Dimensionality Reduction

for Sum-of-Distances Metric

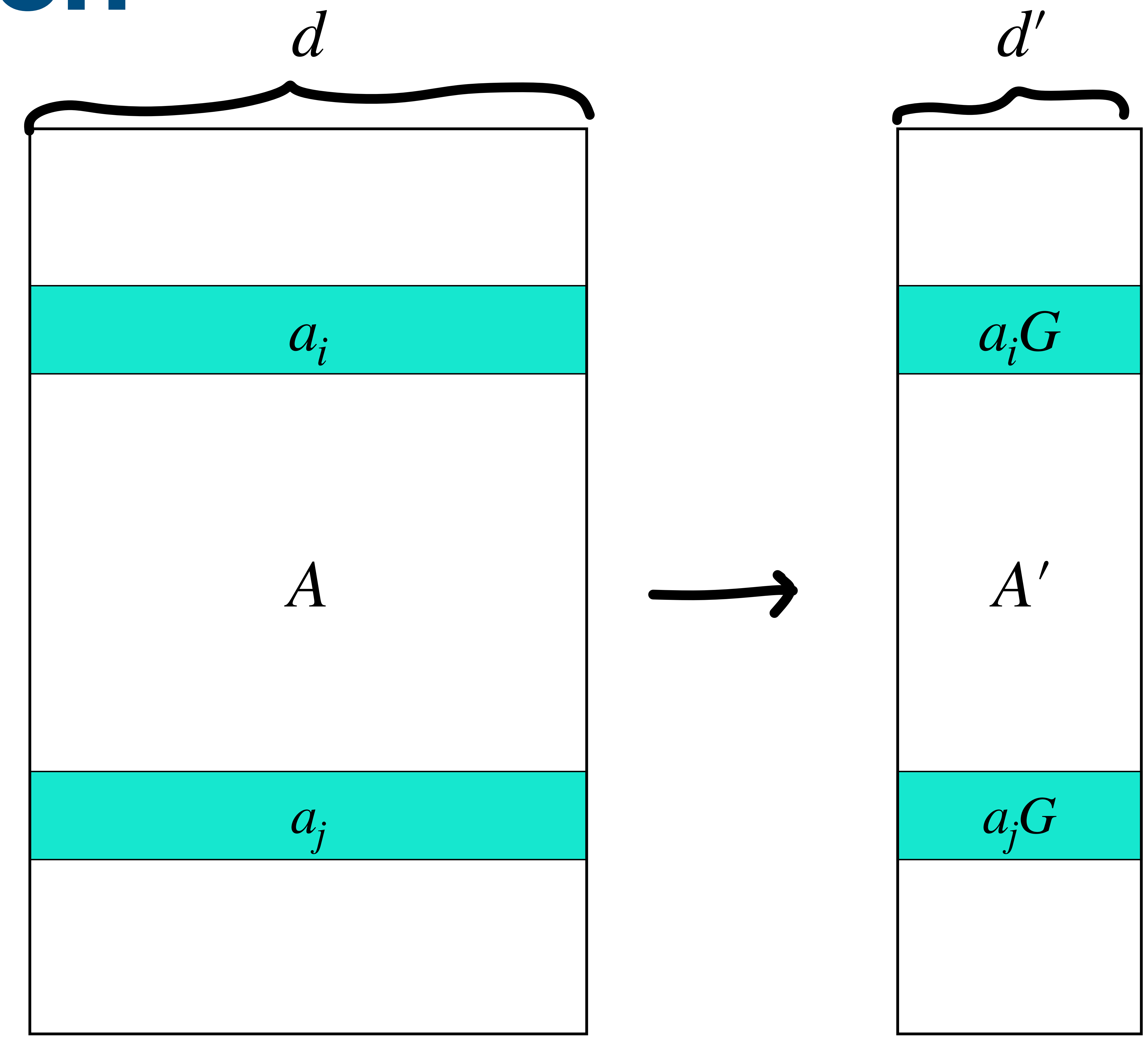
Zhili Feng, Praneeth Kacham* and David Woodruff (CMU)

Introduction

- Datasets these days are huge and high-dimensional
- Crucial to decrease size of the data to save on storage and computation
- Two ways to achieve dataset reduction:
  - Dimensionality reduction - decreasing d
  - Coresets - decreasing n (typically a weighted subset of the dataset)

Dimensionality Reduction

- If $d' \ll d$, can attain significant size reduction
- A' depends on the task we want to perform on A
- Example: If all we need is $\|a_i - a_j\|_2$, we can have $A' = AG$, where G is a Gaussian matrix.
- JL Lemma ⇒ $d' = O(\log(n)/\epsilon^2)$
- Queries can be answered in $O(d')$

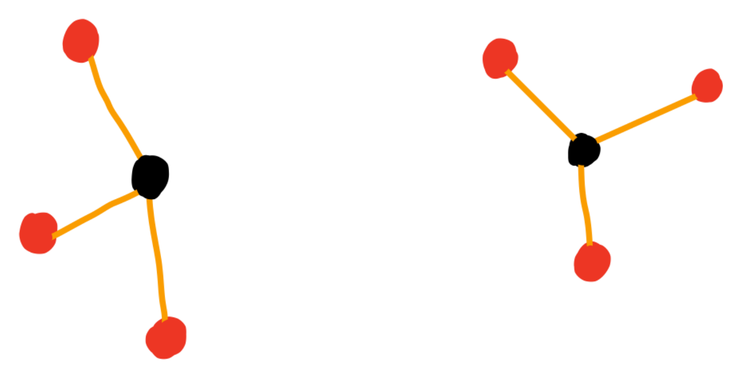


Shape Fitting

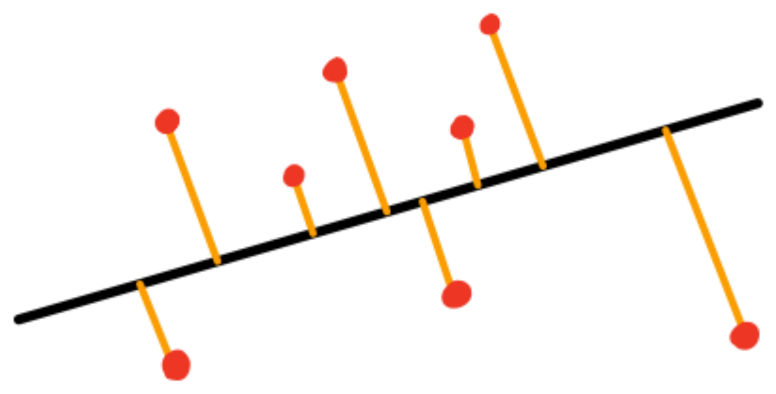
- Given A and a set of "shapes" $\mathcal{S}$, we want to find a $S \in \mathcal{S}$ that minimizes

$$d(A, S) = \sum_i d(a_i, S) = \sum_i \min_{s \in S} d(a_i, s)$$

- Captures:
  - k-median



  - Subspace Approximation



- More robust to outliers than sum of squared distances

Our Results

- We give dimensionality reduction to approximate upto a $1 \pm \epsilon$ factor, the distance to any "shape" S that lies in a k dimensional space

- We project A onto a poly(k/ϵ) dimensional subspace P

- proj(a_i, P) and dist(a_i, P) are all we need to approximate $d(A, S)$ upto an ϵ factor

- P can be stored using $d \cdot$ poly(k/ϵ) parameters and all the projections can be stored using $n \cdot$ poly(k/ϵ) parameters

Our Results

Theorem : The subspace P of poly(k/ϵ) dimensions can be computed in time nnz$(A)/\epsilon^2 + (n + d) \cdot$ poly(k/ϵ)

- For constant ϵ, the algorithm runs in input-sparsity time which can be much smaller than $n \cdot d$
- Can compute approximate projections and approximate distances to the subspace P in time nnz$(A) + (n + d) \cdot$ poly(k/ϵ)

Previous Work

- Sohler and Woodruff show that a subspace satisfying the following condition is sufficient:

$$\text{for all } W, \quad d(A, P) - d(A, P + W) \leq \epsilon^2 \text{OPT}(A)$$

- Here W is any k dimensional subspace and OPT(A) is the optimal k -Subspace approximation cost

- Existence of a k/ϵ^2 dimensional subspace P is easy

Previous Work

- Sohler and Woodruff give an algorithm to find such a subspace P
- But it runs in time $\text{nnz}(A) + (n+d) \cdot \text{poly}(k/\epsilon) + \exp(\text{poly}(k/\epsilon))$
- The $\exp(\text{poly}(k/\epsilon))$ makes it infeasible to run their algorithm in practice even for small values of k and $1/\epsilon$
- Obtaining such a subspace P in polynomial time is our major technical contribution

Obtaining such a subspace

- Suppose we have an algorithm given arbitrary A and P that can find a subspace Q of r dimensions such that

$$d(A(I-P), Q) \leq (1+\epsilon) \cdot \mathrm{OPT}(A(I-P))$$

- Run algorithm with $P_0 = \{0\}$ to get Q_1 such that

$$d(A, Q_1) \leq (1+\epsilon) \cdot \mathrm{OPT}(A)$$

- Let $P_i = Q_1 + \cdots + Q_i$ and run the algorithm with subspace P_i to get Q_{i+1} that satisfies

$$d(A, P_{i+1}) = d(A(I-P_i), Q_{i+1}) \leq (1+\epsilon) \cdot \mathrm{OPT}(A(I-P_i))$$

- This implies that, for all k-dim W,

$$d(A, P_{i+1}) \leq (1+\epsilon) \cdot d(A, P_i + W)$$

- Repeat the process for $T = 10/\epsilon$ iterations

Obtaining such a subspace

- As $d(A, P_1) \leq (1+\epsilon) \cdot \mathsf{OPT}(A)$, we have that

$$\sum_{i=1}^{T-1} d(A, P_i) - d(A, P_{i+1}) \leq (1+\epsilon) \cdot \mathsf{OPT}(A)$$

- At least $8/\epsilon$ summands above are $\leq \epsilon \cdot \mathsf{OPT}(A)$

- By definition of P_{i+1}, we also have that for all k-dim subspaces W,

$$d(A, P_{i+1}) \leq (1+\epsilon) \cdot d(A, P_i + W)$$

- Therefore for many values of i, and all k-dim subspaces W,

$$\begin{aligned} d(A, P_i) - d(A, P_i + W) &\leq \epsilon \cdot \mathsf{OPT}(A) + \epsilon \cdot d(A, P_i + W) \\ &\leq O(\epsilon) \cdot \mathsf{OPT}(A) \end{aligned}$$

Obtaining such a subspace

- Running for $10/\epsilon^2$ iterations with parameter ϵ^2 and picking subspace after a random iteration gives the desired subspace of dimension $O(r/\epsilon^2)$
- We use the framework of Clarkson and Woodruff to obtain $1+\epsilon$ approximate solutions with $r = \text{poly}(k/\epsilon)$
- Our algorithm has two stages:
  - Find an $O(1)$-approximate solution
  - Perform "residual sampling" using the $O(1)$ solution to get $1+\epsilon$ approximate solution

Finding $O(1)$ approximation

- We show using "lopsided embeddings" that if S is a Gaussian matrix with $O(k)$ columns, then

$$\min_{rank-k\ X} \|ASX - A\|_{1,2} \leq (3/2) \cdot \text{OPT}$$

- Essentially shows that column span of AS contains a good solution

- We then argue that if L is an ℓ_1 subspace embedding for the column space of AS and satisfies $E_L[\|LM\|_{1,2}] = \|M\|_{1,2}$ for any matrix M, then

$$\|A(I - (LA)^{+}LA)\|_{1,2} \leq O(1) \cdot \text{OPT}$$

- Such a matrix L with $\tilde{O}(k)$ rows can be found using Lewis Weight Sampling algorithm of Cohen and Peng.

Finding $1 + \epsilon$ approximation

- Let P be an arbitrary subspace such that:

$$\|A(I-P)\|_{1,2} \leq O(1) \cdot \mathrm{OPT}(A)$$

- Define $r_i = \|A_{i*}(I-P)\|_2$ to be the residual of i-th row

- A result of Clarkson and Woodruff shows that if A_S is obtained by sampling $\tilde{O}(k^3/\epsilon^2)$ rows of matrix A independently with probabilities proportional to r_i, then $P' = \text{rowspace}(A_S) + P$ satisfies, with constant probability,

$$\|A(I-P')\|_{1,2} \leq (1+\epsilon) \cdot \mathrm{OPT}(A)$$

Wrap up

- Rest of the analysis involves showing that the previous algorithm can be adaptively implemented with desired time complexity
- For the dense case, when $\text{nnz}(A) \approx n \cdot d$, we give an algorithm that runs in time $n \cdot d + (n + d) \cdot \text{poly}(k/\epsilon)$
- See our paper for more details
- Thank you!