Exponentially Improved Dimension Reduction in ℓ_1:

Subspace Embeddings and Independence Testing

Taisuke Yasuda (CMU)

with

Yi Li (NTU)

David Woodruff (CMU)

Dimension Reduction

- *Dimension Reduction*: techniques which reduce the dimensionality of datasets, while (approximately) preserving properties of interest
  - Input: data in n-dimensions, where n is very large
  - Want a map $f: \mathbb{R}^n \to \mathbb{R}^r$ for $r \ll n$, such that $f(x)$ approximates x
  - Goal: minimize r

Linear Sketching

- *Linear Sketching*: dimension reduction map f is a linear map
  - $f(x) = Sx$, where S is an $r \times n$ matrix
  - Sx is known as the “sketch” of x

ℓ_1 Subspace Embeddings

Independence Testing

- Useful for:
  - Norm estimation
    - Given Sx, estimate $|x| \approx |Sx|$
  - Distance estimation
    - Given Sx and Sy, estimate $|x - y| \approx |S(x - y)| = |Sx - Sy|$
  - Streaming/dynamic environments
    - Sketch is very easy to update: $S(x + \Delta) = Sx + S\Delta$
  - Distributed environments
    - Sketch is very easy to aggregate: $S(x + y) = Sx + Sy$

Part (1): Subspace Embeddings

Norm Estimation in ℓ_2

- Johnson-Lindenstrauss (1984)
  - Let S to be an $r \times n$ matrix of i.i.d. Gaussians

  - Let x be an n-dimensional vector and $\varepsilon > 0$
  - If $r = \Theta(\varepsilon^{-2})$, then $|Sx|_2 = (1 \pm \varepsilon)|x|_2$

  - Let X be a set of m vectors
  - If $r = \Theta(\varepsilon^{-2} \log m)$, then $|Sx|_2 = (1 \pm \varepsilon)|x|_2$ for all $x \in X$

  - Let A be an $n \times d$ matrix
  - If $r = \Theta(\varepsilon^{-2}\, d)$, then $|Sx|_2 = (1 \pm \varepsilon)|x|_2$ for all $x \in \mathrm{span}(A)$

ℓ_2 Subspace Embedding

Norm Estimation in ℓ_1

	ℓ_2 Johnson-Lindenstrauss (1984)	ℓ_1 Upper Bound Wang-Woodruff (2019)	ℓ_1 Lower Bound Wang-Woodruff (2019)
1 vector	ε^{-2}	$\exp(\exp(\varepsilon^{-2}))$	
m vectors	$\varepsilon^{-2}\log m$	$\exp(\exp(\varepsilon^{-2}\log m))$	$\exp(\sqrt{m})$
d-dimensional subspace	$\varepsilon^{-2}d$	$\exp(\exp(\varepsilon^{-2}d))$	$\exp(\sqrt{d})$

*Suppressing big Oh, big Omega, and log factors

Our Results

	ℓ_2 Johnson-Lindenstrauss (1984)	ℓ_1 Upper Bound Li-Woodruff-Y (2021)	ℓ_1 Lower Bound Wang-Woodruff (2019)
1 vector	ε^{-2}	$\cancel{\exp(\exp(\varepsilon^{-2}))}$ $\exp(\varepsilon^{-1})$	
m vectors	$\varepsilon^{-2}\log m$	$\cancel{\exp(\exp(\varepsilon^{-2}\log m))}$ $\exp(\varepsilon^{-1}m)$	$\exp(\sqrt{m})$
d-dimensional subspace	$\varepsilon^{-2}d$	$\cancel{\exp(\exp(\varepsilon^{-2}d))}$ $\exp(\varepsilon^{-1}d)$	$\exp(\sqrt{d})$

*Suppressing big Oh, big Omega, and log factors

Our Results

- Improved dependence on ε, d from doubly exponential to singly exponential
- Singly exponential dependence on d is tight
- ℓ_1 is very different from ℓ_2
  - ℓ_1 doesn't care whether we embed d vectors or their span
  - In ℓ_2, there is an exponential difference

Li-Woodruff-Y (2021)

	ℓ_2	ℓ_1 UB	ℓ_1 LB
1 vector	ε^{-2}	$\exp(\varepsilon^{-1})$	
m vectors	$\varepsilon^{-2}\log m$	$\exp(\varepsilon^{-1}m)$	$\exp(\sqrt{m})$
d-dim subspace	$\varepsilon^{-2}d$	$\exp(\varepsilon^{-1}d)$	$\exp(\sqrt{d})$

*Suppressing big Oh, big Omega, and log factors

Our Techniques

- Improving the ε dependence:
  - Classical technique of sampling and hashing → $O(1)$ distortion
  - Randomizing sampling rates themselves → $(1+\varepsilon)$ distortion
- Improving the d dependence:
  - Net argument → doubly exponential bound
  - Applying 1 vector result to the ℓ_1 leverage score vector → singly exponential bound

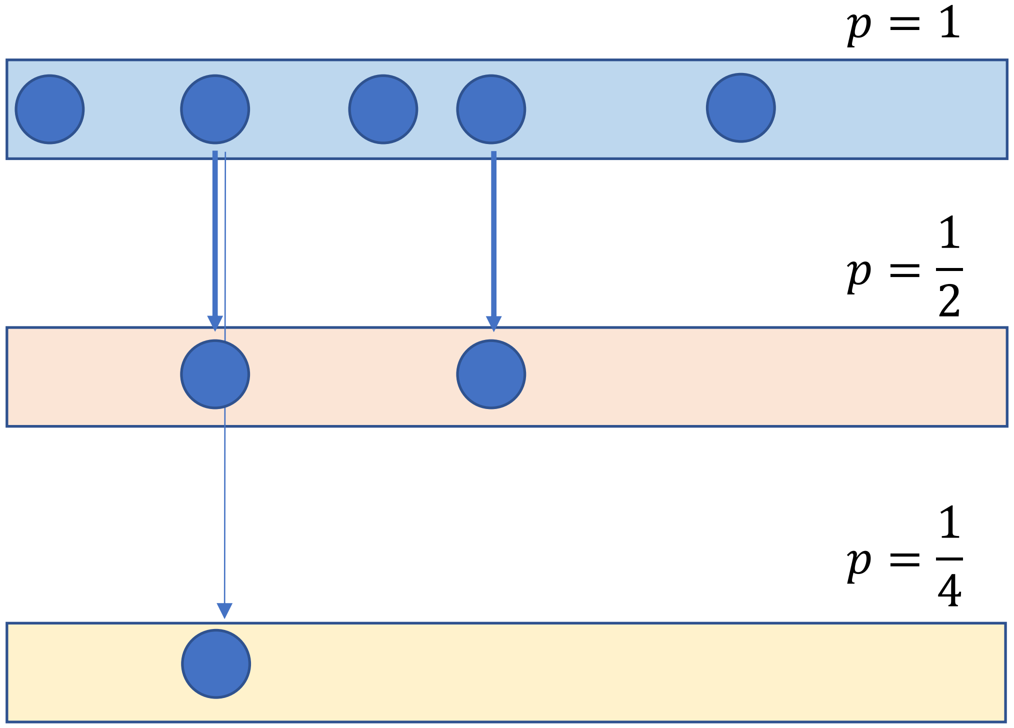
Li-Woodruff-Y (2021)

	ℓ_2	ℓ_1 **UB**	ℓ_1 **LB**
1 vector	ε^{-2}	$\exp(\varepsilon^{-1})$	
m vectors	$\varepsilon^{-2}\log m$	$\exp(\varepsilon^{-1}m)$	$\exp(\sqrt{m})$
d-dim subspace	$\varepsilon^{-2}d$	$\exp(\varepsilon^{-1}d)$	$\exp(\sqrt{d})$

*Suppressing big Oh, big Omega, and log factors

Improving the ε dependence

- Sample entries with probability p for $\log n$ levels $p = 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots, \frac{1}{n}$
- Estimate = $\frac{1}{p}$(sum of heavy survivors) at each level
- This is good in expectation, but not with high probability!
  - We can't take medians in this model
- Our idea: randomize sampling rates p to get high probability bounds



Part (2): Independence Testing

Independence Testing

- q-dimensional distribution given by a data stream of q-tuples:
  - Each stream element is $(i_1, \dots, i_q)$, where $i_j \in \{1, \dots, d\}$
  - Empirical joint distribution P:

$$p(i_1, \dots, i_q) = \frac{\text{number of occurrences of } (i_1, \dots, i_q)}{\text{length of stream}}$$

  - Empirical product distribution $Q = Q_1 \times Q_2 \times \cdots \times Q_q$

$$q_j(i) = \frac{\text{number of occurrences of } (*, \dots *, i, *, \dots, *)}{\text{length of stream}}$$

i appears at the j-th position

- Question: Estimate $\|P - Q\|_1$

Independence Testing

Braverman-Ostrovsky (2010)

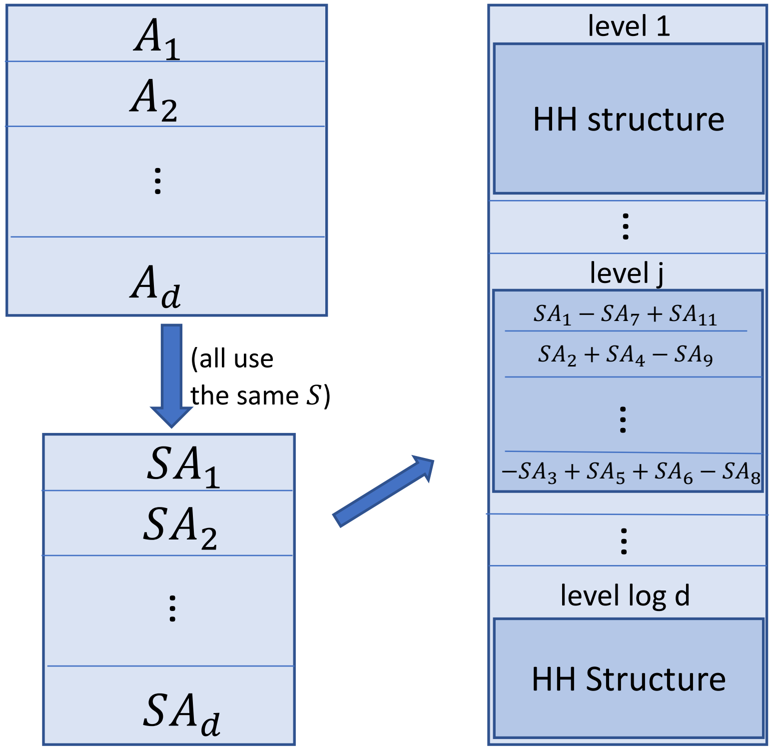
$\|P - Q\|_1$ can be estimated in $(\frac{1}{\epsilon}\log d)^{q^{O(q)}}$ space.

Li-Woodruff-Y (2021)

$\|P - Q\|_1$ can be estimated in $2^{O(q^2)}(\frac{q}{\epsilon}\log d)^{O(q)}$ space.

Estimate ℓ_1-Norm of ~~Tensor~~ Matrix

Earlier: sketch S and decoding algorithm $\mathcal{D}$ s.t. $\mathcal{D}(Sx) = (1 \pm \epsilon)\|x\|_1$



- Idea: to get the ℓ_1 norm of the matrix, we want to apply this to the vector of ℓ_1 norms of the rows
- Instead, we apply the sketch S to every row
- Whenever we want the ℓ_1 norm of a row, we estimate using the decoding algorithm $\mathcal{D}$
- Works because of the form of the sketch S

Estimate ℓ_1-Norm of Tensor

- Nested sketch
  - Bucket at mode j contains the sketch for tensor of mode $j-1$
  - Run the decoding algorithm to recover the ℓ_1 norm inside each bucket

$$\overset{\text{mode } q}{(\epsilon_q = \epsilon, \delta_q = \Theta(1))} \longleftarrow \cdots \longleftarrow \overset{\text{mode } j}{(\epsilon_j, \delta_j)} \longleftarrow \cdots \longleftarrow \overset{\text{mode } 2}{(\epsilon_2, \delta_2)} \longleftarrow \overset{\text{mode } 1}{(\epsilon_1, \delta_1)}$$

- Overall sketch length = $2^{O(q^2)}(\frac{q}{\epsilon}\log d)^{O(q)}$

Estimate ℓ_1-Norm of Tensor

- For the product distribution, unclear how to directly maintain SQ
- However, our sketch is a tensor product $S = S_1 \otimes S_2 \otimes \cdots \otimes S_q$!
- We can compute $S_1 Q_1, S_2 Q_2, \ldots, S_q Q_q$ and assemble them with a tensor product:

$$SQ = (S_1 Q_1) \otimes (S_2 Q_2) \otimes \cdots \otimes (S_q Q_q)$$

- Compute $\|P - Q\|_1$ by linearity of the sketches

Conclusion

- We gave two exponentially improved bounds for dimension reduction in ℓ_1
- For subspace embeddings in ℓ_1:
  - Previous bounds [WW19] required $\exp\big(\exp(\varepsilon^{-2}d)\big)$ dimensions
  - We show $\exp(\varepsilon^{-1}d)$ is possible
  - Our new techniques include sampling with random sampling rates and avoiding net arguments by using the ℓ_1 leverage score vector
- For independence testing:
  - Previous bounds [BO10] use $(\frac{1}{\epsilon}\log d)^{q^{O(q)}}$ space
  - We show $2^{O(q^2)}(\frac{q}{\epsilon}\log d)^{O(q)}$ space is possible
  - We recursively apply sampling and heavy hitter sketches over the modes of the tensor