Welcome to Ma221! (Mar 8)

Approximate $x \in \mathbb{R}^n$ by $F x \sim_{k \times m} \|F x\| \approx \|x\|

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

What other choices of $F$ are there? (See courses for RandLAPACK design document)

To construct random orthogonal $Q$

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

$A = QR$ \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 $F x$ costs $m \cdot k$ when $x$ dense, still too much in some cases
Sub sampled randomized trig transform (SRTT)

\[ \text{trig} = \text{FFT} \]

\[ F_x \text{ will cost } O(m \log m) \text{ or even } O(m \log k) \]

\[ F = R \cdot \text{FFT} \cdot D \quad \text{so} \quad F_x = R(\text{FFT}(Dx)) \]

\[ D = \text{mxm diagonal matrix} \]

where \( D(i,j) \) uniformly distributed on unit circle in \( \mathbb{C} \)

\[ \text{FFT} = \text{Fast Fourier Transform} \]

\[ R_{k \times m} \text{, a random subsample of} \]

\[ k \text{ rows of } m \times m \text{ I} \]

Real Case SRTT =

Sub sampled randomized Hadamard transform

\[ \text{FFT replaced by } H = \text{Hadamard Transform} \]

\[ H_2 = \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix}, \quad H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} \]

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

Intuition for why \( \|F_x\| \approx \|x\| \):

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 \( Fx = O(n \text{nnz}(x)) \)

\[ F = SD \quad Fx = S(Dx) \]

\( D \) m×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 \( y_i \)

\( y_{ij} = y_i \pm x_i \)

*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 (Chapter 6)
$$x_{\text{true}} = \arg \min_x \|Ax-b\|_2$$

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

Use JL for $F^{\text{random orthogonal}}$
choose $k = n \log n / \epsilon^2$ rows
in order to get
$$\|Ax_{\text{approx}} - b\|_2 \leq (1+\epsilon) \|Ax_{\text{true}} - b\|_2$$

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

Cost: if $F$ dense, computing $FA$
using dense matmul cost $O(k \cdot \text{min})$

$$= O(m \cdot n^2 \log n / \epsilon^2)$$
biggger than doing $A = \mathcal{O}(\sqrt{m})$

Use Cheaper $F'$:
SRTT (SRFFT or SRHT)
with dense $A$

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

$FA$ has size $kn$ so solving
smaller LS problem $\arg \min_x \|FAx-Fb\|_2$
costs $O(kn^2) = O(n^3 \log n / \epsilon^2)$

$\Rightarrow$ Total cost = $O(m \cdot n \log n + n^3 \log n / \epsilon^2)$
potentially much cheaper than QR, $O(mn^2)$ when $m \gg n$

May be ok if $\varepsilon$ not too small

(if $\varepsilon$ 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

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

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 $\arg\min_k \| (FA)x - Fb \|_2$ using dense QR would cost

$O(kn^2) = O(n^4 \cdot \log^6 (\frac{n}{\varepsilon}) / \varepsilon^2)$

much larger than $SRTT$
To trick: use randomization again to solve new LS problem.
(Use SRTT)

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

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

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

$A^{mn}$, assume target rank $k < n$.
Usually don't know $k$ accurately, so in practice pick $k + p$, $p$ extra columns in $F$, oversampling for "safety"

Given $A^{mn}$, 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) $B = QT A$ $(k+p) \times n$

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 $A = U_A \Sigma_A V_A^T$

$QQ^T A = U (1:m, 1:k+p) \Sigma (1:k+p, 1:k+p)^T$

$\cdot (V (1:n, 1:k+p))$

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

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

Our goal is just to get error proportional to $\sigma_{k+1}$
Thm: If each $F(i,j)$ is i.i.d. $N(0,1)$
then $E\left\|A-QQ^TA\right\|_2^2$
$$\leq \left(1 + \frac{4}{\sqrt{p}} \frac{\sqrt{k_p}}{\min(m,n)} \right) \sigma_{k+1}$$

$$\text{Prob} \left( \left\|A-QQ^TA\right\|_2^2 \leq \left(1 + \frac{11}{\sqrt{p}} \frac{\sqrt{k_p}}{\min(m,n)} \right) \sigma_{k+1} \right) \geq 1 - \frac{6}{p}$$

$p=6 \Rightarrow \text{prob} \approx 0.9999$

When is Randomized Low Rank Approximation cheaper than QRPCP, which costs $O(m\cdot n\cdot (k+p))^2$?

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

(2) $Y = A - F$ costs $2\text{nnz}(A) \cdot (k+p)$
(3) $Y = QR$ costs $2m(k+p)^2$
(4) $B = Q^TA$ costs $2\text{nnz}(A) \cdot (k+p)$

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

Whether cost of (3) dominates
(2) and (4) depends on how dense $A$ is: if $A$ has at least $k + p$ nonzeros 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 $F$

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

If we use SRTT for $F$

$\text{cost } (A \cdot F)$ drops to $O(m \cdot n \cdot \log m)$, much faster than QRCP

Factoring $F = QR$ 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. $Y = A \cdot F$, $m \times (k+p)$.
3. $Y = QR$.
4. Find "most linearly independent" $k+p$ rows of $Q$:
   \[ Q \equiv \begin{bmatrix} Q_1 \vDash_{k+p} \cr Q_2 \vDash_{m-(k+p)} \end{bmatrix} \]
   Write $PQ = \begin{bmatrix} Q_1 \vDash_{k+p} \cr Q_2 \vDash_{m-(k+p)} \end{bmatrix}$.

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

5. $X = PQ \cdot Q_i^T = \begin{bmatrix} Q_1 \vDash_{k+p} \cr Q_2 \vDash_{m-(k+p)} \end{bmatrix}$.

   We expect $\|X\| \approx O(1)$.

   (true if QRCP provides a strong rank revealing factorization).

6. $PA = \begin{bmatrix} A_1 \vDash_{k+p} \\ A_2 \vDash_{m-(k+p)} \end{bmatrix}$ result is
   \[ A \approx P^TX \cdot A_1 \]
   \[ \boxed{\cdot} \]
Cost one dense $A$

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

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

(4) G-EPP on $Q$ or QRCP on $Q^T :$

$$2m (k + p)^2$$

(5) $Q_2 \cdot Q_1^{-1} : O(m (k + p)^4)$

much better than previous

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

If QRCP or G-EPP works well in (5)

i.e. $\|x\| = O(1)$, then approximations nearly as good as $QQ^T A$

**Thm:** $\| A - P^T x \cdot A \|_2 \leq (1 + \|x\|_2) \| A - QQ^T A \|_2$

**proof:** assume $P = I$

$$\| A - x \cdot A \|_2 = \| A - QQ^T A + QQ^T A - x \cdot A \|_2$$

$$\leq \| A - QQ^T A \|_2 + \| QQ^T A - x \cdot A \|_2$$

$$= \| A - QQ^T A \|_2 + \| x \cdot Q \cdot Q^T A - x \cdot A \|_2$$

$$\leq \| A - QQ^T A \|_2 + \| x \|_2 \cdot \| Q \cdot Q^T A - A \|_2$$

$$\leq \| A - QQ^T A \|_2 + (\| x \|_2 \cdot \| Q \cdot Q^T A - [A_2 \cdot A_1] \|_2$$
\[ \| A - QQ^T A \|^2_2 \leq \| 1 \times \|^2_2 \| QQ^T A - A \|^2_2 \\
= (1 + \| 1 \times \|^2_2) \| (A - QQ^T A) \|^2_2 \]