Welcome to Ma221! (Mar 8)

Approximate $x \in \mathbb{R}^m$ by Fx $\quad F^{k \times m}$ $\quad k \ll m$

$$\|Fx\| \approx \|x\|$$

Last time: used JL Lemma: F was a scaled random orthogonal matrix

What other choices of F are there?
(see bcourses for RandLAPACK design document)

To construct random orthogonal Q

$A^{m \times k}$ each entry i.i.d. $N(0,1)$

$A = QR$ $\quad Q$ random orthogonal

$F = \sqrt{\frac{m}{k}}\, Q$

expensive: QR costs $O(mk^2)$

Some applications ok where
each $F(i,j)$ is i.i.d. $N(0,1)$
delay QR until later in algorithm

but $F \cdot x$ costs $m \cdot k$ when x dense,
still too much in some cases

Subsampled randomized trig transform (SRTT)

trig = FFT

Fx will cost $O(m \log m)$ or even $O(m \log k)$

$F = R \cdot FFT \cdot D$ so $Fx = R(FFT(Dx))$

D = $m \times m$ diagonal matrix where $D(i,i)$ uniformly distributed on unit circle in $\mathbb{C}$

FFT = Fast Fourier Transform

$R^{k \times m}$, a random subset of k rows of $m \times m$ I

Real Case SRHT = Subsampled randomized Hadamard transform

FFT replaced by H = Hadamard Trans.

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix}$$

$$H_{2^n} = \begin{bmatrix} H_{2^{n-1}} & H_{2^{n-1}} \\ H_{2^{n-1}} & -H_{2^{n-1}} \end{bmatrix}$$

Intuition for why $\|Fx\| \approx \|x\|$:

FFT, or H, "mixes" entries of x so sampling k of them (multiplying by R) good enough

When x is sparse, want it faster

Goal: cost of $Fx = O(nnz(x))$

$F = S \cdot D \qquad Fx = S(Dx)$

D $m \times m$ diagonal $D_{ii} = \pm 1$

S is $k \times m$, each column is a randomly selected column of I_k

$y = S \cdot Dx$

for each nonzero x_i

pick random y_j

$y_j = y_j \pm x_i$

Called Randomized Sparse Embedding

F not as statistically "strong" as previous F's, so need larger k

---

Apply these choices of F to LS

"sketch and solve": project onto smaller problem, done

"sketch and iterate": use randomization to build a "preconditioner", iterate (Chap 6)

$$x_{true} = \arg\min_x \|Ax-b\|_2 \quad A^{m\times n} \quad m>n$$

$$x_{approx} = \arg\min_x \|F(Ax-b)\|_2$$

Use J.L for $F^{k\times m}$ (random orthogonal)

choose $k = n\log n/\varepsilon^2$ rows

in order to get

$$\|Ax_{approx}-b\|_2 \le (1+\varepsilon)\|Ax_{true}-b\|_2$$

No bound on $\|x_{approx}-x_{true}\|_2$

Cost: if F dense, computing $F\cdot A$

using dense matmul cost $O(k\cdot m\cdot n)$

$= O(m\cdot n^2\log n/\varepsilon^2)$

bigger than doing $A=QR$ $\quad (O(mn^2))$

Use Cheaper F:

SRTT (SRFFT or SRHT)

with dense A

FA costs $O(n\cdot m\log m)$

FA has size $k\times n$ so solving

smaller LS problem $\arg\min_x \|(FA)x-Fb\|_2$

costs $O(kn^2) = O(n^3\log n/\varepsilon^2)$

$\Rightarrow$ Total cost $= O(m\cdot n\log n + n^3\log n/\varepsilon^2)$

potentially much cheaper than QR, $O(mn^2)$ when $m \gg n$

May be ok if ε not too small
(if ε small, need to "sketch and iterate", see Chap 6)

Sparse LS: goal: cost $O(nnz(A))$ + "lower order terms"

see papers by Clarkson + Woodruff
Meng + Mahoney

F = Randomized Sparse Embedding

$$k = O\left(\left(\frac{n}{\varepsilon}\right)^2 \cdot \log^6\left(\frac{n}{\varepsilon}\right)\right)$$

Forming FA and Fb costs $nnz(A)$ and $nnz(b)$

Since $k = \Omega(n^2)$ much larger than SRTT for which $k = O(n)$

If we solved $\arg\min_x \|(FA)x - Fb\|_2$ using dense QR would cost

$$O(kn^2) = O(n^4 \cdot \log^2(\tfrac{n}{\varepsilon})/\varepsilon^2)$$

much larger than SRTT

Trick: use randomization again to solve new LS problem: (use SRTT)

Thm: With probability $\geq \frac{2}{3}$

$$\|Ax_{approx} - b\|_2 \leq (1+\epsilon)\|Ax_{true} - b\|_2$$

To make probability of success larger, run s times, pick result with smallest residual, probability of success $= 1 - \frac{1}{3^s}$

Randomized Low Rank Factorizations

$A^{m\times n}$, assume target rank $k \ll n$, usually don't know k accurately, so in practice pick $k+p$, p extra columns in F, oversampling for "safety"

Given $A^{m\times n}$, choose F, 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 $Y = A \cdot F$, $m \times (k+p)$

expect Y to accurately span column space of A

3) factor $Y = QR$, Q also spans column space of A accurately

4) $B = Q^T A$ $\quad (k+p) \times n$

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

QQ^T = orthogonal projection onto space approximating column space of A

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

then $QB = (QU) \Sigma V^T$

= approximate SVD 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 of A

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

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

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

then $E(\|A-QQ^TA\|_2)$

$$\le \left(1+\frac{4\sqrt{k+p}}{p-1}\sqrt{\min(m,n)}\right)\sigma_{k+1}$$

$\text{Prob}(\|A-QQ^TA\|_2 \le$

$$\left(1+11\cdot\sqrt{k+p}\sqrt{\min(m,n)}\right)\sigma_{k+1})$$

$$\ge 1-\frac{6}{p^p}$$

$p=6 \Rightarrow \text{prob} \approx .9999$

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 algorithm cost:

(2) $Y=A\cdot F$ costs $2\,\text{nnz}(A)\cdot(k+p)$

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

(4) $B=Q^TA$ costs $2\,\text{nnz}(A)\,(k+p)$

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

Whether cost of (3) dominates

(2) and (4) depends on how dense A is: if A has at least $k+p$ nonzeroes per row, (2) and (4) dominate (3)

(Chap 7 has more approximate algs for SVD, can combine with randomization to accelerate)

This was all for sparse case — what about dense A?

If we use explicit dense F cost $(A \cdot F) = 2(mn(k+p))$, comparable to QRCP

If we use SRTT for F cost(AF) drops to $O(m \cdot n \cdot \log m)$, much faster than 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:

Need another 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) Find "most linearly independant" $k+p$ rows of Q:

write $PQ = \begin{array}{cc} k+p & \\ \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} & \begin{array}{l} k+p \\ m-(k+p) \end{array} \end{array}$

P permutation, can use GEPP, or TSLU on Q or QRCP on Q^T

(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 $\|X\| \sim O(1)$
(true if QRCP provides a strong rank revealing factorization)

(6) $PA = \begin{array}{cc} n & \\ \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} & \begin{array}{l} k+p \\ m-(k+p) \end{array} \end{array}$ result is

$A \sim P^T X \cdot A_1$

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

(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
$O(m\cdot n\cdot(k+p))$ when $k+p \ll n$

If QRCP or GEPP works well in (5)
i.e. $\|X\|=O(1)$, then approximation
nearly as good as QQ^TA

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

proof: assume $P=I$

$$\|A-XA_1\|_2=\|A-QQ^TA+QQ^TA-XA_1\|_2$$
$$\le\|A-QQ^TA\|_2+\|QQ^TA-XA_1\|_2$$
$$=\|A-QQ^TA\|_2+\|XQ_1\cdot Q^TA-XA_1\|_2$$
$$\le\|A-QQ^TA\|_2+\|X\|_2\cdot\|Q_1Q^TA-A_1\|_2$$
$$\le\|A-QQ^TA\|_2+\|X\|_2\cdot\left\|\begin{bmatrix}Q_1\\Q_2\end{bmatrix}Q^TA-\begin{bmatrix}A_1\\A_2\end{bmatrix}\right\|_2$$

$$= \|A - QQ^T A\|_2 + \|X\|_2 \, \|QQ^T A - A\|_2$$

$$= (1 + \|X\|_2)(\|A - QQ^T A\|_2)$$