Sparse and Low-Rank: Resource-Efficient Methods in Machine Learning

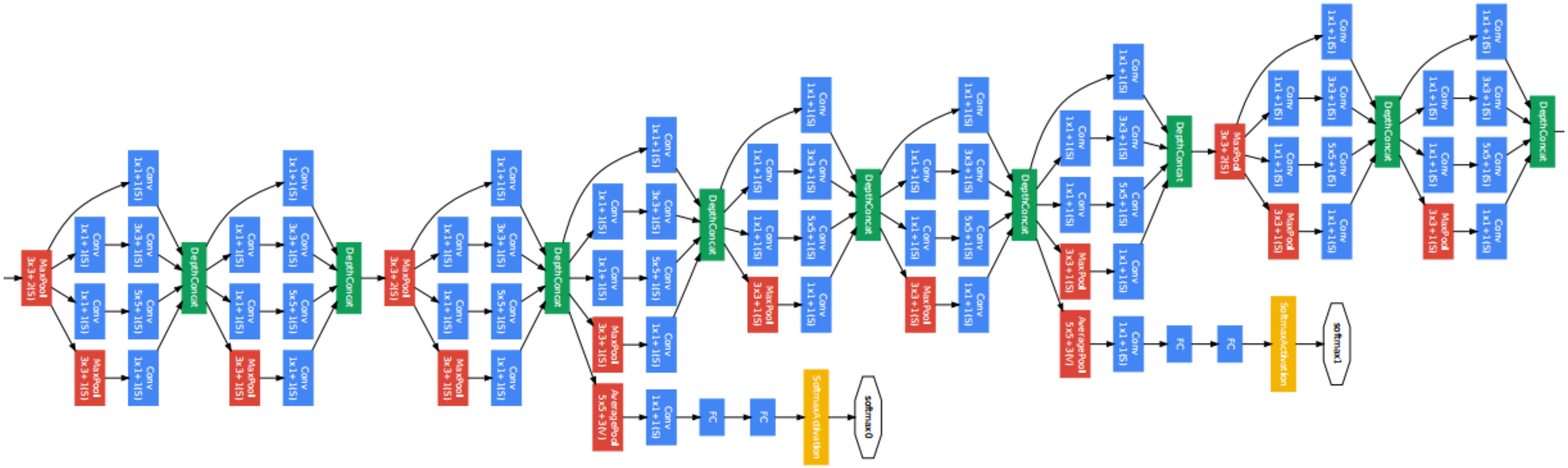
Kiran Vodrahalli

Columbia University

January 11, 2022

Resource-Efficient Machine Learning

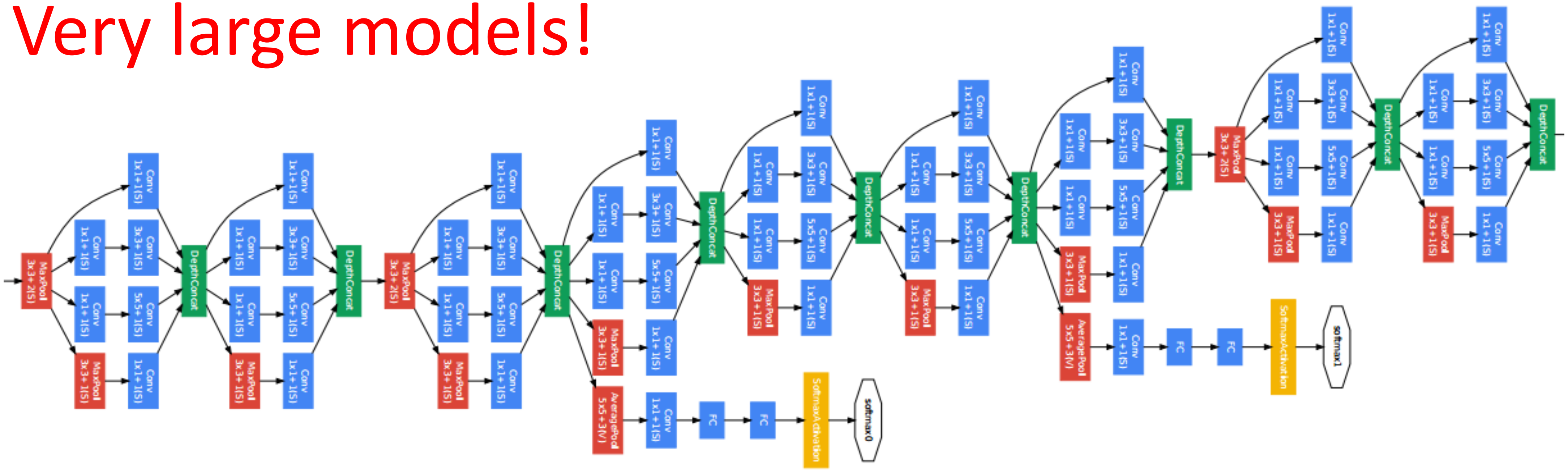
A challenge in modern machine learning:



Resource-Efficient Machine Learning

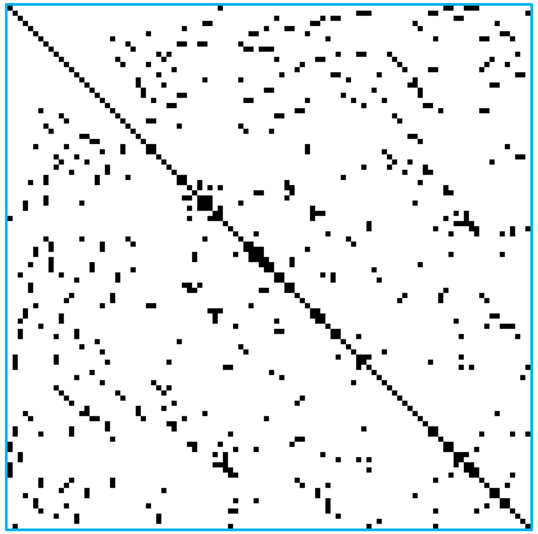
A challenge in modern machine learning:

Very large models!



Resource-Efficient Machine Learning

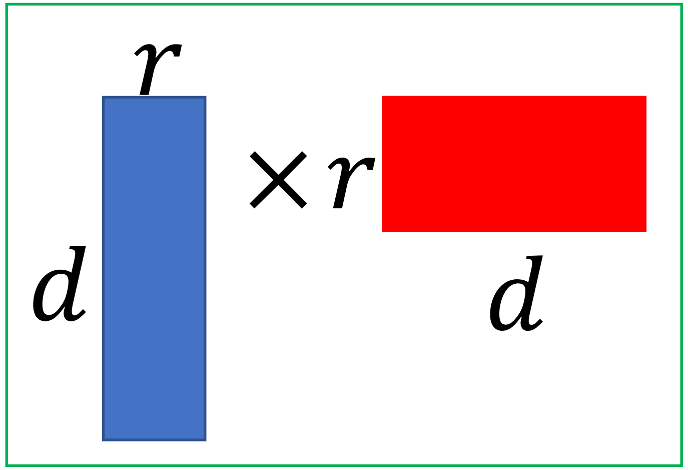
Can we mitigate computational/statistical strains of large nonlinear models via

Sparse models?: $\sigma($  $x)$

Resource-Efficient Machine Learning

Can we mitigate computational/statistical strains of large nonlinear models via

Low-rank models?: $\sigma($



$x)$

Outline of the Talk

1. **Sparse** machine learning for monomials

2. **Low rank** deep learning

3. Future research plans

Sparse Machine Learning

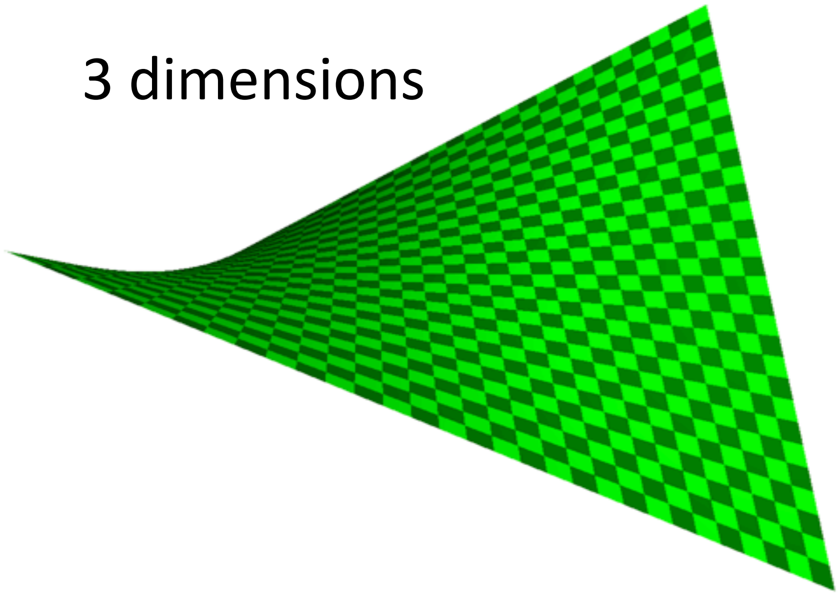
(Attribute-Efficient Learning of Monomials over Highly-Correlated Variables)

Alexandr Andoni, Rishabh Dudeja, Daniel Hsu, **Kiran Vodrahalli**

ALT 2019

Learning Sparse Monomials

Sparse high-dim
log-log regression

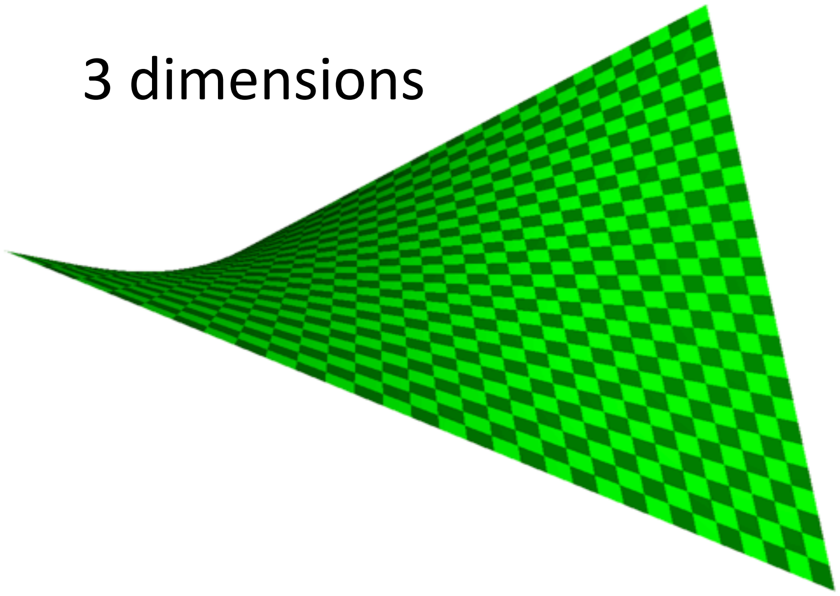


In p dimensions
and k sparse

Ex: $f(x_1, \ldots, x_p) := \underbrace{x_3 \cdot x_{17} \cdot x_{44} \cdot x_{79}}_{k=4}$

Learning Sparse Monomials

A Simple
Nonlinear
Function Class



In p dimensions
and k sparse

Ex: $f(x_1, \ldots, x_p) := \underbrace{x_3 \cdot x_{17} \cdot x_{44} \cdot x_{79}}_{k=4}$

The Learning Problem

Given: $\left\{\left(\boldsymbol{x}^{(i)}, f\left(\boldsymbol{x}^{(i)}\right)\right)\right\}_{i=1}^{m}$, drawn i.i.d.

Assumption 1: f is a k-sparse monomial function

Assumption 2: $\boldsymbol{x}^{(i)} \sim \mathcal{N}(0, \Sigma)$

Goal: Recover f exactly

Attribute-Efficient Learning

- Sample efficiency: $m = \text{poly}(\log(p), k)$

- Runtime efficiency: $\text{poly}(p, k, m)$ ops

- Goal: achieve both!

Motivation

$x_i \in \{\pm 1\}$

- Monomials $\equiv$ Parity functions
- No attribute-efficient algs!
  [Helmbold+ '92, Blum'98, Klivans&Servedio'06, Kalai+'09, Kocaoglu+'14…]

$x_i \in \mathbb{R}$

- Sparse sums of monomials
  [Andoni+'14]

For **uncorrelated** features:

$$\mathbb{E}[xx^T] = \begin{pmatrix} \sigma_1^2 & & & & & \\ & \sigma_2^2 & & & 0 & \\ & & \sigma_3^2 & & & \\ & & & \sigma_4^2 & & \\ & 0 & & & \sigma_5^2 & \\ & & & & & \sigma_6^2 \end{pmatrix}$$

Motivation

$x_i \in \{\pm 1\}$

- Monomials
- No attribut

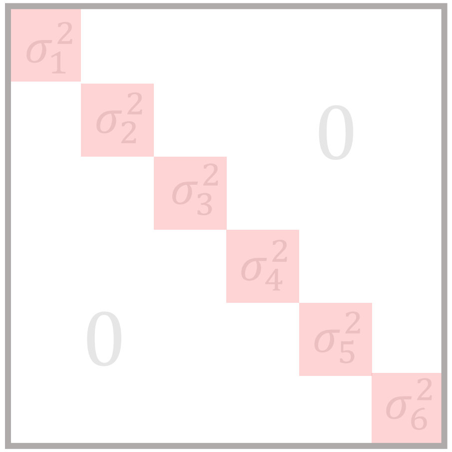
[Helmbold+ '92, B
Kalai+'09, Kocaog

$x_i \in \mathbb{R}$

regression
+'04, Bickel+'09...]

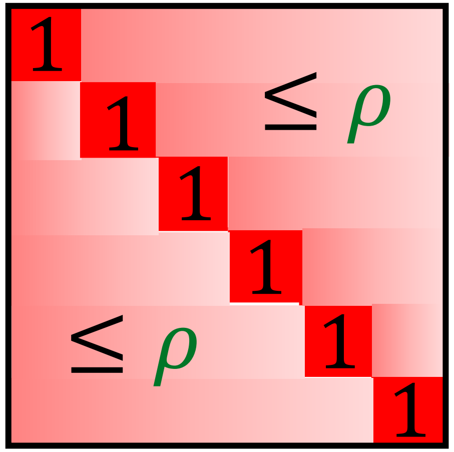
omials

ated features:



Question: What if

$$\mathbb{E}[xx^T] =$$



?

Outline for This Project

1. Algorithm
2. Analysis
3. Conclusion

1. Algorithm

The Algorithm

$$\text{Ex: } f(x_1, \dots, x_p) \coloneqq x_3 \cdot x_{17} \cdot x_{44} \cdot x_{79}$$

Step 1

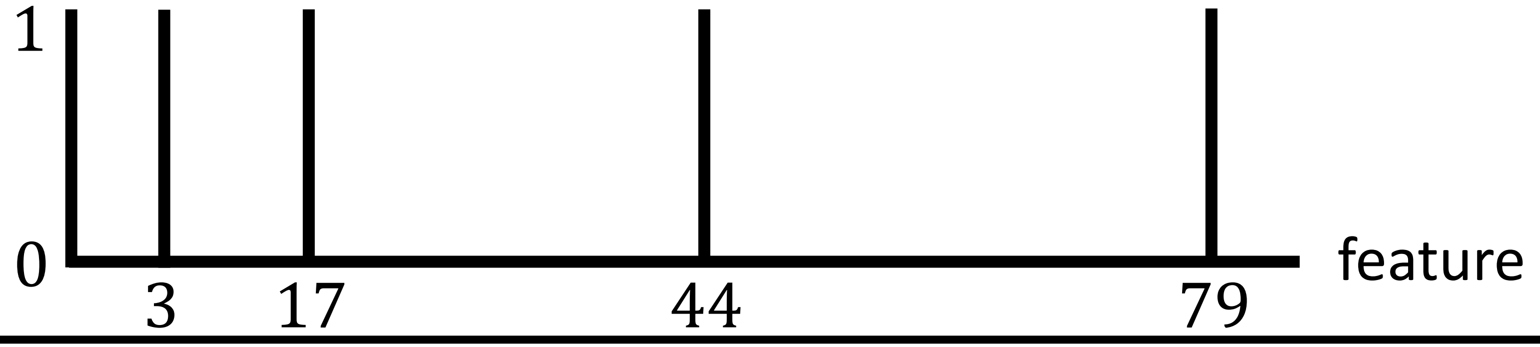
$$\left\{\left(\boldsymbol{x}^{(i)}, f\left(\boldsymbol{x}^{(i)}\right)\right)\right\}_{i=1}^{m} \xrightarrow{\log|\cdot|} \left\{\left(\log\left|\boldsymbol{x}^{(i)}\right|, \log\left|f\left(\boldsymbol{x}^{(i)}\right)\right|\right)\right\}_{i=1}^{m}$$

Gaussian Data → Log-transformed Data

Step 2

Sparse Regression:

(Ex: Basis Pursuit)



Why is our Algorithm Attribute-Efficient?

- Runtime: basis pursuit is efficient

- **Key Question:** Sample complexity
  - Sparse **linear** regression analysis on transformed vars? E.g.:

$$\log\left|f\left(x_1, \ldots, x_p\right)\right| := \log|x_3| + \log|x_{17}| + \log|x_{44}| + \log|x_{79}|$$

  - **To prove:** sparse linear regression recovery holds

2. Analysis

Restricted Eigenvalue Condition [Bickel, Ritov, & Tsybakov '09]

Restricted Eigenvalue $RE(k)$

$$\min_{v \in C} \frac{v^T X X^T v}{||v||_2^2} > \epsilon$$

"restricted strong convexity"

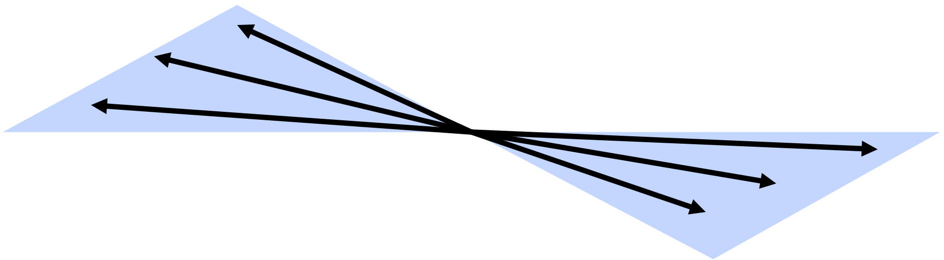
Note: $RE(k) \geq \lambda_{min}(XX^T)$

Sufficient to prove exact recovery for basis pursuit in sparse linear regression!

Ex: $S = \{3, 17, 44, 79\}$

$k = 4$

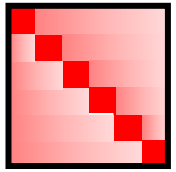
Cone restriction



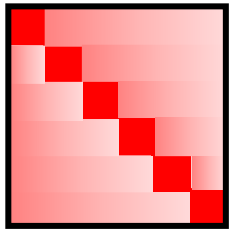
$C = \{v \colon ||v_S||_1 \geq ||v_{S^c}||_1\}$

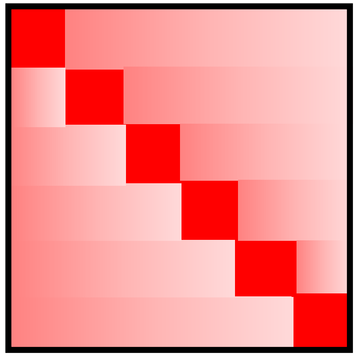
$|S| = k$

Degenerate High Correlation

 $= \mathbb{E}[xx^T]$

Consider the example:

 $= \begin{bmatrix} 1 & 0 & \sqrt{.5} & & & \\ 0 & 1 & \sqrt{.5} & & \mathbf{0} & \\ \sqrt{.5} & \sqrt{.5} & 1 & & & \\ & & & \ddots & & \\ & \mathbf{0} & & & 1 & 0 \\ & & & & 0 & 1 \end{bmatrix}$

 $\begin{bmatrix} -1/2 \\ -1/2 \\ 1/\sqrt{2} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{0}$

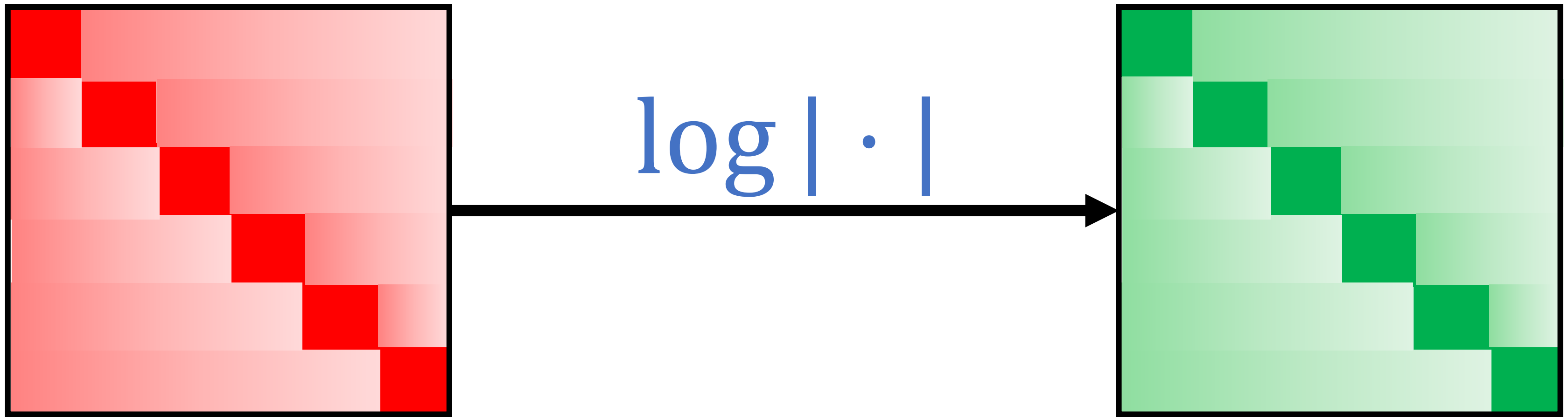
3-sparse

0-eigenvectors can be k-sparse

↓

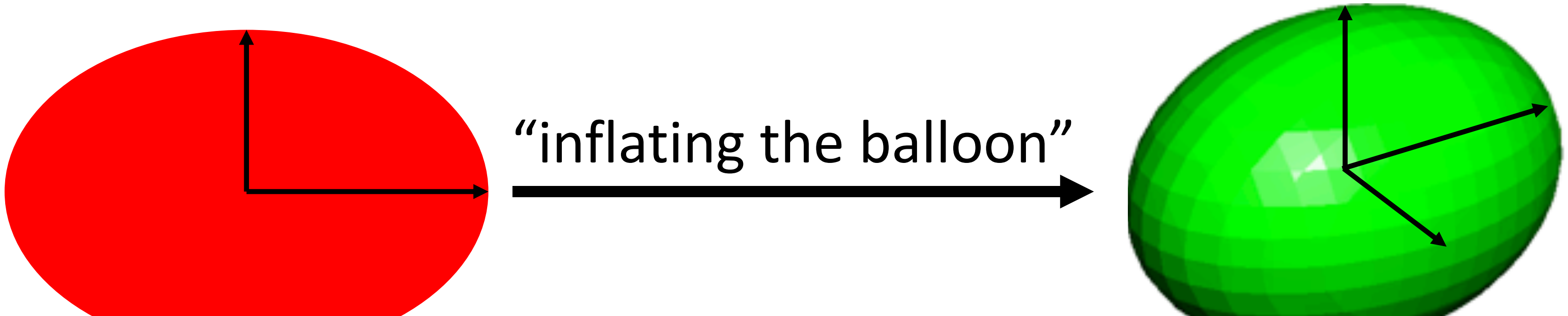
Sparse recovery conditions false!

Log-Transform affects Data Covariance

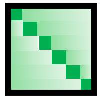


$\mathbb{E}[xx^T] \succcurlyeq 0$ $\qquad$ $\mathbb{E}[\log|x| \log|x|^T] \succ 0$

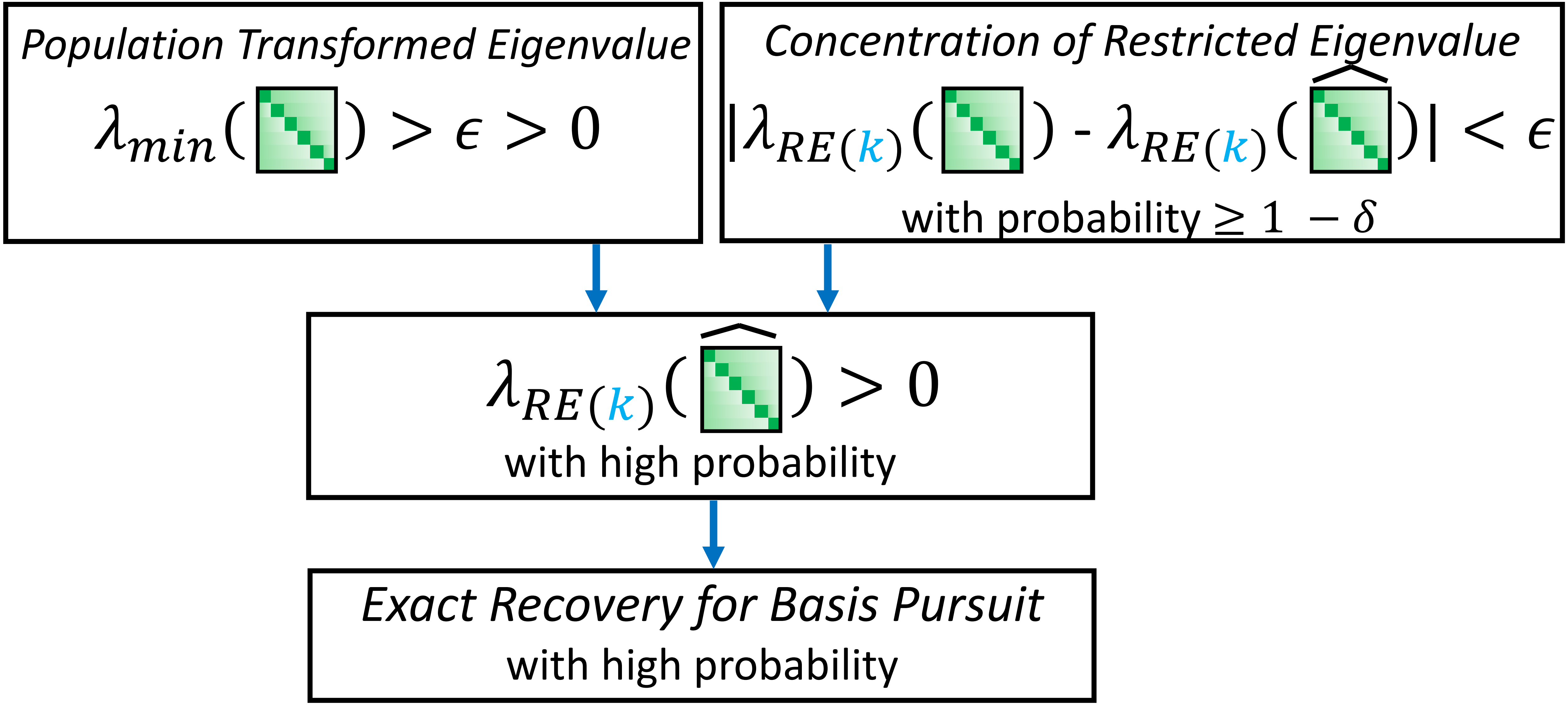
Spectral View:



Destroys correlation structure

 $= \mathbb{E}[\log|x| \log|x|^T]$

Sample Complexity Analysis



$= \mathbb{E}[\log|x| \log|x|^T]$

Sample Complexity Analysis

Population Transformed Eigenvalue

$\lambda_{min}(\quad) > \epsilon > 0$

Concentration of Restricted Eigenvalue

$|\lambda_{RE(k)}(\quad) - \lambda_{RE(k)}(\widehat{\quad})| < \epsilon$

ility $\geq 1 - \delta$

Sample Complexity Bound:

$$m = O\left(\frac{k^2 \log 2k}{1-\rho} \cdot \log^2 \frac{2p}{\delta}\right)$$

with high probability

Exact Recovery for Basis Pursuit

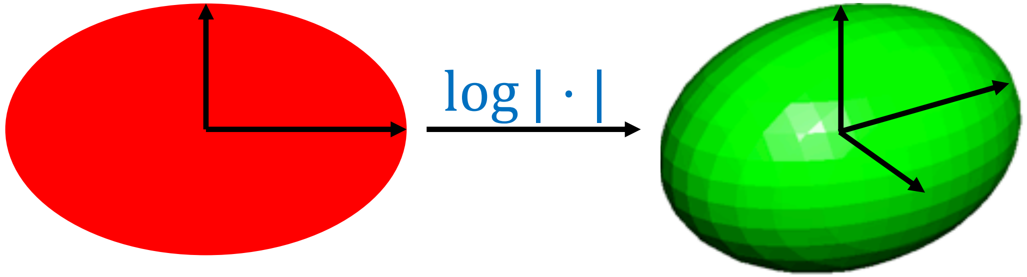
with high probability

3. Conclusion

Recap

- Attribute-efficient algorithm for **monomials**
  - Prior (nonlinear) work: **uncorrelated** features
  - This work: allow highly **correlated** features
    - Works beyond multilinear monomials

- Blessing of nonlinearity



Low Rank Deep Learning

Kiran Vodrahalli, Rakesh Shivanna, Mahesh Sathiamoorthy, Sagar Jain, Ed Chi

Google Brain Research Internship (Summer + Fall 2021)

Speeding Up Deep Nets

- Deep neural networks are extremely large today.
- Goal: speed up forward and backward passes
- Example approach: sparse deep networks

Low Rank Deep Models

Replace full-rank layers with low-rank equivalents:

Given weights of layer i:

$$W_i = U_i V_i^T$$

Then replace the standard parameterization w/RHS.

Prior Work [Khodak et. al. 2021]

- Low-rank methods outperform sparse methods if tuned correctly

- Key issues:
  1) Initialization of the low-rank parameters
  2) Regularization of weights

Impact of Initialization

- Can achieve ~ 1% additive gain in accuracy by choosing better initialization
- Khodak et al 2021 studies small image classification datasets
- Key approach: **spectral initialization**

Outline for This Project

1. Low-Rank Initialization Scheme

2. Theory

1. Low-Rank Initialization Scheme

Spectral Initialization

For each layer $W \in R^{m \times n} \sim D$:

Minimize the Frobenius distance between the full-rank initialization **parameters** and the low rank parameters:

$$\min_{U \in R^{m \times r}, V \in R^{n \times r}} ||W - UV^T||_F^2$$

Generalized Spectral Initialization

For each layer $W \in R^{m \times n} \sim D$ with nonlinearity σ:

Perform **distillation**: Minimize ℓ_2^2 error between the **function** outputs of full-rank initialization and low rank initialization:

$$\min_{U \in R^{m \times r}, V \in R^{n \times r}} E_{x \sim N(0,I)}[||\sigma(Wx) - \sigma(UV^T x)||_2^2]$$

Generalized Spectral Initialization

For each layer $W \in R^{m \times n} \sim D$ with nonlinearity σ:

Perform **distillation**: Minimize ℓ_2^2 error between the **function** outputs of full-rank initialization and low rank initialization:

Results in empirical gains over spectral initialization!

$$\min_{U \in R^{m \times r}, V \in R^{n \times r}} E_{x \sim N(0,I)}[||\sigma(Wx) - \sigma(UV^T x)||_2^2]$$

2. Theory

Is GSpectral Initialization Tractable?

- Suppose $W_{ij} \sim N\left(0, \frac{1}{\sqrt{n}}\right)$ for layer weight $W \in R^{m \times n}$
- Define $f_W(x) \coloneqq \sigma(Wx)$
- Given $D_W = \left\{\left(x, f_W(x)\right): x \sim N(0, I_{n \times n})\right\}$, find $\widehat{U}, \widehat{V}$ s.t.

$$\mathbb{E}_{x \sim \mathcal{N}(0, I_{n \times n})} \left[\left\| \sigma(\hat{U}\hat{V}^T x) - f_W(x) \right\|_2^2 \right] < \mathrm{OPT} + \epsilon$$

Is GSpectral Initialization Tractable?

For GSpectral algorithm to be efficient:

- Return $\widehat{U}, \widehat{V}$ w/ $opt + \epsilon$ error w/ prob $\geq \frac{3}{4}$
- $\epsilon = \frac{1}{10}$
- Runtime poly(n)

Related Work

ReLU Regression (additive square loss, learn $\sigma(w^T x)$)

- Realizable case w/Gaussian x: Gradient Descent succeeds
  [Soltanolkotabi 2017]

- Agnostic case w/Gaussian x : GD (+ any SQ algorithm) achieving squared loss generalization error $opt + \epsilon$ requires $\exp(\Theta(n^c))$ statistical queries or $n^{\Theta\left(\left(\frac{1}{\epsilon}\right)^{2b}\right)}$ samples per query for some $b, c \in (0, ½)$. Also the basic problem is SPWN-hard.
  [Goel et. al. 2019, 2020]

- Agnostic case w/log-concave x: $O(opt) + \epsilon$ has polytime algo
  [Diakonikolas et. al. 2020]

Our Setting

- Combines average-case weights and Gaussian data

- Not realizable, but not arbitrary output distribution

- Common assumptions from theory are practical!

Main Results Teaser

- There is an efficient algorithm for constant rank!

- **Algorithm:**
  1. Exact algorithm in runtime poly(n) when W is given
  2. Recover W from samples via m realizable ReLU regressions.

Main Results Teaser

- Suppose width is super-linear in dimension.

- High-dimension + Low Rank
  - Then, gap between GSpectral and Spectral grows with dimension.

Future Research Plans

Resource-Efficient Sequence Modeling

- Sequence modeling / time series abound in the sciences and ML
  - Language modeling
  - Brain recordings
  - ….
- Broadly useful simulation/modeling tool
- Limitation of modern deep sequence models: small context!

Resource-Efficient Sequence Modeling

Key Question 1:

Can we employ techniques from low-rank and sparse modeling to achieve **long-range context** neural sequence models in **sublinear space and time**?

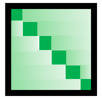
Resource-Efficient Sequence Modeling

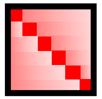
Key Question 2:

Can we employ techniques from sketching/streaming theory to analyze **long-range context** neural sequence models in **sublinear space and time**?

For reference

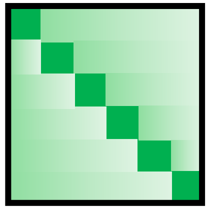
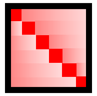
Monomials

 $= \mathbb{E}[\log|x| \log|x|^T]$

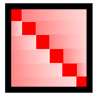
 $= \mathbb{E}[xx^T]$

Population Minimum Eigenvalue

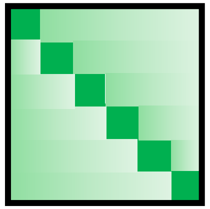
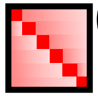
- Hermite expansion of $\log|\cdot|$:

 $= c_0^2 1_{pxp} + \sum_{l=1}^{\infty} c_{2l}^2$  $^{(2l)}$

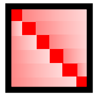
- $l \geq 1$: $c_{2l}^2 \sim \frac{\sqrt{\pi}}{4} \cdot \frac{1}{l^{3/2}}$

-  $^{(2l)}$ off-diagonals decay fast!

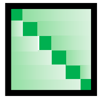
- Apply λ_{min} to Hermite formula:

λ_{min}  $\geq \sum_{l=1}^{\infty} c_{2l}^2 \lambda_{min}$  $^{(2l)}$

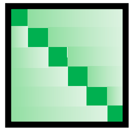
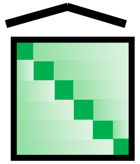
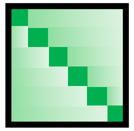
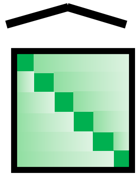
- Apply Gershgorin Circle Theorem:

λ_{min}  $^{(2l)} \geq 1 - (p-1)\rho^{2l}$

(for large enough l)

 $= \mathbb{E}[\log|x| \log|x|^T]$

Concentration of Restricted Eigenvalue

- $|\lambda_{RE(k)}($  $) - \lambda_{RE(k)}($  $)| < k \cdot ||$  $-$  $||_\infty$

- Log-transformed variables are **sub-exponential**

- Elementwise ℓ_∞ error concentrates
  - [Kuchibhotla & Chakrabortty '18]