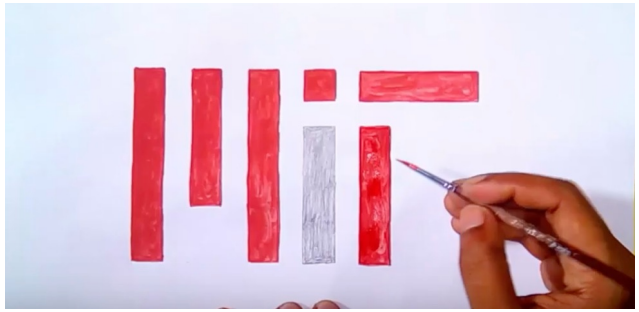
Learning-Based Low-Rank Approximations

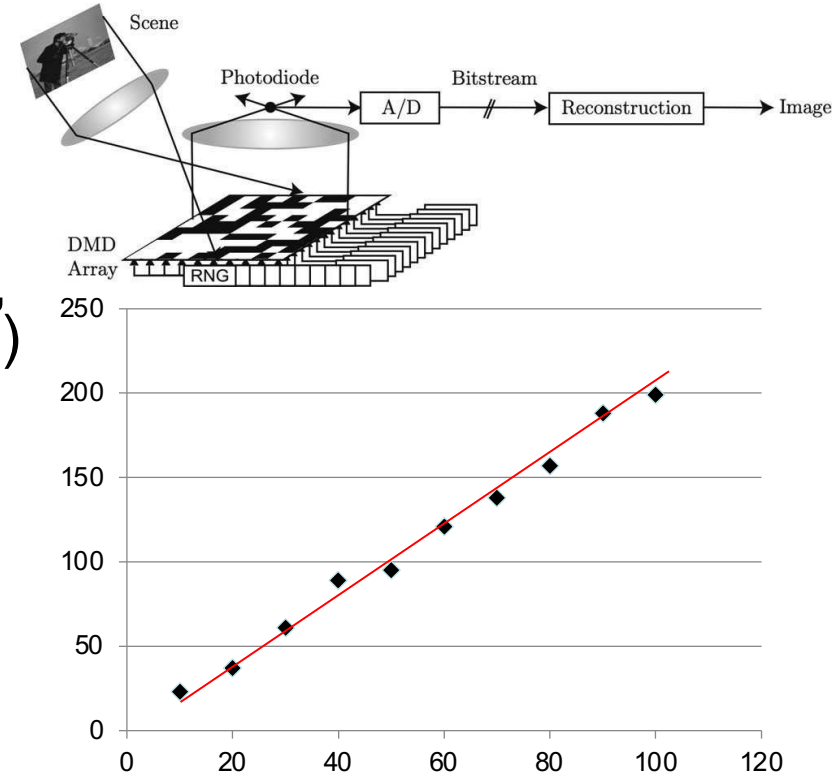
Piotr Indyk



Joint work with Peter Bartlett, Tal Wagner, David Woodruff, Ali Vakilian, Yang Yuan

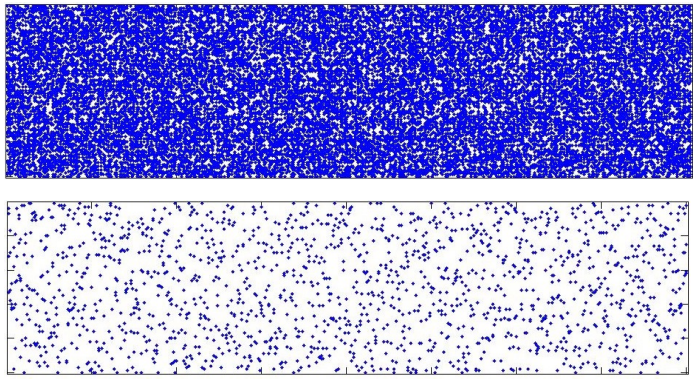
Linear Sketches

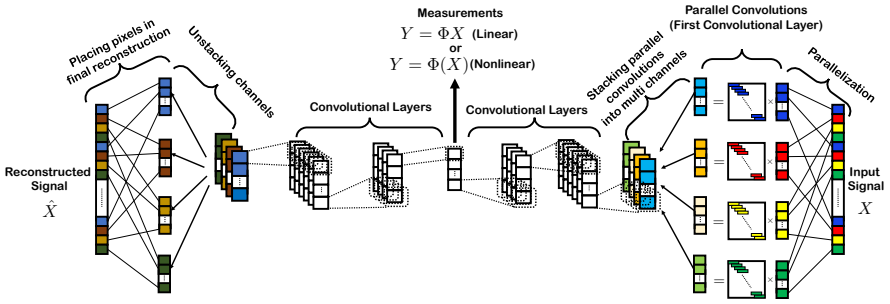
- Many algorithms are obtained using **linear sketches**:
  - Input: represented by x
  - Sketching: compress x into Sx
    - S=sketch matrix
  - Computation: on Sx
- Examples:
  - Compressed sensing
  - Dimensionality reduction (e.g., Johnson-Lindenstrauss lemma)
  - Streaming algorithms
  - ….
  - Linear algebra (regression, low-rank approximation,..)



Linear Sketches, ctd.

- S is almost always a random matrix
  - Independent Gaussians, random partial Fourier, random sparse,...
  - Pros: simple, efficient, worst-case guarantees
  - Cons: does not adapt to data
- Why not **learn** S from examples ?
  - Dimensionality reduction: e.g., PCA
  - Compressed sensing
    - Autoencoders: x → Sx →x'
  - Linear algebra ? This talk





Mousavi-Patel-Baraniuk, 2015

….

Low Rank Approximation

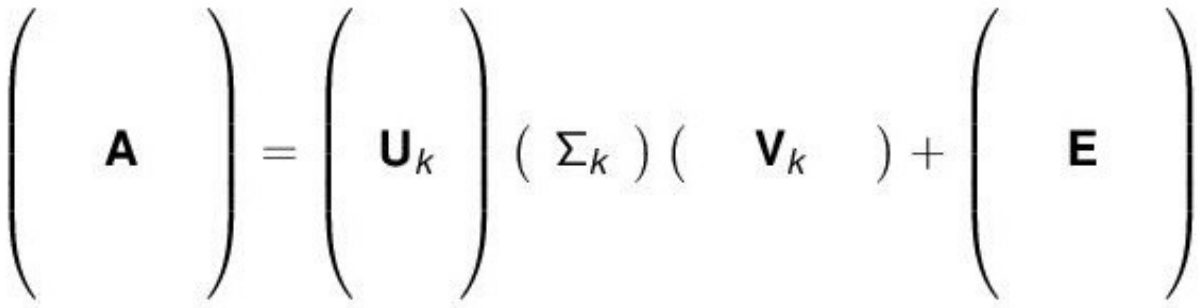
Singular Value Decomposition (SVD)

Any matrix $A = U \Sigma V$, where:

- U has orthonormal columns
- Σ is diagonal
- V has orthonormal rows

Rank-k approximation: $A_k = U_k \Sigma_k V_k$

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



Approximate Low Rank Approximation

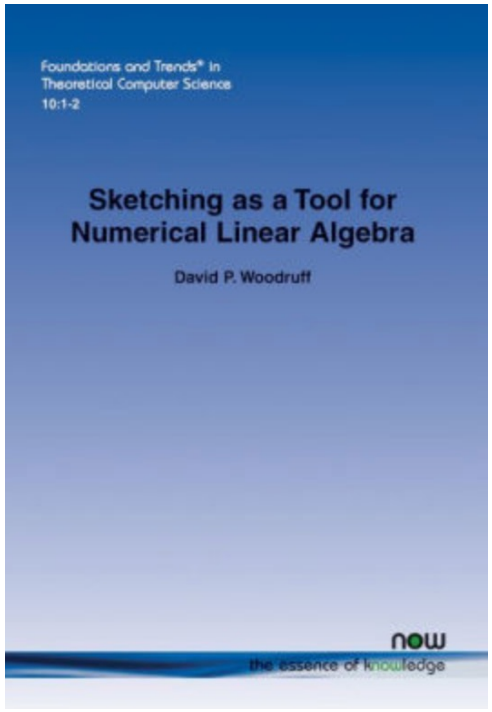
- Instead of

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

output a rank-k matrix A', so that

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

- More efficient than computing exact A_k
  - Sarlos'06, Liberty et al '07, Clarkson-Woodruff'09,13; Halko et al'11,…
  - See Woodruff'14 for a survey
- Most of these algorithms use linear sketches SA
  - S can be dense (partial Fourier) or sparse (0/+1/-1)
  - We focus on sparse S



Sarlos-ClarksonWoodruff

- **Streaming** algorithm (two passes over A):
  - Compute SA (first pass)
  - Compute orthonormal V that spans rowspace of SA
  - Compute AV^T (second pass)
  - Return $SCW(S,A) := [AV^T]_k V$
- Space:
  - Suppose that A is $n \times d$, S is $m \times n$
  - Then SA is $m \times d$, AV^T is $n \times m$
  - Space proportional to m
  - Theory: $m = O(k/\varepsilon)$
- Can we achieve lower m with learned matrix S ?

Learning-Based Low-Rank Approximation

[Indyk-Vakilian-Yuan, NeurIPS'19]

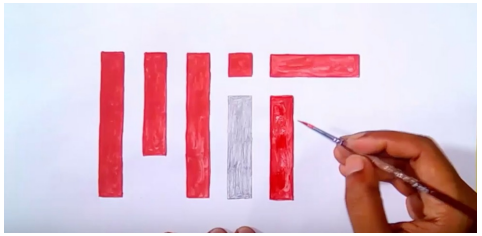
- Sample matrices $A_1 \ldots A_N$
- Find S that minimizes the following loss function

$$\sum_i ||A_i - SCW(S, A_i)||_F$$

- Use S in SCW in future computation
- "Details":
  - Use sparse matrices S
    - Random support, optimize values
  - Minimize loss using SGD in Pytorch
    - Need to differentiate loss w.r.t. S
    - Represent SVD as a sequence of power-method applications (each is differentiable)
    - Use random matrix as a starting point

$$\boldsymbol{S} = \begin{bmatrix} \rho_1 & \rho_2 & 0 & 0 & 0 & \rho_6 & 0 \\ 0 & 0 & 0 & \rho_4 & 0 & 0 & \rho_7 \\ 0 & 0 & \rho_3 & 0 & \rho_5 & 0 & 0 \end{bmatrix}$$

Evaluation



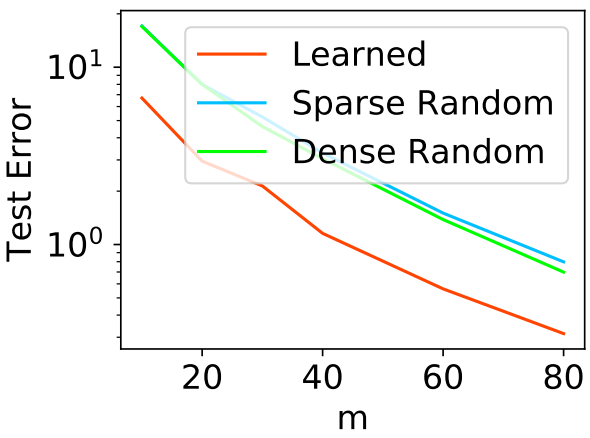
- Datasets (collections of matrices)
  - Videos: MIT Logo, Friends, Eagle
  - Hyperspectral images (HS-SOD)
  - TechTC-300
- 200/400 training, 100 testing
- Optimize the matrix S using testing matrices
- Compute the recovery error over test matrices

$$\sum_i ||A_i - SCW(S, A_i)||_F \text{ - } ||A_i - [A_i]_k||_F$$

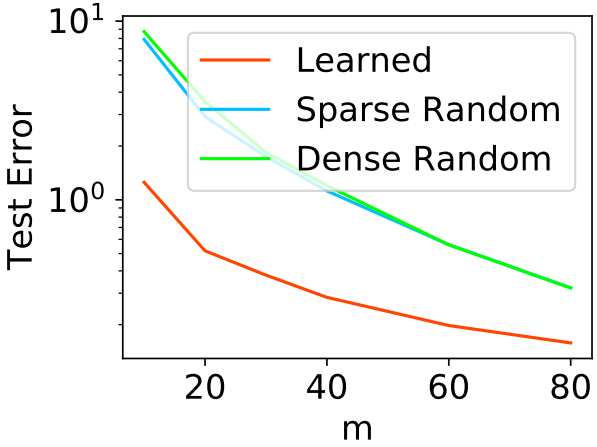
- Compare to random matrices S

Results

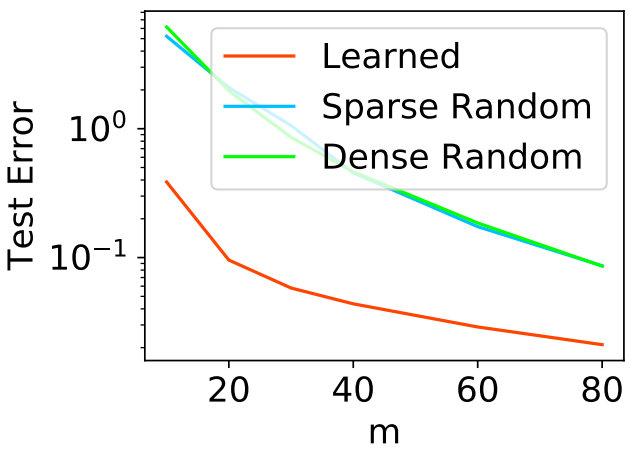
k=10



Tech



Hyper



MIT Logo

Fallback option

- Learned matrices work (much) better, but no guarantees per matrix
- Solution: combine **learned** S with **random** rows R
- Fact: augmenting R with additional (learned) matrix S cannot increase the error of SCW
- The algorithm inherits worst-case guarantees from R

Mixed matrices - results

k	m	Sketch	Logo	Hyper	Tech
10	20	Learned	0.1	0.52	2.95
10	20	Mixed	0.2	0.78	3.73
10	20	Random	2.09	2.92	7.99
10	40	Learned	0.04	0.28	1.16
10	40	Mixed	0.05	0.34	1.31
10	40	Random	0.45	1.12	3.28

Questions

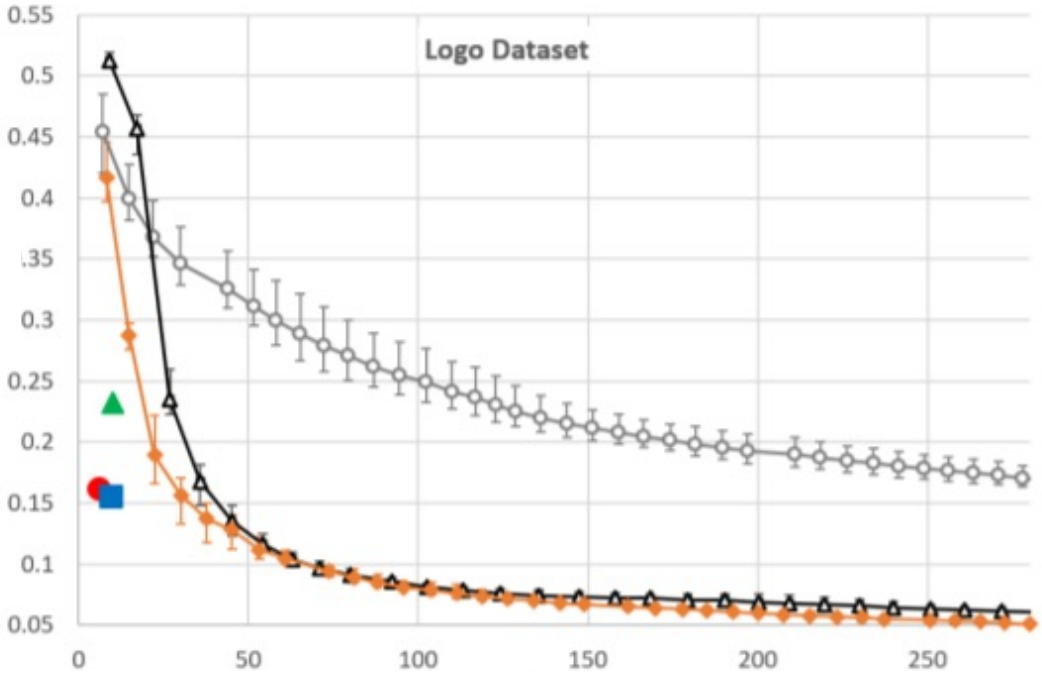
- We showed learned sketches can improve the accuracy/measurement tradeoff for low-rank approximation
- Can retain some guarantees by mixing random and learned rows
- Issues:
  - Training time is long
  - Other guarantees:
    - Sampling complexity: how many matrices are needed to learn the sketch matrix S ?
    - Provably minimize the loss function ?

Followups

Reference	Algorithm	Comments
[Indyk-Vakilian-Yuan, NeurIPS'19]	IVY	Described earlier
[Ailon-Leibovich-Nair, UAI'21]	Butterfly LRA	Learns partial Fourier-like matrix instead of random sparse matrix
[Liu-Liu-Vakilian-Wan-Woodruff'20]	Multisketch LRA	Learns **locations** of non-zero entries, not just values
[Indyk-Wagner-Woodruff, NeurIPS'21]	Few-shot LRA	Faster but less accurate training
[Bartlett-Indyk-Wagner, COLT'22]	Lower bound	Lower bound for sample complexity

Faster training

"Logo" dataset



Legend:

- IVY (2019)
- Butterfly (2021)
- Few shot algorithms (2021)
- Few shot + IVY (2021)

Sampling complexity

- Can we **provably** learn good algorithms from past inputs?
- Gupta & Roughgarden (2016):
  - View as statistical learning problem
  - Prove upper bounds on fat shattering dimension (real-valued analog VC dimension)

  ⇒ PAC-learning generalization bounds on number of training samples

- **This work**: Bounds for data-driven numerical linear algebra algorithms

Data-Driven Algorithms: Setting

A loss minimization problem:

- Inputs: $x \in X$
- Algorithms: $\mathcal{L} = \{L_\rho : \rho \in \mathbb{R}^n\}$, parameterized by $\rho \in \mathbb{R}^n$
- Losses: Identify L_ρ with a map $L_\rho : X \to [0,1]$ that maps inputs to losses
  - $L_\rho(x)$ is the loss of solving for x with parameters ρ

Our case: The low-rank approximation (LRA) problem

- Inputs: X is the set of matrices $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ with $\|\boldsymbol{A}\|_F = 1$
- Algorithms: $\mathcal{L} = \{L_{\boldsymbol{S}} : \boldsymbol{S} \in \mathbb{R}^{m \times n}\}$, parameterized by auxiliary matrices $\boldsymbol{S} \in \mathbb{R}^{m \times n}$
- Loss: $L_{\boldsymbol{S}}(\boldsymbol{A}) = \left\|\boldsymbol{A} - \widetilde{\boldsymbol{A}}_{\boldsymbol{S}}\right\|_F^2$, where $\widetilde{\boldsymbol{A}}_{\boldsymbol{S}}$ is the LRA of $\boldsymbol{A}$ computed with aux. matrix $\boldsymbol{S}$

Statistical Learning and ERM

Statistical learning: Suppose we have a distribution D over X

- Goal: Estimate the best parameters for D

$$\rho^* = \text{argmin}_{\rho \in \mathbb{R}^n} \mathbb{E}_{x \in D}\left[L_\rho(x)\right]$$

- Method: Draw s samples $x_1, \dots, x_s \sim D$ and use Empirical Risk Minimization (ERM)

$$\hat{\rho} = \text{argmin}_{\rho \in \mathbb{R}^n} \frac{1}{s} \sum_{i=1}^{s} L_\rho(x_i)$$

- We say $\mathcal{L} = \{L_\rho : \rho \in \mathbb{R}^n\}$ is (ϵ, δ)-*learnable* with s samples (by ERM) if

$$\Pr_{x_1,\dots,x_s \sim D}\left[\mathbb{E}_{x \in D}\left[L_{\hat{\rho}}(x)\right] \leq \mathbb{E}_{x \in D}\left[L_{\rho^*}(x)\right] + \epsilon\right] \geq 1 - \delta$$

- ***Question****: What is the smallest number of samples s that suffices?*

VC-Dimension and Fat Shattering Dimension

Definition: Let $\mathcal{L}$ be a family of functions $X \to \{0,1\}$.

- A set $x_1, \dots, x_s \in X$ is **shattered** by $\mathcal{L}$ if for every $I \subset \{1, \dots, s\}$, there is $L \in \mathcal{L}$ s.t.:

$$L(x_i) = 1 \Leftrightarrow i \in I.$$

- The **VC-dimension** $\mathrm{VCdim}(\mathcal{L})$ of $\mathcal{L}$ is the size of the largest shattered set.

Definition: Let $\mathcal{L}$ be a family of functions $X \to [0,1]$. Let $\gamma \geq 0$.

- A set $x_1, \dots, x_s \in X$ is **γ-fat shattered** by $\mathcal{L}$ if there are thresholds $r_1, \dots, r_s \in \mathbb{R}$, such that for every $I \subset \{1, \dots, s\}$, there is $L \in \mathcal{L}$ s.t.:

$$i \in I \Rightarrow L(x_i) > r_i + \gamma \quad \text{and} \quad i \notin I \Rightarrow L(x_i) < r_i - \gamma$$

- The **γ-fat shattering dimension** $\mathrm{fat}_\gamma(\mathcal{L})$ of $\mathcal{L}$ is the size of the largest γ-fat shattered set.

Classical learning theory: The sample complexity of (ϵ, δ)-learning $\mathcal{L}$ (by ERM) is proportional to the γ-fat shattering dimension with $\gamma = \Theta(\epsilon)$.

Our Results

Theorem – Fat shattering dimension of SCW:

- Upper bound: The ϵ-fat shattering dimension of learned SCW is

$$O\big(n \cdot (m + k\log(n/k) + \log(1/\epsilon))\big),$$

  with $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{S} \in \mathbb{R}^{m\times n}$, and low rank k.
- Lower bound: The ϵ-fat shattering dimension of learned SCW is $\Omega(n)$, if $\epsilon < 1/(2\sqrt{k})$.
- Techniques apply to other data-driven linear algebra algorithms.

Proof Overview

Starting point – the Goldberg-Jerrum (1995) theorem (informal statement):

- Suppose we can compute the loss of a set of n parameters on a given input, in time t, using only arithmetic operations and *if* statements.
- Then, the fat shattering dimension is $O(nt)$.

Proof strategy:

1. Refine the Goldberg-Jerrum framework
   - Use more refined complexity measures than the running time.
2. Design an approximation algorithm for the SCW loss using only arithmetic operations and *if* statements, which is efficient under the refined complexity measures.
   - SCW needs two subroutines:
     - Orthogonal projection
     - Best rank-k approximation

Conclusions

- We showed learned sketches can improve the accuracy/measurement tradeoff for low-rank approximation
- Can retain some guarantees by mixing random and learned rows
- Training still takes time but doable
- Bounds on sampling complexity
- Questions:
  - Provably minimize the loss function ?

More about learning-based algorithms

- A 2021 workshop on learning-based algorithms, organized by Foundations of Data Science Institute (FODSI)
- Talks available at https://www.youtube.com/c/MIFODS

Thanks!

Confirmed speakers

- Alex Dimakis (UT Austin)
- Yonina Eldar (Weizmann)
- Anna Goldie (Google Brain, Stanford)
- Reinhard Heckel (Technical University of Munich)
- Stefanie Jegelka (MIT)
- Tim Kraska (MIT)
- Benjamin Moseley (CMU)
- David Parkes (Harvard)
- Ola Svensson (EPFL)
- Tuomas Sandholm (CMU, Optimized Markets, Strategy Robot, Strategic Machine)
- Sergei Vassilvitski (Google)
- Ellen Vitercik (CMU/UC Berkeley)
- David Woodruff (CMU)