evecs $= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ $i$th entry

Lemma: if $A x_i = \lambda_i x_i$ for $i = 1 \cdots n$

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

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

then $A = S \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} S^{-1} = S \cdot \Lambda \cdot S^{-1}$

Conversely if $A = S \Lambda S^{-1}$, $\Lambda$ diagonal then columns of $S$ are evecs

proof: $A = S \Lambda S^{-1}$ iff $AS = S\Lambda$

iff $A S[:,i] = \underset{\text{evec}}{S[:,i]}\; \underset{\text{eval}}{\lambda_i}$ $\forall i$ QED

But can't always diagonalize A for 2 reasons

(1) may be mathematically impossible
(recall Jordan form)

(2) may be numerically unstable
even if it exists (when evals close)

Recall Jordan Form: For any $A$
$\exists$ similarity $SAS^{-1} = J = \mathrm{diag}(J_1, \ldots, J_k)$
where each $J_i = \begin{bmatrix} \lambda_i & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda_i \end{bmatrix}$

Up to permuting order of $J_i$ this is unique;
different $J_i$ can have equal $\lambda_i$, eg $A = I$

Only one right or left evec per block

$$\begin{bmatrix} \lambda & 1 & & 0 \\ & \lambda & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \lambda \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

can't diagonalize, need $n$ independent evecs

How sensitive is this? $e \ll 1$

(1) $\begin{bmatrix} 1 & 0 \\ 0 & 1+e \end{bmatrix} : \left(1, \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right), \left(1+e, \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right)$

(2) $\begin{bmatrix} 1 & e \\ e & 1 \end{bmatrix} : \left(1+e, \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right), \left(1-e, \begin{bmatrix} 1 \\ -1 \end{bmatrix}\right)$

evecs rotated $45°$

(3) $\begin{bmatrix} 1 & e \\ 0 & 1+e^2 \end{bmatrix}$ : $\left(1, \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right)$, $\left(1+e^2, \begin{bmatrix} 1 \\ e \end{bmatrix}\right)$

evecs nearly parallel

(4) $\begin{bmatrix} 1 & e \\ 0 & 1 \end{bmatrix}$, $\left(1, \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right)$ just one evec

(5) $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, (1, anything), (1, anything)

When evals (nearly) identical,
Jordan form very ill conditioned

Our goal: backward stability
exact evals, evecs of $A+E$ $\quad \frac{\|E\|}{\|A\|} = O(\varepsilon)$

---

Backward Stable approach to eigenproblem

Recall chap 3: Multiplying by multiple orthogonal matrices backward stable

$$fl(Q_k(\cdots(Q_2(Q_1A)))) = Q(A+E) \qquad \frac{\|E\|}{\|A\|} = O(\varepsilon)$$
$$QQ^T = I$$

Apply twice to get orthogonal similarity

$$fl(Q_k(\cdots(Q_2(Q_1AQ_1^T)Q_2^T)\cdots)Q_k^T) = Q(A+F)Q^T$$
$$QQ^T = I \qquad \frac{\|F\|}{\|A\|} = O(\varepsilon)$$

If we restrict similarity $S$ to be orthogonal, how close to Jordan form can we get?

Thm (Schur canonical Form)

Given an $n \times n$ $A$, $\exists$ unitary $Q$, $QQ^H = I$ s.t. $Q^H A Q = T =$ upper triangular

with evals $T(i,i)$ which can be complex if $A$ real, appear in any order

Computing evecs of $T$: just triangular solve

$$\begin{array}{c} \\ i-1 \\ 1 \\ n-i \end{array} \begin{array}{c} \begin{array}{ccc} i-1 & 1 & n-i \end{array} \\ \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ 0 & T(i,i) & T_{23} \\ 0 & 0 & T_{33} \end{bmatrix} \end{array} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = T(i,i) \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \begin{array}{c} i-1 \\ 1 \\ n-i \end{array}$$

$$T_{11} \cdot x_1 + T_{12} x_2 + T_{13} \cdot x_3 = T(i,i) \cdot x_1 \quad (1)$$

$$T(i,i) \cdot x_2 + T_{23} \cdot x_3 = T(i,i) \cdot x_2 \quad (2)$$

$$T_{33} \cdot x_3 = T(i,i) x_3 \quad (3)$$

if $T(i,i)$ unique (3): $\underbrace{(T_{33} - T(i,i)I)}_{\text{nonsingular}} x_3 = 0$

$\Rightarrow x_3 = 0$

(2): $T(i,i) \cdot x_2 = T(i,i) x_2$ , $x_2 = 1$

(1): $T_{11} x_1 + T_{12} = T(i,i) x_1$

$$\left.(\underbrace{T_{11} - T(i,i)I}_{\text{triangular, nonsingular}})\, x_1 = -T_{12} \quad\right\}\ \text{triangular solve}$$

What does Matlab do with

$$[V, D] = \mathrm{eig}\left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\right) \quad ?$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad V = \begin{bmatrix} 1 & 1 \\ 0 & \underbrace{2^{-969}}_{\approx 2\cdot 10^{-292}} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & \frac{\text{underflow}}{\varepsilon} \end{bmatrix}$$