CSE 392: Matrix and Tensor Algorithms for Data

Instructor: Shashanka Ubaru

University of Texas, Austin

Spring 2024

Lecture 9: Countsketch; sketch and solve

Outline

1. Countsketch
2. Sketch and solve

Types of sketching matrices

Gaussian sketching matrix:

- Performs well. Small sketch size.
- $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ requires generating $m \cdot n$ random i.i.d entries.
- Computing $\boldsymbol{SA}$ takes $O(mnd)$ time.

SHRT: Subsampled Randomized Hadamard Transform

- $\boldsymbol{S} = \boldsymbol{PHD} \in \mathbb{R}^{m \times n}$, fewer random bits.
- Faster to apply. $\boldsymbol{SA}$ in $O(mn \log(d))$ time.
- Sketch size needed is larger.
- $\boldsymbol{A}$ should be dense.
- Issues with parallel and distributed computing.

Faster Embeddings: Countsketch

- **Sparse Embeddings:** Adaptation of `CountSketch` from streaming algorithms.
- $\boldsymbol{S}$ is of the form:

$$\begin{bmatrix} 0 & -1 & 0 & 0 & \cdots & 0 \\ +1 & 0 & 0 & +1 & \cdots & 0 \\ 0 & 0 & -1 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \cdots & -1 \end{bmatrix}$$

- One random ± 1 per column.
- Row $\boldsymbol{A}_{i*}$ of $\boldsymbol{A}$ contributes $\pm \boldsymbol{A}_{i*}$ to one of the rows of $\boldsymbol{S}\boldsymbol{A}$.

Sparse Embeddings

- **Sparse sketching matrix:** For $i \in [n]$, pick uniformly and independently: $h_i \in [m]$, $s_i \in \{-1, +1\}$, and define $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ as:

$$\boldsymbol{S}_{h_i,i} \to s_i \text{ for } i \in [n],$$

  and $\boldsymbol{S}_{j,i} \to 0$ otherwise.

- $\boldsymbol{s}$ is a sign (Radamacher) vector. The vector $\boldsymbol{h}$ hashes to m “hash buckets”. That is,

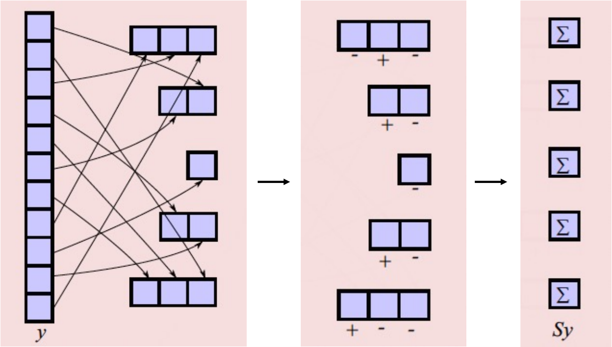
$$\boldsymbol{S}_{j*} = \sum_{i:h_i=j} s_i \boldsymbol{e}_i^\top,$$

  and so

$$[\boldsymbol{S}\boldsymbol{A}]_{j*} = \sum_{i:h_i=j} s_i \boldsymbol{e}_i^\top \boldsymbol{A} = \sum_{i:h_i=j} s_i \boldsymbol{A}_{i*}.$$

- **Fast sketching:** Can compute $\boldsymbol{S}\boldsymbol{A}$ in $O(nnz(\boldsymbol{A}))$ time.

- If $\boldsymbol{s}$ is a sign (Radamacher) vector, then $\mathbb{E}[(\boldsymbol{s}^\top \boldsymbol{y})^2] = \|\boldsymbol{y}\|_2^2$.
- For $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, each row of $\boldsymbol{S}$:
  (a) collects a subset of entries y_i's; (b) applies the signs, and (c) adds
- $\mathbb{E}[\|\boldsymbol{S}\boldsymbol{y}\|_2^2] = \|\boldsymbol{y}\|_2^2$.



Analysis of sparse embeddings

Variance of Countsketch

For $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ a sparse sketching distribution, and $\boldsymbol{y} \in \mathbb{R}^n$ a unit vector,

$$\mathrm{Var}[\|\boldsymbol{S}\boldsymbol{y}\|_2^2] \leq \frac{3}{m}.$$

Analysis of sparse embeddings

Variance of Countsketch

For $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ a sparse sketching distribution, and $\boldsymbol{y} \in \mathbb{R}^n$ a unit vector,

$$\mathrm{Var}[\|\boldsymbol{S}\boldsymbol{y}\|_2^2] \leq \frac{3}{m}.$$

Proof: Let $\boldsymbol{z} = \boldsymbol{S}\boldsymbol{y}$. We have $\mathbb{E}[\|\boldsymbol{z}\|_2^2] =$

$\mathrm{Var}[\|\boldsymbol{z}\|_2^2] =$

Analysis of sparse embeddings

Variance of Countsketch

For $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ a sparse sketching distribution, and $\boldsymbol{y} \in \mathbb{R}^n$ a unit vector,

$$\mathrm{Var}[\|\boldsymbol{S}\boldsymbol{y}\|_2^2] \leq \frac{3}{m}.$$

Proof: Let $\boldsymbol{z} = \boldsymbol{S}\boldsymbol{y}$. We have $\mathbb{E}[\|\boldsymbol{z}\|_2^2] =$

$\mathrm{Var}[\|\boldsymbol{z}\|_2^2] =$

$\mathbb{E}[\|\boldsymbol{z}\|_2^4] =$

Analysis of sparse embeddings

Variance of Countsketch

For $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ a sparse sketching distribution, and $\boldsymbol{y} \in \mathbb{R}^n$ a unit vector,

$$\mathrm{Var}[\|\boldsymbol{S}\boldsymbol{y}\|_2^2] \leq \frac{3}{m}.$$

Proof: Let $\boldsymbol{z} = \boldsymbol{S}\boldsymbol{y}$. We have $\mathbb{E}[\|\boldsymbol{z}\|_2^2] =$

$\mathrm{Var}[\|\boldsymbol{z}\|_2^2] =$

$\mathbb{E}[\|\boldsymbol{z}\|_2^4] =$

$\mathbb{E}_{s,h}[z_j^4] =$

Countsketch Embedding

Countsketch - subspace embedding

For $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ a countsketch matrix and $\boldsymbol{A} \in \mathbb{R}^{n \times d}$, if $m = O\left(\frac{d^2}{\delta \epsilon^2}\right)$, then with probability at least $1 - \delta$:

$$\|\boldsymbol{SAx}\|_2 = (1 \pm \epsilon)\|\boldsymbol{Ax}\|_2.$$

Countsketch Embedding

Countsketch - subspace embedding

For $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ a countsketch matrix and $\boldsymbol{A} \in \mathbb{R}^{n \times d}$, if $m = O\left(\frac{d^2}{\delta \epsilon^2}\right)$, then with probability at least $1 - \delta$:

$$\|\boldsymbol{SAx}\|_2 = (1 \pm \epsilon)\|\boldsymbol{Ax}\|_2.$$

We use the AMM and JL moment result.

We have $\mathrm{Var}[\|\boldsymbol{Sy}\|_2^2] \leq \frac{K}{m}$.

If $\frac{K}{m} \leq \epsilon^2 \delta$, we know $\boldsymbol{S}$ is ϵd-embedding with probability at least $1 - \delta$.

Types of sketching matrices

Sketching matrix	Sketch size m	Cost to sketch $\boldsymbol{SA}$
JL - i.i.d subGaussians	$m = O\left(\frac{d\log(1/\delta)}{\epsilon^2}\right)$	$O(mnd)$
Fast JL -SRHT	$m = O\left(\frac{d\log(d)\log(1/\delta)}{\epsilon^2}\right)$	$O(mn\log(d))$
Countsketch	$m = O\left(\frac{d^2}{\delta\epsilon^2}\right)$	$O(nnz(\boldsymbol{A}))$

We have other sparse embeddings where nnz per column is > 1, e..g, OSNAPs, sparse graphs.

Can improve $m = O\left(\frac{d\log(d)\log(1/\delta)}{\epsilon^2}\right)$ with $s = \Theta(\log(1/\delta))$ nonzero entries per column.

Further Reading

Countsketch was first introduced by:

- Clarkson, Kenneth L., and David P. Woodruff. "Low-rank approximation and regression in input sparsity time." Journal of the ACM (JACM) 63.6 (2017): 1-45

Above analysis is from:

- Nelson, Jelani, and Huy L. Nguyen. "OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings." 2013 ieee 54th annual symposium on foundations of computer science. IEEE, 2013.

Sketch and solve - least squares regression

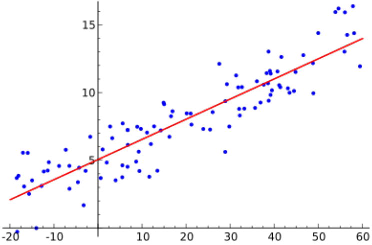
Least squares linear regression

Given a data matrix $\boldsymbol{A} \in \mathbb{R}^{n \times d}$ with n samples $\{\boldsymbol{a}_i\}_{i=1}^n \in \mathbb{R}^d$ of d-dimensional features, and a column vector $\boldsymbol{b} \in \mathbb{R}^n$ (targets):

- In the *least-squares* regression problem, assuming $d < n$, we solve:

$$\boldsymbol{x}^* = \min_{\boldsymbol{x} \in \mathbb{R}^d} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2.$$

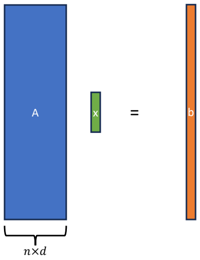
- A linear function and Euclidean- (ℓ_2) norm (squared) loss function.
- The observed targets, $b_i = \boldsymbol{a}^\top \boldsymbol{x} + \varepsilon_i$, for $i = 1, \ldots, n$ and ε_i is noise..



Overdetermined problems

$$\boldsymbol{x}^* = \min_{\boldsymbol{x}\in\mathbb{R}^d} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2.$$

- We are interested in over-constrained least-squares problems, $n \gg d$.
- Typically, there is no $\boldsymbol{x}^*$ such that $\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{b}$.
- Want to find the “best: $\boldsymbol{x}^*$ such that $\boldsymbol{A}\boldsymbol{x}^* \approx \boldsymbol{b}$.



Exact solution and ϵ-approximation

- The solution is given by the psuedo-inverse $\boldsymbol{x}^* = \boldsymbol{A}^\dagger \boldsymbol{b} = (\boldsymbol{A}^\top \boldsymbol{A})^{-1} \boldsymbol{A}^\top \boldsymbol{b}$.
- In terms of SVD, we have $\boldsymbol{A}^\dagger = \boldsymbol{V} \Sigma^{-1} \boldsymbol{U}^\top$, and
- QR factorization, we have $\boldsymbol{A}^\dagger = \boldsymbol{R}^{-1} \boldsymbol{Q}^\top$.

Complexity is $O(nd^2)$, but constant factors differ.

Exact solution and ϵ-approximation

- The solution is given by the psuedo-inverse $\boldsymbol{x}^* = \boldsymbol{A}^\dagger \boldsymbol{b} = (\boldsymbol{A}^\top \boldsymbol{A})^{-1} \boldsymbol{A}^\top \boldsymbol{b}$.
- In terms of SVD, we have $\boldsymbol{A}^\dagger = \boldsymbol{V} \Sigma^{-1} \boldsymbol{U}^\top$, and
- QR factorization, we have $\boldsymbol{A}^\dagger = \boldsymbol{R}^{-1} \boldsymbol{Q}^\top$.

Complexity is $O(nd^2)$, but constant factors differ.

ϵ-approximation

For an error parameter ϵ, compute $\tilde{\boldsymbol{x}}$ such that

$$\|\boldsymbol{A}\tilde{\boldsymbol{x}} - \boldsymbol{b}\|_2 \leq (1+\epsilon)\|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|_2$$

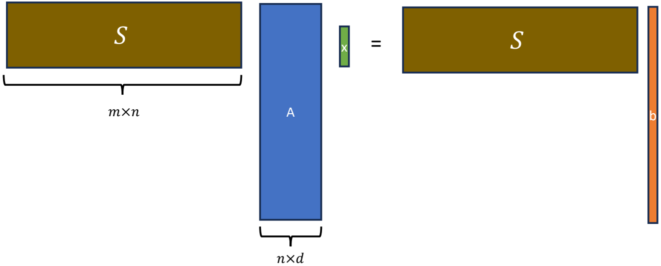
Sketch and solve

Use *Sketching*:

- Generate a sketching matrix $\boldsymbol{S} \in \mathbb{R}^{m \times n}$.
- Compute sketches $\boldsymbol{SA}$ and $\boldsymbol{Sb}$.
- Solve:

$$\tilde{\boldsymbol{x}} = \min_{\boldsymbol{x} \in \mathbb{R}^d} \|\boldsymbol{SAx} - \boldsymbol{Sb}\|_2^2.$$

- Typically, $m = \text{poly}(d/\epsilon)$.



Recall: subspace embedding

Subspace embedding

For $\boldsymbol{A} \in \mathbb{R}^{n \times d}$, a matrix $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ is a subspace ϵ-embedding for $\boldsymbol{A}$ if $\boldsymbol{S}$ is an ϵ-embedding for $span(\boldsymbol{A}) = \{\boldsymbol{A}\boldsymbol{x} \mid \boldsymbol{x} \in \mathbb{R}^d\}$. I.e., for all $\boldsymbol{x} \in \mathbb{R}^d$,

$$\|\boldsymbol{S}\boldsymbol{A}\boldsymbol{x}\|_2 = (1 \pm \epsilon)\|\boldsymbol{A}\boldsymbol{x}\|_2.$$

Sketching matrix	Sketch size m	Cost to sketch $\boldsymbol{S}\boldsymbol{A}$
JL - i.i.d subGaussians	$m = O\left(\frac{d \log(1/\delta)}{\epsilon^2}\right)$	$O(mnd)$
Fast JL -SRHT	$m = O\left(\frac{d \log(d) \log(1/\delta)}{\epsilon^2}\right)$	$O(mn \log(d))$
Countsketch	$m = O\left(\frac{d^2}{\delta \epsilon^2}\right)$	$O(nnz(\boldsymbol{A}))$

Subspace embedding for sketch and solve

Sketch and solve

Suppose $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ is a subspace ϵ-embedding for $span([\boldsymbol{A}\ b])$.
Let,

$$\boldsymbol{x}^* = \min_{\boldsymbol{x} \in \mathbb{R}^d} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$$
$$\tilde{\boldsymbol{x}} = \min_{\boldsymbol{x} \in \mathbb{R}^d} \|\boldsymbol{S}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})\|_2,$$

for $\epsilon \leq 1/3$, we have

$$\|\boldsymbol{A}\tilde{\boldsymbol{x}} - \boldsymbol{b}\|_2 \leq (1 + 3\epsilon)\|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|_2$$

Proof:

For $\boldsymbol{y} = \begin{bmatrix} \boldsymbol{x} \\ -1 \end{bmatrix}, \boldsymbol{x} \in \mathbb{R}^d$,

$\|\boldsymbol{S}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})\|_2 =$

Proof:

For $\boldsymbol{y} = \begin{bmatrix} \boldsymbol{x} \\ -1 \end{bmatrix}, \boldsymbol{x} \in \mathbb{R}^d$,

$\|\boldsymbol{S}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})\|_2 =$

$\|\boldsymbol{A}\tilde{\boldsymbol{x}} - \boldsymbol{b}\|_2 \leq$

Proof:

For $\boldsymbol{y} = \begin{bmatrix} \boldsymbol{x} \\ -1 \end{bmatrix}, \boldsymbol{x} \in \mathbb{R}^d$,

$\|\boldsymbol{S}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})\|_2 =$

$\|\boldsymbol{A}\tilde{\boldsymbol{x}} - \boldsymbol{b}\|_2 \leq$

and so for $\epsilon \leq 1/3$, $\|\boldsymbol{A}\tilde{\boldsymbol{x}} - \boldsymbol{b}\|_2 \leq (1 + 3\epsilon)\|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|_2$.

Computational cost:

Matlab demo