Exponentially convergent stochastic k-PCA *without* variance reduction

Cheng Tang
Amazon (AWS AI Labs)

Poster ID: #200
Thu Dec 12th 10:45 AM -- 12:45 PM @ East Exhibition Hall B + C #200

Principal Component Analysis — brief overview

Over 100 years of history (older than computers!)

- **[Pearson 1901]: On lines and planes of closest fit to systems of points in space**
- **[Hotelling 1933]: Analysis of a complex of statistical variables into principal components**

Still widely used with increased scalability

- ⋆ **Principled approach to compress high-dimensional data efficiently**
  - ⋆ **Minimize information loss & Increase data interpretability**

- ⋆ **Many variants that adapts to different data and paradigms**
  - ⋆ **Sparse PCA,**
  - ⋆ **Robust PCA,**
  - ⋆ **Streaming PCA,**
  - ⋆ **Stochastic and Online PCA,**
  - ⋆ **Distributed PCA**

See [Joeliffe-Cadima 2016] for a review

Stochastic k-PCA

The traditional view of PCA
(batch PCA)

- View PCA as finding the variance maximizing k-dimensional subspace on a fixed dataset
- Can be done by SVD

PCA as a learning problem
(stochastic PCA)

- View PCA as finding the variance maximizing subspace of the population covariance matrix
- Have i.i.d. sample access of unknown population distribution
  - Stochastic PCA = Stochastic optimization for the PCA learning problem

[Arora-Cotter-Livescu-Srebro, 2012]
[Arora-Cotter-Srebro, 2013]
[Balsubramani-Dasgutpa-Freund, 2013]
[DeSa-Olukotun-Re, 2014]
[Shamir, 2016]

Stochastic PCA objective functions

1-PCA objective

For a centered r.v. $X \in \mathbb{R}$

$$\max_{w \in \mathbb{R}^d, \|w\|=1} var(w^T X) \quad \text{(Raleigh quotient)} \qquad \text{OR} \qquad \min_{w \in \mathbb{R}^d, \|w\|=1} \mathbb{E}\|X - ww^T X\|^2$$

k-PCA objective

$$\min_{W \in \mathbb{R}^{k \times d}, WW^\top = I_k} \mathbb{E}\|X - W^\top W X\|^2$$

Property of the k-PCA objective

- ⋆ **Non-convex**
- ⋆ **Has closed form solution (top k eigenvectors of data covariance matrix)**
  - ⋆ **But cannot be directly computed**
- ⋆ **= One hidden layer linear auto-encoder**

Two popular stochastic 1-PCA algorithms & their convergence rates

Oja's rule (1982)

☆ Online version of power method (has a *length normalization* step)

$$w^t \leftarrow \frac{w^{t-1} + \eta^t x x^T w^{t-1}}{\|w^{t-1} + \eta^t x x^T w^{t-1}\|}$$

☆ Converges in order O(1/t) for 1-PCA
[TODO: add more citations]

Krasulina's method (1969)

★ Stochastic gradient ascent on Rayleigh quotient for 1-PCA

$$w^t \leftarrow w^{t-1} + \eta^t (x x^T - \frac{(x^T w^{t-1})^2}{\|w^{t-1}\|^2} I_d) w^{t-1}$$

★ Converges in order O(1/t) for 1-PCA

[Balsubramani-Dasgupta-Freund 2013]

Two popular online 1-PCA algorithms & their convergence rates

Oja's rule

- ☆ Easy to generalize to k-PCA
- ☆ Converges in order O(1/t) for k-PCA by recent analysis

[Allen-Zhu and Li,FOCS17]
[Shamir, ICML16-a]
[Jain et al, COLT16]
[De Sa et al, ICML15]
[Balsubramani-Dasgupta-Freund 13]

Krasulina's method

- ☆ Not obvious how to generalize to k-PCA

Our observation 1

For 1-PCA, Krasulina's update rule

$$w^t \leftarrow w^{t-1} + \eta^t (xx^T - \frac{(x^T w^{t-1})^2}{\|w^{t-1}\|^2} I_d) w^{t-1}$$

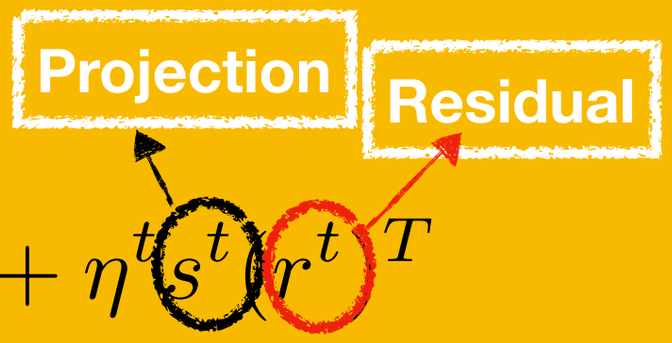
Can be re-written as

$$w^t \leftarrow w^{t-1} + \|w^t\| \eta^t s^{t-1} (r^{t-1})^\top$$

$$s^t = (\frac{w^t}{\|w^t\|})^\top x \qquad r^t = x^\top - s^t (\frac{w^t}{\|w^t\|})^\top$$

Our stochastic k-PCA algorithm— Matrix Krasulina

$W^t \in \mathbb{R}^{k \times d}$ algorithm's approximate to k-PCA subspace



$$W^t \leftarrow W^{t-1} + \eta^t s^t (r^t)^T$$

Orthonormalize rows of W^t

- Matrix Krasulina generalizes (non-obviously) Krasulina's method to k-PCA problem
- We provide two types of convergence rate guarantee for Matrix Krasulina

Our stochastic k-PCA algorithm— Matrix Krasulina

$W^t \in \mathbb{R}^{k \times d}$ algorithm's approximate to k-PCA subspace

Projection

Residual

$$W^t \leftarrow W^{t-1} + \eta^t s^t (r^t)^T$$

$$r^t = x - W^{t-1} s^t$$

$$s^t = W^{t-1} x$$

Orthonormalize rows of W^t

Recall k-PCA loss

$$\mathbb{E}_{WW^\top = I} \|X - W^\top W X\|^2 = \mathbb{E}\|r\|^2$$

Our stochastic k-PCA algorithm— Matrix Krasulina

Projection

Residual

$$W^t \leftarrow W^{t-1} + \eta^t (s^t)(r^t)^T$$

Orthonormalize rows of W^t

Recall k-PCA loss

$$\mathbb{E}_{WW^\top = I} \|X - W^\top W X\|^2 = \mathbb{E}\|r\|^2$$

This implies variance of the algorithm's update decreases as the k-PCA loss decreases

Our observation 2

Matrix Krasulina for k-PCA can be viewed as

$$W^t \leftarrow W^{t-1} + \eta^t s^t (r^t)^T$$

$$Var(s^t r^t) \propto \text{loss}$$

When data is of rank k, $Var(s^t (r^t)^\top) \to 0$

This implies Matrix Krasulina has a natural “variance reduction” effect

Variance Reduction (VR) for k-PCA

- Variance Reduction [Johnson-Zhang, 2013] is a technique developed to speed up convergence of stochastic optimization algorithms for convex problems
  - Typical convergence rate of SGD on (strongly) convex problem is O(1/t)
  - With VR, SGD has exponential convergence

- [Shamir 2016] applied it to the non-convex k-PCA objective
  - VR is applied to Oja's algorithm for 1-PCA generalized to k-PCA
  - It proves that VR+Oja has a per-epoch exponential convergence rate

Drawback of VR + Oja:

- Per epoch, a full gradient needs to be evaluated
- This means the algorithm is not online k-PCA

Our first result on Matrix Krasulina

Theorem 1 (Exponential convergence with constant learning rate)**.** *Suppose assumption Eq.* (2.4) *holds. Suppose the initial estimate* $W^o \in \mathbb{R}^{k' \times d}$ $(k' \geq k)$ *in Algorithm 1 satisfies that, for some* $\tau \in (0, 1)$*,*

$$\Delta^o \leq 1 - \tau \,,$$

Suppose for any $\delta > 0$*, we choose a constant learning rate* $\eta^t = \eta$ *such that*

$$\eta \leq \min \left\{ \frac{\sqrt{2}-1}{b}, \frac{\lambda_k \tau}{\lambda_1 b(k+3)}, \frac{2\lambda_k \tau}{\frac{8}{1-\tau} \ln \frac{1}{\delta}(b + \|\Sigma^*\|_F)^2 + b(k+1)\lambda_1} \right\},$$

Then there exists event $\mathcal{G}_t$ *such that* $\mathbb{P}(\mathcal{G}_t) \geq 1 - \delta$ *, and*

$$\mathbb{E}\left[\Delta^t | \mathcal{G}_t\right] \leq \frac{1}{1-\delta} \exp\left(-t\eta\tau\lambda_k\right) \,.$$

In summary

- Matrix Krasulina has free “variance-reduction” effect on rank-k data
- Matrix Krasulina has exponential convergence towards the true k-PCA subspace
  - This is faster than other state-of-the-art online k-PCA algorithms

Information-theoretic lower bound

- This implies a O(1/t) upper bound on convergence rate of *ALL* online k-PCA algorithms
  - Including stochastic k-PCA algorithms with data coming one-at-a-time

$$\mathbb{E}[\Delta^n] \geq \Omega(\frac{\sigma^2}{n})$$

$$\text{for } \frac{\lambda_1 \lambda_{k+1}}{(\lambda_k - \lambda_{k+1})^2},$$

[Vu and Lei, AISTATS, 2012]

Information-theoretic lower bound — a closer look

- The lower bound to-the-right implies a 1/t upper bound on convergence rate of online k-PCA algorithms **in the general case**

- Our result says otherwise when data is **low-rank**

$$\mathbb{E}[\Delta^n] \geq \Omega(\frac{\sigma^2}{n})$$

$$\text{for } \frac{\lambda_1 \lambda_{k+1}}{(\lambda_k - \lambda_{k+1})^2},$$

[Vu and Lei, AISTATS, 2012]

Our second main result on Matrix Krasulina

Theorem 2 (Linear convergence on full rank data). *Suppose* $\mathbb{P}\left(\sup \|X\|^2 > b\right) = 0$. *Suppose the initial estimate* $W^o \in \mathbb{R}^{k' \times d}$ $(k' \geq k)$ *in Algorithm 1 satisfies* $\Delta^o \leq \frac{1-\tau}{2}$. *Let* $\delta \in (0, \frac{1}{e})$. *If choose learning rate schedule* $\eta^t = \frac{c}{t_o+t}$, *for constants* c, t_o. *Let* $B := \max\left(8(b + \|\Sigma^*\|_F)^2 k, (kb + 2cb^2 + c^2b^3)\lambda_1 \|\Sigma^*\|_F^2\right)$. *If we choose* c, t_o *such that*

$$c > \frac{1}{2\lambda_k \tau} \text{ and } t_o \geq \max\{\frac{64Bc^2 \ln \frac{1}{\delta}}{(\Delta^o)^2}, 1\},$$

Then for any $\delta > 0$, *there exists event* $\mathcal{G}_t$ *such that* $\mathbb{P}(\mathcal{G}_t) \geq 1 - \delta$, *and* $\mathbb{E}[\Delta^t | \mathcal{G}_t] \leq O(\frac{1}{t})$.

In summary

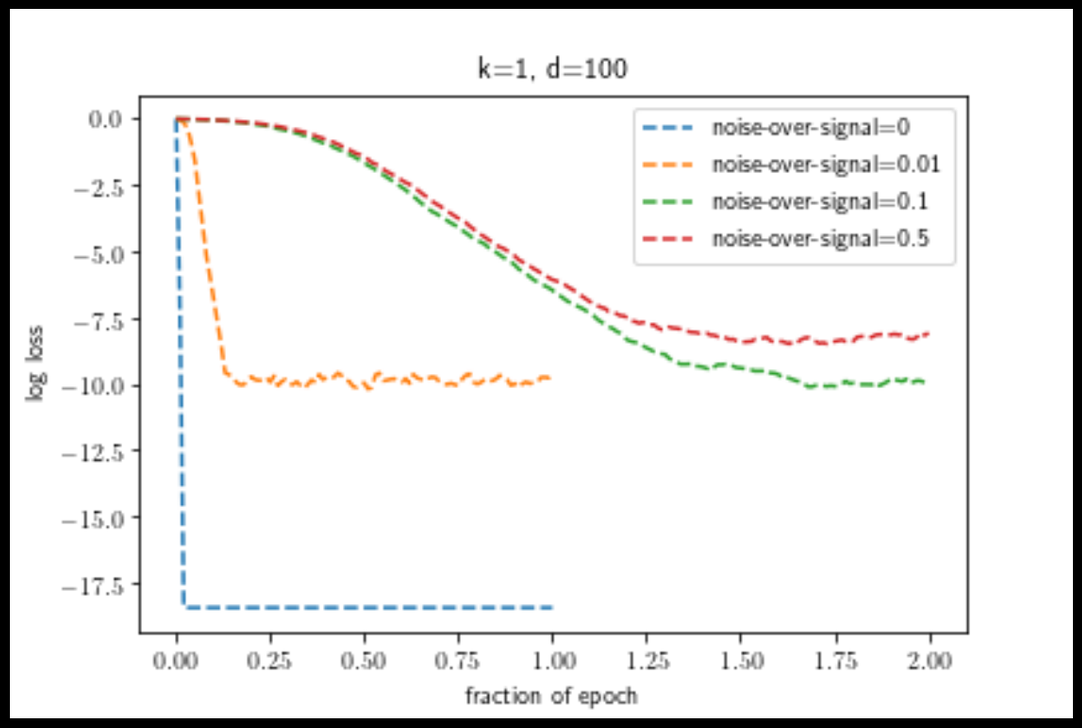
- Matrix Krasulina has O(1/t) convergence rate on full-rank data
- This is comparable to other state-of-the-art online k-PCA algorithm

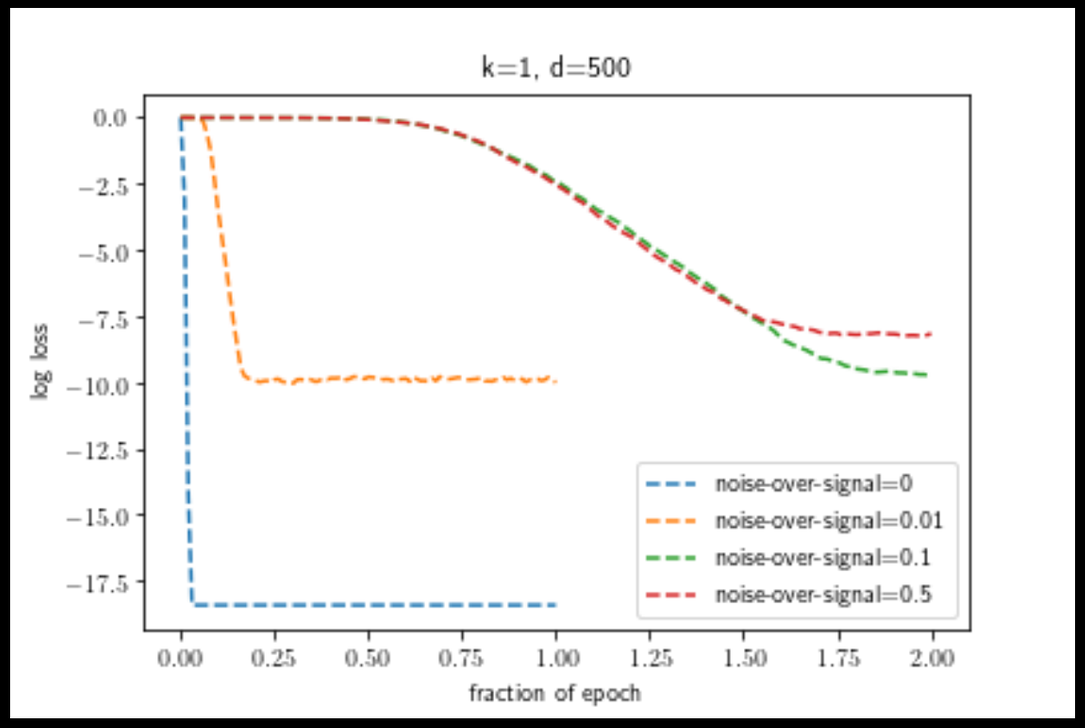
Comparison against other algorithms

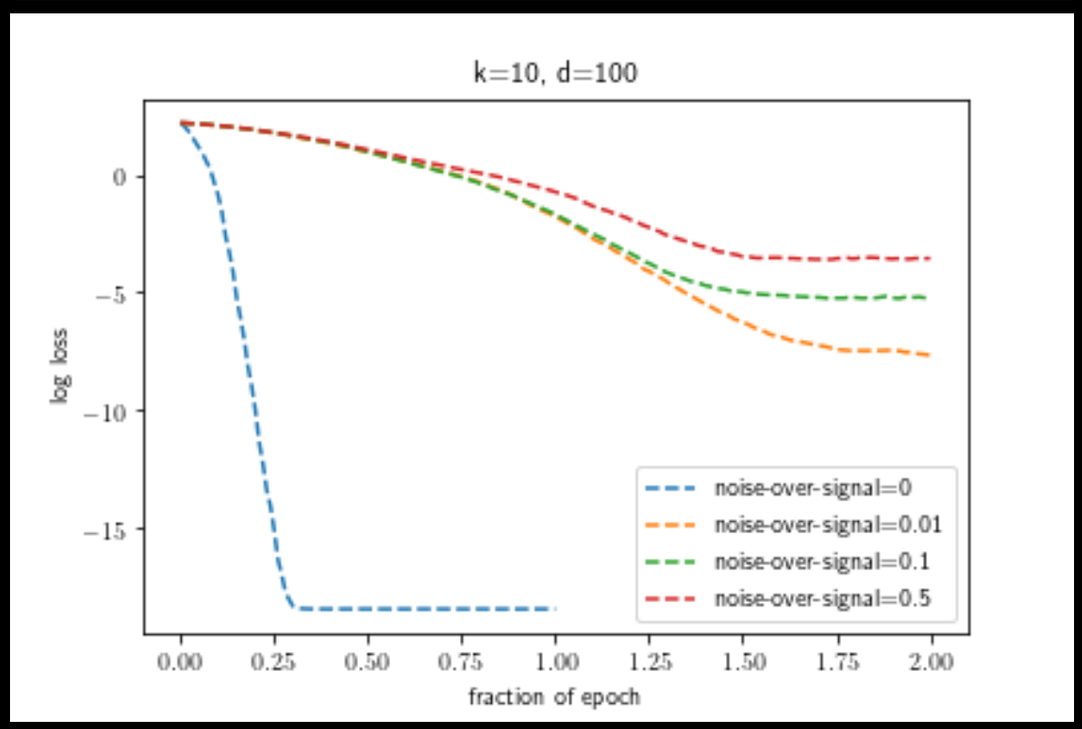
Algorithm	1-PCA	k-PCA, k>=1	Online/Batch
Oja and variants: **[Allen-Zhu&Li,FOCS17]** **[Shamir, ICML16-a]** **[Jain et al, COLT16]** **[De Sa et al, ICML15]**	O(1/t)	O(1/t)	Online/Streaming
VR-PCA **[Shamir, ICML16-b]**	O(exp(-t))	O(exp(-t))	Batch
Our algorithm (Matrix Krasulina)	O(1/t) for general case O(exp(-t)) for rank-1 case	O(1/t) for general case O(exp(-t)) for rank-k case	Online

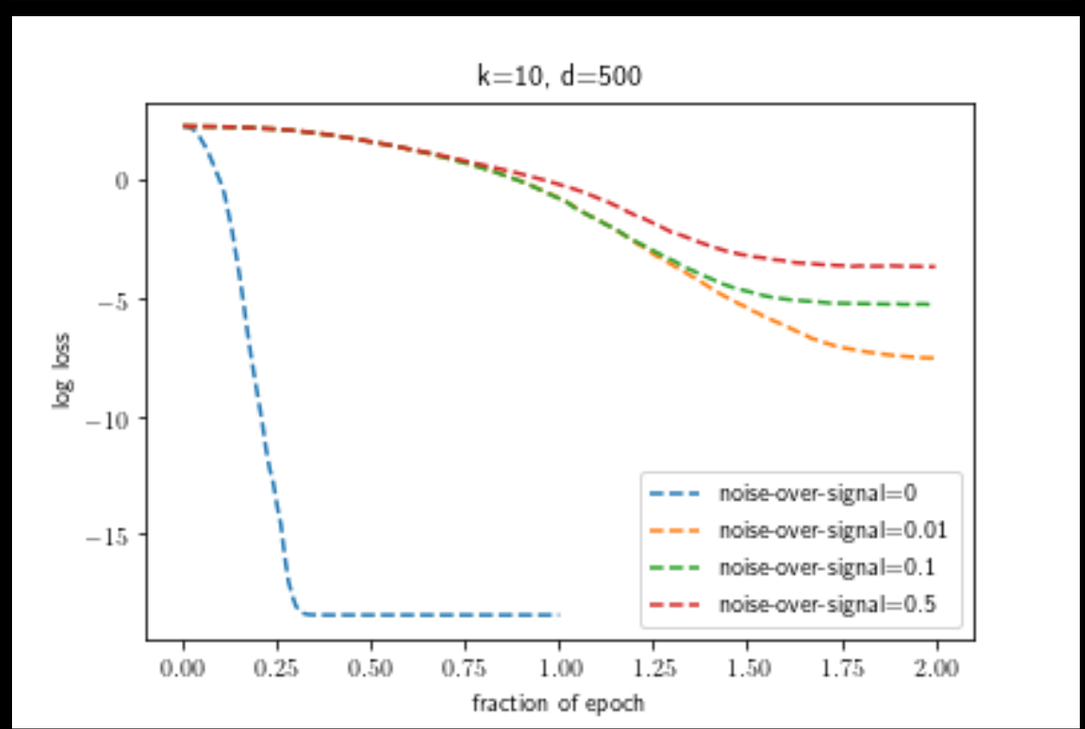
Simulation experiments

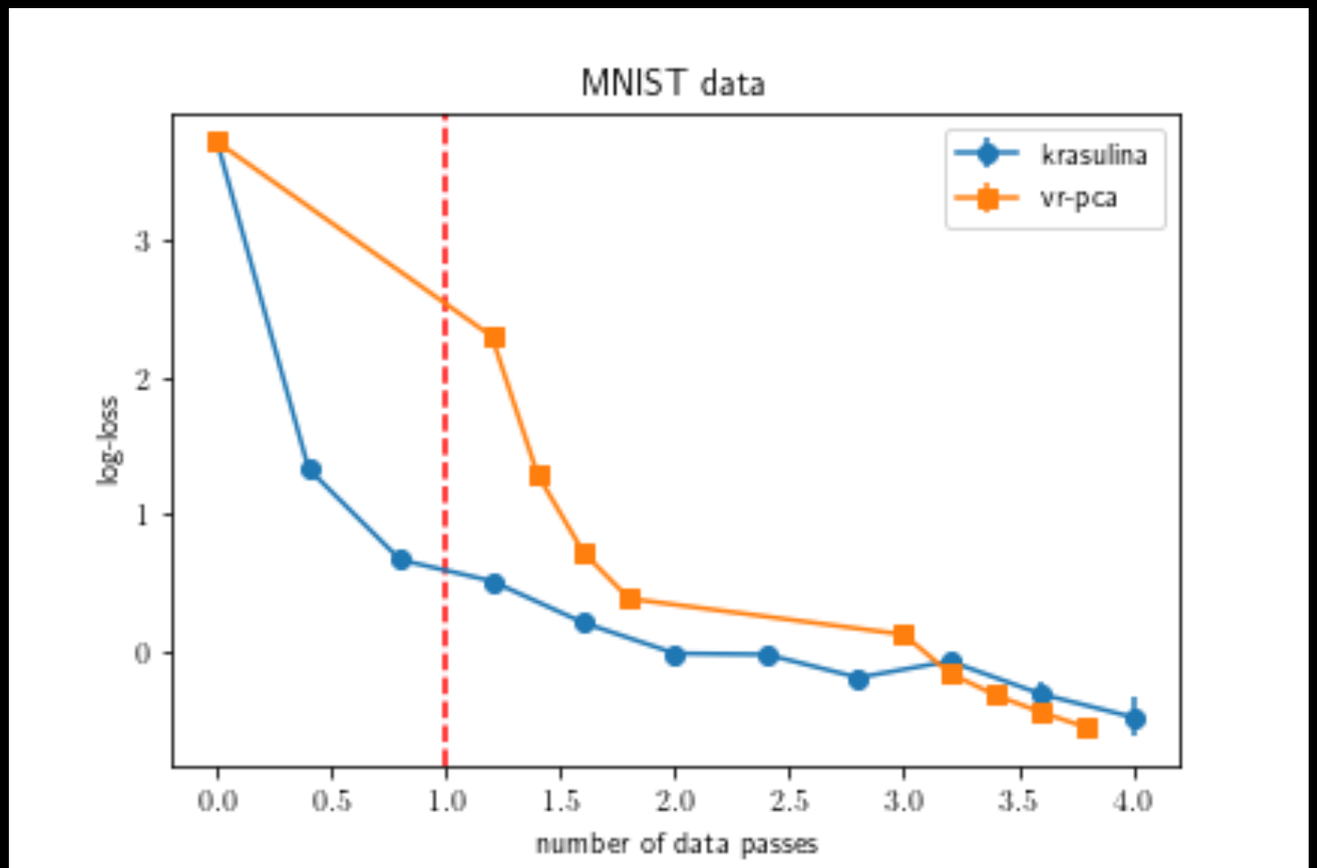
- Validates exponential convergence rate of Matrix Krasulina on rank-k data
- Tests Matrix Krasulina on almost-rank-k data
- Tests robustness to ambient dimension d and intrinsic dimension k

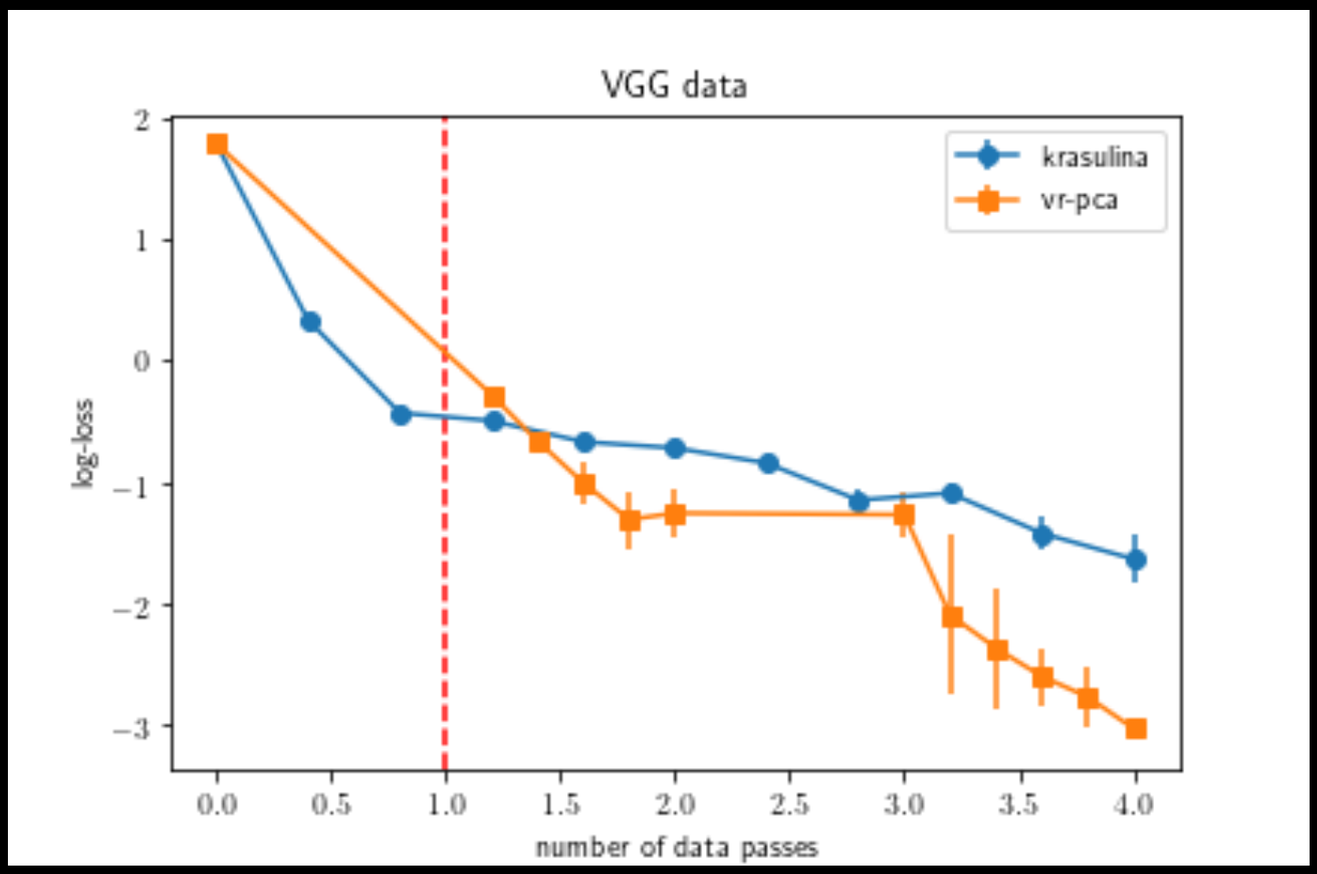










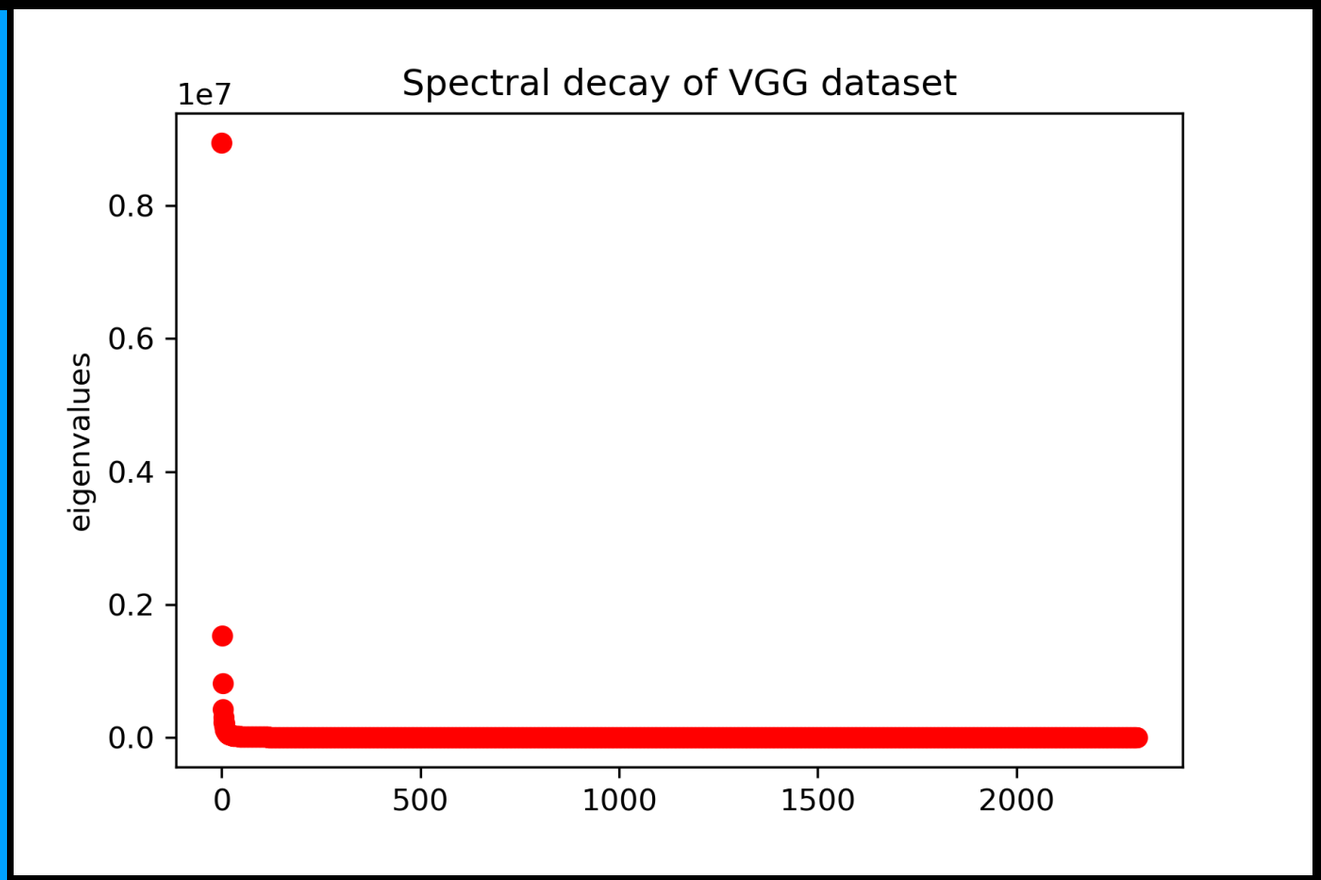


Real-world data experiments

- Shows superiority of Matrix Krasulina over VR-PCA (VR+Oja) in the online setting (data pass <1)
- Shows that Matrix Krasulina still has exponential convergence even if data is not strictly low rank

Open problem

Can we characterize convergence rate when data is *nearly low-rank ?*



Poster ID: #200

Thu Dec 12th 10:45 AM -- 12:45 PM
@ East Exhibition Hall B + C #200