Welcome back to Ma221! Lecture 22, Oct 13

Randomized Linear Algebra

LS or SVD on $A^{m \times n}$ $\quad m >> n$

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

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

Cost? $Q^T A$ costs $2mnk$ if done as standard dense matmul,
only 2x cheaper than QRCP, k steps

$\Rightarrow$ can also use cheap structured Q to make $Q^T A$ cheaper

First big speedup, beats LAPACK for LS (for some matrix sizes) Blendenpik, 2010
Avron, Maymounkov, Toledo,

Best theoretical result so far, 2012,
for LS, $O(nnz(A))$ for sparse A
Clarkson, Woodruff 2012

Examples in low dimension of why random $Q^T A$ good idea

Ex: $x \in \mathbb{R}^2$, $q \in \mathbb{R}^2$ $\quad \|q\|_2 = 1$

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

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

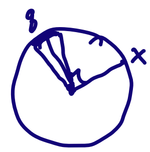
What is distribution of

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

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

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

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



$$\text{Prob}(|\cos^2 t| < e) \approx 2 e^{\frac{1}{2}} / \pi$$

when $e << 1$

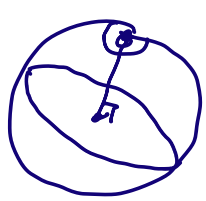
Ex: $x \in \mathbb{R}^3$, Q random plane through 0

i.e. random 3×2 orthog. matrix

$x^T Q \Rightarrow$ projection of x onto plane,

How well does $\|x^T Q\|_2^2$ approx $\|x\|_2^2$?

$\|x^T Q\|_2^2 < e \|x\|_2^2$ when x nearly parallel to the perpendicular to Q



chance that x lies in little circle proportional to e vs $\sqrt{e}$

Johnson-Lindenstrauss (JL) Lemma

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

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

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

Then with probability $\geq \frac{1}{n}$ for all $1 \leq i, j \leq n$ $i \neq j$

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

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

Proof: think of F fixed, x_i random instead

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

just need to reason about sums of squares of $N(0,1)$ random variables

(Dasgupta+Gupta, web page)

How to generate random orthog?

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

$A = QR$ Q random orthog

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

expensive: cost $O(mk^2)$ same as deterministic LS

Cheaper: represent Q as product

$Q = \prod H_i$ $H_i = (I - 2u_i u_i^T)$

pick each u_i random unit vector

cost $= O(mk)$

Some uses of F can use $N(0,1)$ entries, save QR for later in alg

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

Alternatives to random orthog:

Subsampled randomized trig transform (SRTT) trig = FFT

$F \cdot x$ will cost $O(m \cdot \log m)$ or $O(m \cdot \log k)$

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

$D = m \times m$ diagonal matrix where $D(i,i)$ uniform on unit circle

FFT = Fast Fourier Transform

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

Real Case: SRHT =

subsampled randomized Hadamard transf.

FFT replaced by H = Hadamard

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

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

Intuition for why $\|Fx\|_2 \approx \|x\|_2$

FFT, or H, "mixes" the entries of x

so sampling k of them (multiply by R)

good enough to estimate norm

When x is sparse, want it faster

Goal: $\text{cost}(Fx) = O(\text{nnz}(x))$

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

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

S is $k \times m$, each column is a random selected column of $I_{k\times k}$

$y = S \cdot Dx$:

for each nonzero x_i

pick random y_j

$y_j = y_j \pm x_i$

Called Randomized Sparse Embedding

$F = S \cdot D$ not as statistically strong as previous F's $\Rightarrow$ need larger k

---

Apply these choices of F to LS

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

$$\underset{x}{\operatorname{argmin}} \| FAx - Fb \|_2$$

"sketch and iterative"

use $F \cdot A$ to build a preconditioner (see Chap6) to iteratively solve

$$\underset{x}{\operatorname{argmin}} \| Ax - b \|_2$$

Suppose we use JL, $k = n \log n / \varepsilon^2$ rows

provides $\underset{x}{\operatorname{argmin}} \| FAx - Fb \|_2 = x_{approx}$

vs. $\underset{x}{\operatorname{argmin}} \| Ax - b \|_2 = x_{true}$

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

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

Cost: if F dense, computing $F \cdot A$ using dense matmul cost $O(m \cdot n \cdot k)$

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

worse than doing $A = QR$ $O(m \cdot n^2)$

Use cheaper F: SRTT, with dense A

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

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

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

potentially cheaper than $A = QR$ when $m >> n$ $O(mn^2)$

May be ok if ε not too small

(if ε small, need to sketch + iterate, Chap 6)

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

Clarkson + Woodroff, Meng + Mahoney

F = Randomized sparse embedding

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

Forming FA and Fb, cost $nnz(A)$ and $nnz(b)$

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

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

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

much worse than SRTT

Solution: use randomization again to solve $\arg\min_x \|(FA)x - Fb\|_2$ using SRFT

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 smallest residual, probability of success

$$= 1 - \frac{1}{3^s}$$