CountSketch Satisfies the JL Property

- (JL Property) A distribution on matrices $S \in R^{kx\ n}$ has the (ϵ, δ, ℓ)-JL moment property if for all $x \in R^n$ with $|x|_2 = 1$,

$$E_S\left||Sx|_2^2 - 1\right|^{\ell} \leq \epsilon^{\ell} \cdot \delta$$

- We show this property holds with $\ell = 2$. First, let us consider $E_S[|Sx|_2^2]$

- For CountSketch matrix S, let
  - h:[n] -> [k] be a 2-wise independent hash function
  - σ: $[n] \rightarrow \{-1,1\}$ be a 4-wise independent hash function

- Let $\delta(E) = 1$ if event E holds, and $\delta(E) = 0$ otherwise

- $E[|Sx|_2^2] = \sum_{j \in [k]} E[\left(\sum_{i \in [n]} \delta(h(i) = j)\sigma_i x_i\right)^2]$

$$= \sum_{j \in [k]} \sum_{i1,i2 \in [n]} E[\delta(h(i1) = j)\delta(h(i2) = j)\sigma_{i1}\sigma_{i2}]x_{i1}x_{i2}$$

$$= \sum_{j \in [k]} \sum_{i \in [n]} E[\delta(h(i) = j)^2]x_i^2$$

$$= \left(\frac{1}{k}\right) \sum_{j \in [k]} \sum_{i \in [n]} x_i^2 = |x|_2^2$$

CountSketch Satisfies the JL Property

- $\mathrm{E}[|Sx|_2^4] = \mathrm{E}[\sum_{j \in [k]} \sum_{j' \in [k]} \quad \left(\sum_{i \in [n]} \delta(h(i) = j)\sigma_i x_i\right)^2 \left(\sum_{i' \in [n]} \delta(h(i') = j')\sigma_{i'} x_{i'}\right)^2] =$

  $\sum_{j_1, j_2 , i_1, i_2, i_3, i_4} \mathrm{E}[\sigma_{i1}\sigma_{i2}\sigma_{i3}\sigma_{i4}\delta(h(i_1) = j_1)\delta(h(i_2) = j_1)\delta(h(i_3) = j_2)\delta\big(h(i_4 = j_2)\big)]x_{i1}\, x_{i2} x_{i3} x_{i4}$

- We must be able to partition $\{i_1, i_2, i_3, i_4\}$ into equal pairs

- Suppose $i_1 = i_2 = i_3 = i_4$. Then necessarily $j_1 = j_2$. Obtain $\sum_j \frac{1}{k} \sum_i x_i^4 = |x|_4^4$

- Suppose $i_1 = i_2$ and $i_3 = i_4$ but $i_1 \neq i_3$. Then get $\sum_{j_1, j_2, i_1, i_3} \frac{1}{k^2} x_{i_1}^2 x_{i_3}^2 = |x|_2^4 - |x|_4^4$

- Suppose $i_1 = i_3$ and $i_2 = i_4$ but $i_1 \neq i_2$. Then necessarily $j_1 = j_2$. Obtain $\sum_j \frac{1}{k^2} \sum_{i_1, i_2} x_{i_1}^2 x_{i_2}^2 \leq \frac{1}{k} |x|_2^4$. Obtain same bound if $i_1 = i_4$ and $i_2 = i_3$.

- Hence, $\mathrm{E}[|Sx|_2^4] \in [|x|_2^4, |x|_2^4(1 + \frac{2}{k})] = [1, 1 + \frac{2}{k}]$

- So, $\mathrm{E}_S\big||Sx|_2^2 - 1\big|^2 \leq \left(1 + \frac{2}{k}\right) - 2 + 1 = \frac{2}{k}$. Setting $k = \frac{2}{\epsilon^2 \delta}$ finishes the proof

Where are we?

- (JL Property) A distribution on matrices $S \in R^{kx\,n}$ has the (ϵ, δ, ℓ)-JL moment property if for all $x \in R^n$ with $|x|_2 = 1$,

$$E_S\left||Sx|_2^2 - 1\right|^{\ell} \leq \epsilon^{\ell} \cdot \delta$$

- (From vectors to matrices) For $\epsilon, \delta \in \left(0, \frac{1}{2}\right)$, let D be a distribution on matrices S with k rows and n columns that satisfies the (ϵ, δ, ℓ)-JL moment property for some $\ell \geq 2$. Then for A, B matrices with n rows,

$$\Pr_S\left[\left|A^T S^T SB\ - A^T B\right|^2{}_F \geq 3\ \epsilon^2\ |A|_F^2 |B|_F^2\right] \leq \delta$$

- We showed CountSketch has the JL property with $\ell = 2,\ \text{and}\ k = \frac{2}{\epsilon^2 \delta}$

- Matrix product result we wanted was:

  Pr[$|CS^TSD$ – $CD|_F^2$ ‰ (6/(δk)) * $|C|_F^2$ $|D|_F^2$] $\geq 1 - \delta$

- We are now done with the proof CountSketch is a subspace embedding

Course Outline

- Subspace embeddings and least squares regression
  - Gaussian matrices
  - Subsampled Randomized Hadamard Transform
  - CountSketch
- Affine embeddings
  - Application to low rank approximation
- High precision regression
- Leverage score sampling
- Distributed low rank approximation
- L1 Regression
- M-Estimator regression

Affine Embeddings

- Want to solve $\min_{\mathrm{X}} |\mathrm{AX} - \mathrm{B}|_{\mathrm{F}}^2$, A is tall and thin with d columns, but B has a large number of columns

- Can’t directly apply subspace embeddings

- Let’s try to show $|\mathrm{SAX} - \mathrm{SB}|_{\mathrm{F}} = (1 \pm \epsilon)|\mathrm{AX} - \mathrm{B}|_{\mathrm{F}}$ for all X and see what properties we need of S

- Can assume A has orthonormal columns

- Let $\mathrm{B}^* = \mathrm{AX}^* - \mathrm{B}$, where X^* is the optimum

- $|\mathrm{S}(\mathrm{AX} - \mathrm{B})|_{\mathrm{F}}^2 - |\mathrm{SB}^*|_{\mathrm{F}}^2 = |\mathrm{SA}(\mathrm{X} - \mathrm{X}^*) + \mathrm{S}(\mathrm{AX}^* - \mathrm{B})|_{\mathrm{F}}^2 - |\mathrm{SB}^*|_{\mathrm{F}}^2$
  $= |\mathrm{SA}(\mathrm{X} - \mathrm{X}^*)|_{\mathrm{F}}^2 + 2\mathrm{tr}[(\mathrm{X} - \mathrm{X}^*)^{\mathrm{T}}\mathrm{A}^{\mathrm{T}}\mathrm{S}^{\mathrm{T}}\mathrm{SB}^*]$ (use $|\mathrm{C} + \mathrm{D}|_{\mathrm{F}}^2 = |\mathrm{C}|_{\mathrm{F}}^2 + |\mathrm{D}|_{\mathrm{F}}^2 + 2\mathrm{Tr}(\mathrm{C}^{\mathrm{T}}\mathrm{D})$)
  $\in |\mathrm{SA}(\mathrm{X} - \mathrm{X}^*)|_{\mathrm{F}}^2 \pm 2|\mathrm{X} - \mathrm{X}^*|_{\mathrm{F}}\left|\mathrm{A}^{\mathrm{T}}\mathrm{S}^{\mathrm{T}}\mathrm{SB}^*\right|_{\mathrm{F}}$ (use $\mathrm{tr}(\mathrm{CD}) \le |\mathrm{C}|_{\mathrm{F}}|\mathrm{D}|_{\mathrm{F}}$)
  $\in |\mathrm{SA}(\mathrm{X} - \mathrm{X}^*)|_{\mathrm{F}}^2 \pm 2\epsilon|\mathrm{X} - \mathrm{X}^*|_{\mathrm{F}}|\mathrm{B}^*|_{\mathrm{F}}$ (if we have approx. matrix product)
  $\in |\mathrm{A}(\mathrm{X} - \mathrm{X}^*)|_{\mathrm{F}}^2 \pm \epsilon(|\mathrm{A}(\mathrm{X} - \mathrm{X}^*)|_{\mathrm{F}}^2 + 2|\mathrm{X} - \mathrm{X}^*|_{\mathrm{F}}|\mathrm{B}^*|)$ (subspace embedding for A)

Affine Embeddings

- We have

$$|S(AX - B)|_F^2 \ - |SB^*|_F^2 \in |A(X - X^*)|_F^2 \pm \epsilon(|A(X - X^*)|_F^2 + 2|X - X^*|_F|B^*|)$$

- Normal equations imply that

$$|AX - B|_F^2 = |A(X - X^*)|_F^2 + |B^*|_F^2$$

- $|S(AX - B)|_F^2 \ - |SB^*|_F^2 - \left(|AX - B|_F^2 - |B^*|_F^2\right)$

$$\begin{aligned}
&\in \ \epsilon(|A(X - X^*)|_F^2 + 2|X - X^*|_F|B^*|_F) \\
&\in \ \pm\epsilon\left(|A(X - X^*)|_F \ + |B^*|_F\right)^2 \\
&\in \ \pm 2\epsilon\left(|A(X - X^*)|_F{}^2 \quad + |B^*|_F^2 \quad \right) \\
&= \ \pm 2\epsilon|AX - B|_F^2
\end{aligned}$$

- $|SB^*|_F^2 = (1 \pm \epsilon)|B^*|_F^2$ (this holds with constant probability)

Affine Embeddings

- Know: $|S(AX - B)|_F^2 - |SB^*|_F^2 - \left(|AX - B|_F^2 - |B^*|_F^2\right) \in \pm 2\epsilon|AX - B|_F^2$
- Know: $|SB^*|_F^2 = (1 \pm \epsilon)|B^*|_F^2$

- $|S(AX - B)|_F^2 = (1 \pm 2\epsilon)|AX - B|_F^2 \pm \epsilon|B^*|_F^2$
  $= (1 \pm 3\epsilon)|AX - B|_F^2$

- Completes proof of affine embedding!

Affine Embeddings: Missing Proofs

- Claim: $|A + B|_F^2 = |A|_F^2 + |B|_F^2 + 2\mathrm{Tr}(A^T B)$

- Proof: $|A + B|_F^2 = \sum_i |A_i + B_i|_2^2$

$$= \sum_i |A_i|_2^2 + \sum_i |B_i|_2^2 + 2\langle A_i, B_i \rangle$$

$$= |A|_F^2 + |B|_F^2 + 2\mathrm{Tr}(A^T B)$$

Affine Embeddings: Missing Proofs

- Claim: $\mathrm{Tr}(\mathrm{AB}) \leq |\mathrm{A}|_{\mathrm{F}} |\mathrm{B}|_{\mathrm{F}}$

- Proof: $\mathrm{Tr}(\mathrm{AB}) = \sum_{\mathrm{i}} \langle \mathrm{A}^{\mathrm{i}}, \mathrm{B}_{\mathrm{i}} \rangle$ for rows A^{i} and columns B_{i}

$\leq \sum_{\mathrm{i}} \left|\mathrm{A}^{\mathrm{i}}\right|_2 |\mathrm{B}_{\mathrm{i}}|_2$ by Cauchy-Schwarz for each i

$\leq \left(\sum_{\mathrm{i}} \left|\mathrm{A}^{\mathrm{i}}\right|_2^2\right)^{\frac{1}{2}} \left(\sum_{\mathrm{i}} |\mathrm{B}_{\mathrm{i}}|_2^2\right)^{\frac{1}{2}}$ another Cauchy-Schwarz

$= |\mathrm{A}|_{\mathrm{F}} |\mathrm{B}|_{\mathrm{F}}$

Affine Embeddings: Homework Proof

- Claim: $|SB^*|_F^2 = (1 \pm \epsilon)|B^*|_F^2$ with constant probability if CountSketch matrix S has $k = O(\frac{1}{\epsilon^2})$ rows

- Proof is Homework Problem

- $|SB^*|_F^2 = \sum_i |SB_i^*|_2^2$

- By our analysis for CountSketch and linearity of expectation, $E\left[|SB^*|_F^2\right] = \sum_i E[|SB_i^*|_2^2] = |B^*|_F^2$

Low rank approximation

- A is an n x d matrix
  - Think of n points in R^d

- E.g., A is a customer-product matrix
  - $A_{i,j}$ = how many times customer i purchased item j

- A is typically well-approximated by low rank matrix
  - E.g., high rank because of noise

- Goal: find a low rank matrix approximating A
  - Easy to store, data more interpretable

What is a good low rank approximation?

Singular Value Decomposition (SVD)

Any matrix A = U ƒΣ ƒV

- U has orthonormal columns
- Σ is diagonal with non-increasing positive entries down the diagonal
- V has orthonormal rows

- Rank-k approximation: A_k = U_k ƒΣ_k ƒV_k
  - rows of V_k are the top k principal components

$$\begin{pmatrix} \\ \mathbf{A} \\ \\ \end{pmatrix} = \begin{pmatrix} \\ \mathbf{U}_k \\ \\ \end{pmatrix} \begin{pmatrix} \Sigma_k \end{pmatrix} \begin{pmatrix} \quad \mathbf{V}_k \quad \end{pmatrix} + \begin{pmatrix} \\ \mathbf{E} \\ \\ \end{pmatrix}$$

What is a good low rank approximation?

$A_k = \text{argmin}_{\text{rank } k \text{ matrices } B} |A\text{-}B|_F$

$(|C|_F = (\Sigma_{i,j}\ C_{i,j}{}^2)^{1/2}\ )$

Computing A_k exactly is expensive

$$\begin{pmatrix} & & \\ & \mathbf{A} & \\ & & \end{pmatrix} = \begin{pmatrix} & & \\ & \mathbf{U}_k & \\ & & \end{pmatrix} \begin{pmatrix} \Sigma_k \end{pmatrix} \begin{pmatrix} \quad \mathbf{V}_k \quad \end{pmatrix} + \begin{pmatrix} & & \\ & \mathbf{E} & \\ & & \end{pmatrix}$$

Low rank approximation

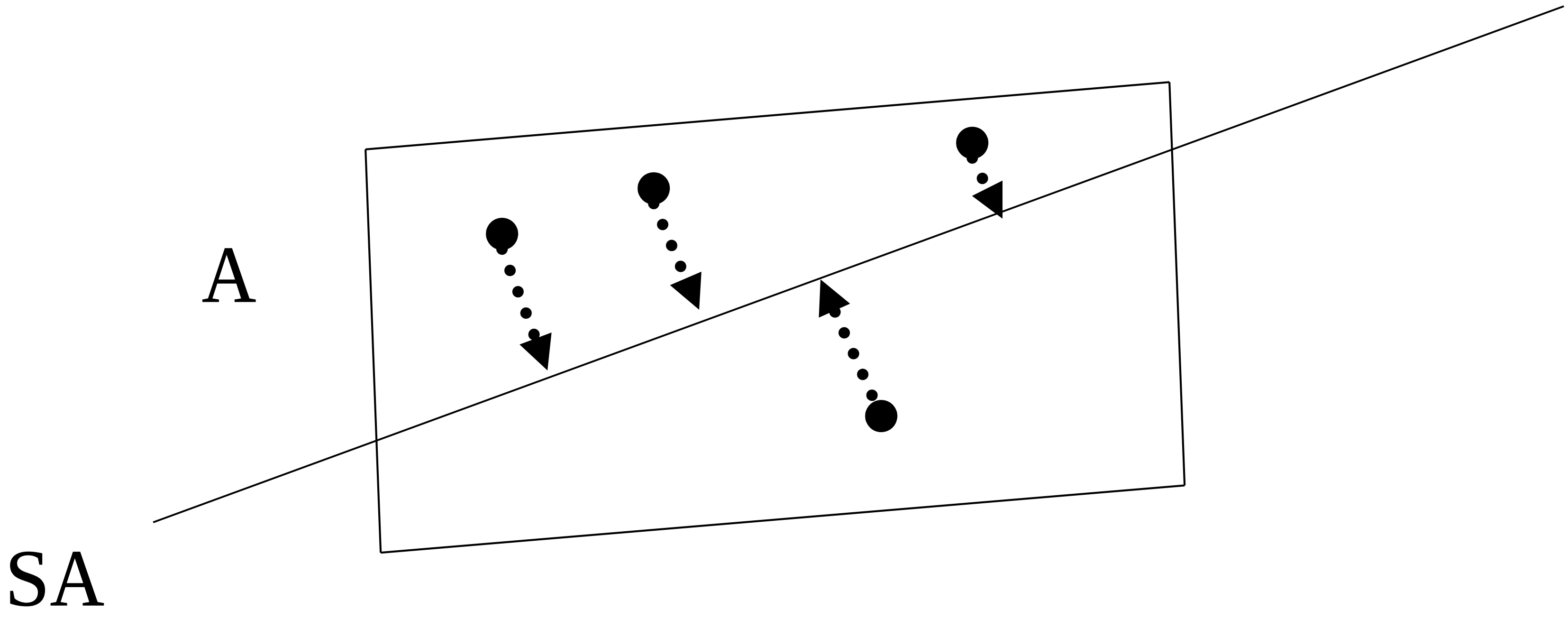
- Goal: output a rank k matrix A', so that

  $|A-A'|_F$ ‰ $(1+\varepsilon)\ |A-A_k|_F$

- Can do this in nnz(A) + (n+d)*poly(k/ε) time [S,CW]
  - nnz(A) is number of non-zero entries of A

Solution to low-rank approximation [S]

- Given n x d input matrix A
- Compute S*A using a random matrix S with k/ε << n rows. S*A takes random linear combinations of rows of A



- Project rows of A onto SA, then find best rank-k approximation to points inside of SA.

What is the matrix S?

- S can be a k/ε x n matrix of i.i.d. normal random variables
- [S] S can be a $\tilde{O}(k/\varepsilon)$ x n Fast Johnson Lindenstrauss Matrix
- [CW] S can be a poly(k/ε) x n CountSketch matrix

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

S ƒA can be computed in nnz(A) time

Why do these Matrices Work?

- Consider the regression problem $\min_X |A_k X - A|_F$
- Let S be an affine embedding
- Then $|SA_k X - SA|_F = (1 \pm \epsilon)|A_k X - A|_F$ for all X
- By normal equations, $\operatorname{argmin}_X |SA_k X - SA|_F = (SA_k)^- SA$
- So, $|A_k (SA_k)^- SA - A|_F \leq (1 + \epsilon)|A_k - A|_F$
- But $A_k (SA_k)^- SA$ is a rank-k matrix in the row span of SA!
- Let’s formalize why the algorithm works now…

Why do these Matrices Work?

- $\min_{\mathrm{rank-k}\ X} |XSA - A|_F^2 \le \left|A_k (SA_k)^- SA - A\right|_F^2 \le (1 + \epsilon)\left|A - A_k\right|_F^2$

- By the normal equations,
$$|XSA - A|_F^2 = |XSA \ - A(SA)^- SA|_F^2 + |A(SA)^- SA \ - A|_F^2$$

- Hence,
$$\min_{\mathrm{rank-k}\ X} |XSA - A|_F^2 = |A(SA)^- SA \ - A|_F^2 + \min_{\mathrm{rank-k}\ X} |XSA - A(SA)^- SA|_F^2$$

- Can write $SA = U\,\Sigma V^T$ in its SVD

- Then, $\min_{\mathrm{rank-k}\ X} |XSA - A(SA)^- SA|_F^2 = \min_{\mathrm{rank-k}\ X} |XU\Sigma - A(SA)^- U\Sigma|_F^2$
$$= \min_{\mathrm{rank-k}\ Y} |Y - A(SA)^- U\Sigma|_F^2$$

- Hence, we can just compute the SVD of $A(SA)^- U\Sigma$

- But how do we compute $A(SA)^- U\Sigma$ quickly?

Caveat: projecting the points onto SA is slow

- Current algorithm:
  1. Compute S*A
  2. Project each of the rows onto S*A
  3. Find best rank-k approximation of projected points inside of rowspace of S*A

- Bottleneck is step 2

$$\min_{\text{rank-k } X} |X(SA)R-AR|_F^2$$

Can solve with affine embeddings

- [CW] Approximate the projection
  - Fast algorithm for approximate regression

$$\min_{\text{rank-k } X} |X(SA)-A|_F^2$$

  - Want nnz(A) + (n+d)*poly(k/ε) time

Using Affine Embeddings

- We know we can just output $\arg\min_{\text{rank}-k\, X} |XSA - A|_F^2$

- Choose an affine embedding R:

$$|XSAR - AR|_F^2 = (1 \pm \epsilon)|XSA - A|_F^2 \text{ for all X}$$

- Note: we can compute AR and SAR in nnz(A) time

- Can just solve $\min_{\text{rank}-k\, X} |XSAR - AR|_F^2$

- $\min_{\text{rank}-k\, X} |XSAR - AR|_F^2 = |AR(SAR)^{-}(SAR) - AR|_F^2 + \min_{\text{rank}-k\, X} |XSAR - AR(SAR)^{-}(SAR)|_F^2$

- Compute $\min_{\text{rank}-k\, Y} |Y - AR(SAR)^{-}(SAR)|_F^2$ using SVD which is $(n+d)\text{poly}\left(\frac{k}{\epsilon}\right)$ time

- Necessarily, $Y = XSAR$ for some X. Output $Y(SAR)^{-}SA$ in factored form. We're done!

Low Rank Approximation Summary

1. Compute SA

2. Compute SAR and AR

3. Compute $\min_{\text{rank}-k\, Y} |Y - AR(SAR)^{-}(SAR)|_F^2$ using SVD

4. Output $Y(SAR)^{-}SA$ in factored form

Overall time: nnz(A) + (n+d)poly(k/ε)