A Stochastic PCA and SVD Algorithm with an Exponential Convergence Rate

Ohad Shamir

Weizmann Institute of Science

ICML
July 2015

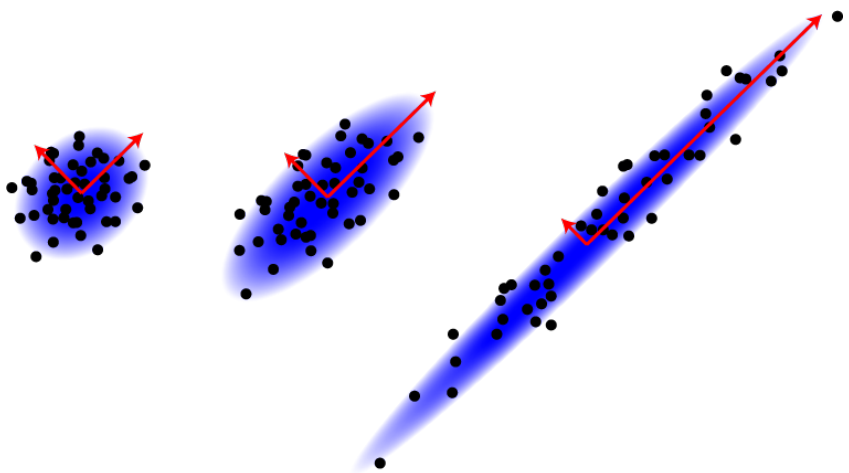
Goal

Given $d \times n$ matrix X, find top k singular vectors:

$$\max_{W \in \mathbb{R}^{d \times k} : W^\top W = I} \left\| X^\top W \right\|_F^2$$

Applications:

- PCA (X is centered data matrix)
- Others too numerous to mention



Existing Approaches

Suppose $k = 1$ for simplicity, and $X = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_n]$
Equivalent to

$$\max_{\mathbf{w}\in\mathbb{R}^d:\|\mathbf{w}\|=1} \mathbf{w}^\top \left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\top\right)\mathbf{w}$$

Existing Approaches

Suppose $k = 1$ for simplicity, and $X = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_n]$
Equivalent to

$$\max_{\mathbf{w}\in\mathbb{R}^d:\|\mathbf{w}\|=1} \mathbf{w}^\top \left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\top\right)\mathbf{w}$$

Approach 1: Eigendecomposition

- Compute leading eigenvector of $\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\top$ exactly (e.g. via QR decomposition)
- Runtime: $\mathcal{O}(\mathbf{d^3} + \mathbf{nd^2})$

Existing Approaches

Approach 2: Power Iterations

- Initialize $\mathbf{w}_1$ at random
- For $t = 1, 2, \ldots$
  - $\mathbf{w}'_{t+1} := \left(\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\top\right)\mathbf{w}_t = \frac{1}{n}\sum_{i=1}^n \langle \mathbf{w}_t, \mathbf{x}_i\rangle \mathbf{x}_i$
  - $\mathbf{w}_{t+1} := \mathbf{w}'_{t+1} / \left\|\mathbf{w}'_{t+1}\right\|$

- $\mathcal{O}\left(\frac{1}{\lambda}\log\left(\frac{1}{\epsilon}\right)\right)$ iterations for ϵ-close solution
  - $\lambda =$ Eigengap
- $\mathcal{O}(\mathbf{nd})$ runtime per iteration
- Overall runtime $\mathcal{O}\left(\frac{\mathbf{nd}}{\lambda}\log\left(\frac{\mathbf{d}}{\epsilon}\right)\right)$

Existing Approaches

Approach 2: Power Iterations

- Initialize $\mathbf{w}_1$ at random
- For $t = 1, 2, \ldots$
  - $\mathbf{w}'_{t+1} := \left(\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\top\right)\mathbf{w}_t = \frac{1}{n}\sum_{i=1}^n \langle \mathbf{w}_t, \mathbf{x}_i\rangle \mathbf{x}_i$
  - $\mathbf{w}_{t+1} := \mathbf{w}'_{t+1} / \left\|\mathbf{w}'_{t+1}\right\|$

- $\mathcal{O}\left(\frac{1}{\lambda}\log\left(\frac{1}{\epsilon}\right)\right)$ iterations for ϵ-close solution
  - $\lambda =$ Eigengap
- $\mathcal{O}(\mathbf{nd})$ runtime per iteration
- Overall runtime $\mathcal{O}\left(\frac{\mathbf{nd}}{\lambda}\log\left(\frac{\mathbf{d}}{\epsilon}\right)\right)$

Approach 2.5: Lanczos Iterations

- More complex algorithm, $\mathcal{O}\left(\frac{1}{\sqrt{\lambda}}\log\left(\frac{1}{\epsilon}\right)\right)$ iterations, but even larger runtime per iteration

Existing Approaches

Approach 3: Stochastic/Incremental Algorithms

Example (Oja's algorithm)

- Initialize $\mathbf{w}_1$ randomly
- For $t = 1, 2, \ldots$
  - Pick $i_t \in \{1, \ldots, n\}$ (randomly or otherwise)
  - $\mathbf{w}'_{t+1} := \mathbf{w}_t + \eta_t \mathbf{x}_{i_t} \mathbf{x}_{i_t}^\top \mathbf{w}_t$
  - $\mathbf{w}_{t+1} := \mathbf{w}'_{t+1} / \left\|\mathbf{w}'_{t+1}\right\|$

Also Krasulina 1969; Arora, Cotter, Livescu, Srebro 2012; Mitliagkas, Caramanis, Jain 2013; Balsubramani, Dasgupta, Freund 2013; Hardt, Price 2014; De Sa, Olukotun, Ré 2015...

Existing Approaches

Approach 3: Stochastic/Incremental Algorithms

Example (Oja's algorithm)

- Initialize $\mathbf{w}_1$ randomly
- For $t = 1, 2, \ldots$
  - Pick $i_t \in \{1, \ldots, n\}$ (randomly or otherwise)
  - $\mathbf{w}'_{t+1} := \mathbf{w}_t + \eta_t \mathbf{x}_{i_t} \mathbf{x}_{i_t}^\top \mathbf{w}_t$
  - $\mathbf{w}_{t+1} := \mathbf{w}'_{t+1} / \left\|\mathbf{w}'_{t+1}\right\|$

Also Krasulina 1969; Arora, Cotter, Livescu, Srebro 2012; Mitliagkas, Caramanis, Jain 2013; Balsubramani, Dasgupta, Freund 2013; Hardt, Price 2014; De Sa, Olukotun, Ré 2015...

- $\mathcal{O}(d)$ runtime per iteration.
- Iteration bound: $\tilde{\mathcal{O}}\left(\frac{d}{\lambda^2 \epsilon}\right)$
- Runtime: $\tilde{\mathcal{O}}\left(\frac{\mathbf{d}^2}{\lambda^2 \epsilon}\right)$

Existing Approaches

Approach 3: Stochastic/Incremental Algorithms

Example (Oja's algorithm)

- Initialize $\mathbf{w}_1$ randomly
- For $t = 1, 2, \ldots$
  - Pick $i_t \in \{1, \ldots, n\}$ (randomly or otherwise)
  - $\mathbf{w}'_{t+1} := \mathbf{w}_t + \eta_t \mathbf{x}_{i_t} \mathbf{x}_{i_t}^\top \mathbf{w}_t$
  - $\mathbf{w}_{t+1} := \mathbf{w}'_{t+1} / \left\|\mathbf{w}'_{t+1}\right\|$

Also Krasulina 1969; Arora, Cotter, Livescu, Srebro 2012; Mitliagkas, Caramanis, Jain 2013; Balsubramani, Dasgupta, Freund 2013; Hardt, Price 2014; De Sa, Olukotun, Ré 2015...

- $\mathcal{O}(d)$ runtime per iteration.
- Iteration bound: $\tilde{\mathcal{O}}\left(\frac{d}{\lambda^2 \epsilon}\right)$
- Runtime: $\tilde{\mathcal{O}}\left(\frac{\mathbf{d}^2}{\lambda^2 \epsilon}\right)$

Other stochastic approaches (e.g. [Halko, Martinsson, Tropp, 2011]): Inherent polynomial dependence on ϵ

Existing Approaches

Up to constants/log-factors:

Algorithm	Time per iter.	# iter.	Runtime
Exact			$d^3 + nd^2$
Power/Lanczos	nd	$\frac{1}{\lambda^p}$	$\frac{nd}{\lambda^p}$
Stochastic	d	$\frac{d}{\lambda^2 \epsilon}$	$\frac{d^2}{\lambda^2 \epsilon}$

Main Question

Can we get the best of both worlds? $\mathcal{O}(d)$ time per iteration **and** high accuracy (logarithmic dependence on ϵ?)

Convex Optimization to the Rescue?

Problem is equivalent to:

$$\min_{\mathbf{w}:\|\mathbf{w}\|=1} \frac{1}{n}\sum_{i=1}^{n}\left(-\langle \mathbf{w}, \mathbf{x}_i\rangle^2\right)$$

Recent progress in high-accuracy stochastic algorithms for strongly-convex+smooth, finite-sum problems

$$\min_{\mathbf{w}\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w})$$

[Le Roux, Schmidt, Bach 2012; Shalev-Shwartz and Zhang 2012; Johnson and Zhang 2013; Zhang, Mahdavi, Jin 2013; Konečný and Richtárik 2013; Xiao and Zhang 2014; Zhang and Xiao, 2014...]

Convex Optimization to the Rescue?

$$\min_{\mathbf{w}:\|\mathbf{w}\|=1} \frac{1}{n}\sum_{i=1}^{n}\left(-\langle \mathbf{w}, \mathbf{x}_i\rangle^2\right)$$

Unfortunately:

- Not strongly convex, nor convex (in fact, concave everywhere)
- Has > 1 global optima, plateaus...

$\Rightarrow$ Existing results don't work as-is

But: Maybe we can borrow some ideas...

Algorithm

$$\min_{\mathbf{w}:\|\mathbf{w}\|=1} \frac{1}{n}\sum_{i=1}^{n}\left(-\langle \mathbf{w}, \mathbf{x}_i\rangle^2\right)$$

Oja Iteration

- Choose $i_t \in \{1,\ldots,n\}$ at random
- $\mathbf{w}'_{t+1} = \mathbf{w}_t + \eta_t \langle \mathbf{w}_t, \mathbf{x}_{i_t}\rangle \mathbf{x}_{i_t}$
- $\mathbf{w}_{t+1} := \mathbf{w}'_{t+1} / \left\|\mathbf{w}'_{t+1}\right\|$

Essentially stochastic gradient descent

Algorithm

Letting $A = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top$, update step is

$$\begin{aligned}
\mathbf{w}'_{t+1} &= \mathbf{w}_t + \eta_t \mathbf{x}_{i_t} \mathbf{x}_{i_t}^\top \mathbf{w}_t \\
&= \mathbf{w}_t + \underbrace{\eta_t A \mathbf{w}_t}_{\text{power/gradient step}} + \underbrace{\eta_t \left(\mathbf{x}_{i_t} \mathbf{x}_{i_t}^\top - A \right) \mathbf{w}_t}_{\text{zero-mean noise}}
\end{aligned}$$

Algorithm

Letting $A = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top$, update step is

$$\begin{aligned}
\mathbf{w}'_{t+1} &= \mathbf{w}_t + \eta_t \mathbf{x}_{i_t} \mathbf{x}_{i_t}^\top \mathbf{w}_t \\
&= \mathbf{w}_t + \underbrace{\eta_t A \mathbf{w}_t}_{\text{power/gradient step}} + \underbrace{\eta_t \left(\mathbf{x}_{i_t} \mathbf{x}_{i_t}^\top - A \right) \mathbf{w}_t}_{\text{zero-mean noise}}
\end{aligned}$$

Main idea: Replace by

$$\mathbf{w}'_{t+1} = \mathbf{w}_t + \underbrace{\eta A \mathbf{w}_t}_{\text{power/gradient step}} + \underbrace{\eta \left(\mathbf{x}_{i_t} \mathbf{x}_{i_t}^\top - A \right) (\mathbf{w}_t - \tilde{\mathbf{w}})}_{\text{zero-mean noise}}$$

where $\tilde{\mathbf{w}}$ "close" to $\mathbf{w}_t$
(similar to SVRG of Johnson and Zhang (2013))

Algorithm

VR-PCA

- **Parameters:** Step size η, epoch length m
- **Input:** Data set $\{\mathbf{x}_i\}_{i=1}^n$, Initial unit vector $\tilde{\mathbf{w}}_0$
- For $s = 1, 2, \ldots$
  - $\tilde{\mathbf{u}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\top \tilde{\mathbf{w}}_{s-1}$
  - $\mathbf{w}_0 = \tilde{\mathbf{w}}_{s-1}$
  - For $t = 1, 2, \ldots, m$
    - Pick $i_t \in \{1, \ldots, n\}$ uniformly at random
    - $\mathbf{w}'_t = \mathbf{w}_{t-1} + \eta\left(\mathbf{x}_{i_t}\mathbf{x}_{i_t}^\top(\mathbf{w}_{t-1} - \tilde{\mathbf{w}}_{s-1}) + \tilde{\mathbf{u}}\right)$
    - $\mathbf{w}_t = \frac{1}{\|\mathbf{w}'_t\|}\mathbf{w}'_t$
  - $\tilde{\mathbf{w}}_s = \mathbf{w}_m$

Algorithm

VR-PCA

- **Parameters:** Step size η, epoch length m
- **Input:** Data set $\{\mathbf{x}_i\}_{i=1}^n$, Initial unit vector $\tilde{\mathbf{w}}_0$
- For $s = 1, 2, \ldots$
  - $\tilde{\mathbf{u}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\top \tilde{\mathbf{w}}_{s-1}$
  - $\mathbf{w}_0 = \tilde{\mathbf{w}}_{s-1}$
  - For $t = 1, 2, \ldots, m$
    - Pick $i_t \in \{1, \ldots, n\}$ uniformly at random
    - $\mathbf{w}'_t = \mathbf{w}_{t-1} + \eta\left(\mathbf{x}_{i_t}\mathbf{x}_{i_t}^\top(\mathbf{w}_{t-1} - \tilde{\mathbf{w}}_{s-1}) + \tilde{\mathbf{u}}\right)$
    - $\mathbf{w}_t = \frac{1}{\|\mathbf{w}'_t\|}\mathbf{w}'_t$
  - $\tilde{\mathbf{w}}_s = \mathbf{w}_m$

To get $k > 1$ directions: Either repeat, or perform orthogonal-like iterations:

- Replace all vectors by $k \times d$ matrices
- Replace normalization step by orthogonalization step

Analysis

Theorem (paraphrase)

Suppose $\max_i \|\mathbf{x}_i\|_2 \leq 1$*; Eigengap* λ *between first two eigenvalues;* $\mathbf{v}_1$ *is a leading eigenvector and* $\langle \tilde{\mathbf{w}}_0, \mathbf{v}_1 \rangle \geq \frac{1}{\sqrt{2}}$

Picking parameters appropriately, algorithm returns unit $\mathbf{w}$ *s.t.* $\langle \mathbf{w}, \mathbf{v}_1 \rangle^2 \geq 1 - \epsilon$*, in runtime* $\mathcal{O}\left(d\left(n + \frac{1}{\lambda^2}\right)\log\left(\frac{1}{\epsilon}\right)\right)$

Analysis

Theorem (paraphrase)

Suppose $\max_i \|\mathbf{x}_i\|_2 \leq 1$*; Eigengap* λ *between first two eigenvalues;* $\mathbf{v}_1$ *is a leading eigenvector and* $\langle \tilde{\mathbf{w}}_0, \mathbf{v}_1 \rangle \geq \frac{1}{\sqrt{2}}$

Picking parameters appropriately, algorithm returns unit $\mathbf{w}$ *s.t.* $\langle \mathbf{w}, \mathbf{v}_1 \rangle^2 \geq 1 - \epsilon$*, in runtime* $\mathcal{O}\left(d\left(n + \frac{1}{\lambda^2}\right)\log\left(\frac{1}{\epsilon}\right)\right)$

- Exponential convergence
- Runtime depends on # examples plus eigengap; Proportional to single data pass if $\lambda \geq 1/\sqrt{n}$
- d can be replaced by average sparsity
- To get appropriate $\tilde{\mathbf{w}}_0$: Can use existing stochastic algorithms/analysis (in practice, not an issue)

Proof Idea

Track decay of $F(\mathbf{w}_t) = 1 - \langle \mathbf{w}_t, \mathbf{v}_1 \rangle^2$

Key Lemma

Assuming $F(\mathbf{w}_t) \leq 3/4$, for constants $c, c' > 0$

$$\mathbb{E}\left[F(\mathbf{w}_{t+1}) | \mathbf{w}_t\right] \;\leq\; (1 - c\lambda\eta)\, F(\mathbf{w}_t) + c'\eta^2 F(\tilde{\mathbf{w}}_{s-1}).$$

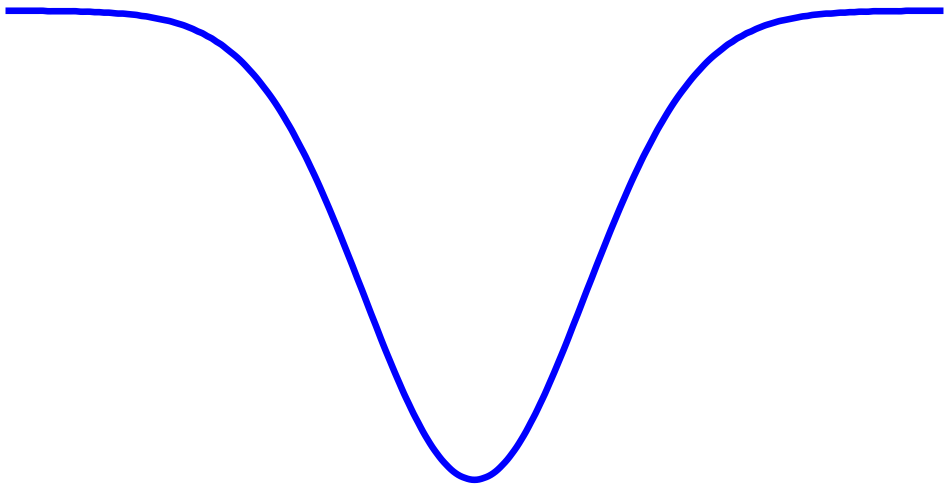
Proof Idea

Track decay of $F(\mathbf{w}_t) = 1 - \langle \mathbf{w}_t, \mathbf{v}_1 \rangle^2$

Key Lemma

Assuming $F(\mathbf{w}_t) \leq 3/4$, for constants $c, c' > 0$

$$\mathbb{E}\left[F(\mathbf{w}_{t+1}) | \mathbf{w}_t\right] \leq (1 - c\lambda\eta) F(\mathbf{w}_t) + c'\eta^2 F(\tilde{\mathbf{w}}_{s-1}).$$



Proof Idea

Assume $\eta = \alpha\lambda$ ($\alpha \ll 1$)

Proof Idea

Assume $\eta = \alpha\lambda$ ($\alpha \ll 1$)

- Using martingale arguments: W.h.p., never reach "flat" region in $m \leq \mathcal{O}\left(\frac{1}{\alpha^2\lambda^2}\right)$ iterations

Proof Idea

Assume $\eta = \alpha\lambda$ ($\alpha \ll 1$)

- Using martingale arguments: W.h.p., never reach "flat" region in $m \leq \mathcal{O}\left(\frac{1}{\alpha^2\lambda^2}\right)$ iterations
- $\Rightarrow$ For all $t \leq m$
  $\mathbb{E}\left[F(\mathbf{w}_{t+1})|\mathbf{w}_t\right] \leq \left(1 - \Theta(\alpha\lambda^2)\right) F(\mathbf{w}_t) + \mathcal{O}\left(\alpha^2\lambda^2 F(\tilde{\mathbf{w}}_{s-1})\right).$

Proof Idea

Assume $\eta = \alpha\lambda$ ($\alpha \ll 1$)

- Using martingale arguments: W.h.p., never reach "flat" region in $m \leq \mathcal{O}\left(\frac{1}{\alpha^2\lambda^2}\right)$ iterations
- $\Rightarrow$ For all $t \leq m$
  $\mathbb{E}\left[F(\mathbf{w}_{t+1})|\mathbf{w}_t\right] \leq \left(1 - \Theta(\alpha\lambda^2)\right) F(\mathbf{w}_t) + \mathcal{O}\left(\alpha^2\lambda^2 F(\tilde{\mathbf{w}}_{s-1})\right).$
- Carefully unwinding recursion, and using $\mathbf{w}_0 = \tilde{\mathbf{w}}_{s-1}$,
  $\mathbb{E}\left[F(\mathbf{w}_m)|\mathbf{w}_0\right] \leq \left(\left(1 - \Theta(\alpha\lambda^2)\right)^m + \mathcal{O}(\alpha)\right) F(\tilde{\mathbf{w}}_{s-1})$

Proof Idea

Assume $\eta = \alpha\lambda$ ($\alpha \ll 1$)

- Using martingale arguments: W.h.p., never reach "flat" region in $m \leq \mathcal{O}\left(\frac{1}{\alpha^2\lambda^2}\right)$ iterations
- $\Rightarrow$ For all $t \leq m$
  $\mathbb{E}\left[F(\mathbf{w}_{t+1})|\mathbf{w}_t\right] \leq \left(1 - \Theta(\alpha\lambda^2)\right) F(\mathbf{w}_t) + \mathcal{O}\left(\alpha^2\lambda^2 F(\tilde{\mathbf{w}}_{s-1})\right).$
- Carefully unwinding recursion, and using $\mathbf{w}_0 = \tilde{\mathbf{w}}_{s-1}$,
  $\mathbb{E}\left[F(\mathbf{w}_m)|\mathbf{w}_0\right] \leq \left(\left(1 - \Theta(\alpha\lambda^2)\right)^m + \mathcal{O}(\alpha)\right) F(\tilde{\mathbf{w}}_{s-1})$
- $\Rightarrow$ If $m \geq \Omega\left(\frac{1}{\alpha\lambda^2}\right)$, $F(\mathbf{w}_m)$ smaller than $F(\tilde{\mathbf{w}}_{s-1})$ by some constant factor

Proof Idea

Assume $\eta = \alpha\lambda$ ($\alpha \ll 1$)

- Using martingale arguments: W.h.p., never reach "flat" region in $m \leq \mathcal{O}\left(\frac{1}{\alpha^2\lambda^2}\right)$ iterations
- $\Rightarrow$ For all $t \leq m$
  $\mathbb{E}\left[F(\mathbf{w}_{t+1})|\mathbf{w}_t\right] \leq \left(1 - \Theta(\alpha\lambda^2)\right) F(\mathbf{w}_t) + \mathcal{O}\left(\alpha^2\lambda^2 F(\tilde{\mathbf{w}}_{s-1})\right).$
- Carefully unwinding recursion, and using $\mathbf{w}_0 = \tilde{\mathbf{w}}_{s-1}$,
  $\mathbb{E}\left[F(\mathbf{w}_m)|\mathbf{w}_0\right] \leq \left(\left(1 - \Theta(\alpha\lambda^2)\right)^m + \mathcal{O}(\alpha)\right) F(\tilde{\mathbf{w}}_{s-1})$
- $\Rightarrow$ If $m \geq \Omega\left(\frac{1}{\alpha\lambda^2}\right)$, $F(\mathbf{w}_m)$ smaller than $F(\tilde{\mathbf{w}}_{s-1})$ by some constant factor
- Overall, if $\frac{1}{\alpha\lambda^2} \ll m \ll \frac{1}{\alpha^2\lambda^2}$, every epoch shrinks F by constant factor

Experiments

Ran VR-PCA with

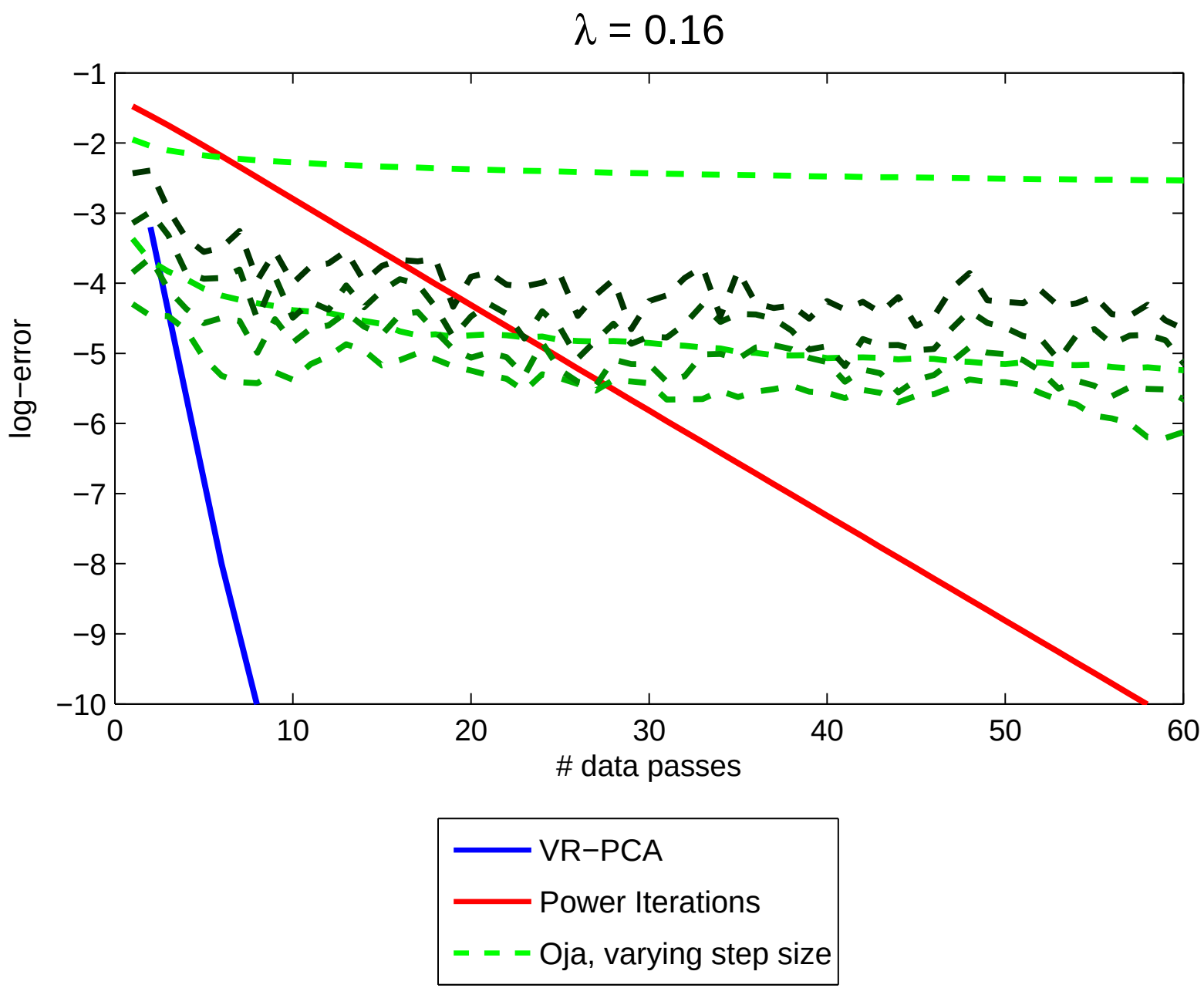
- Random initialization
- $m = n$ (epoch = two passes over data)
- $\eta = \frac{1}{\bar{r}\sqrt{n}}$ where $\bar{r} = \frac{1}{n}\sum_i \|\mathbf{x}_i\|^2$ (based on analysis)

Compared to

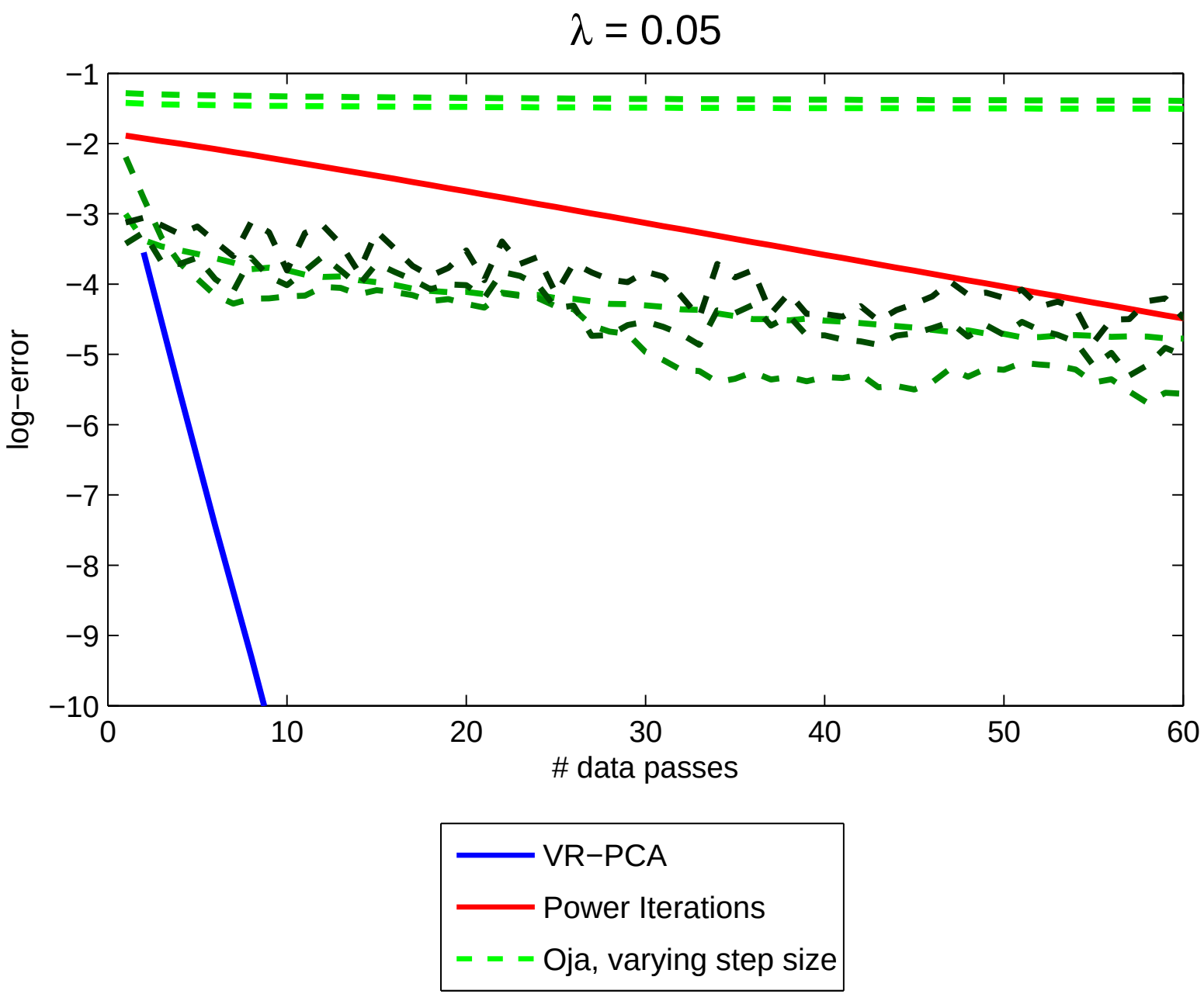
- Oja's algorithm with hand-tuned step-size
- Power iterations

Same starting point

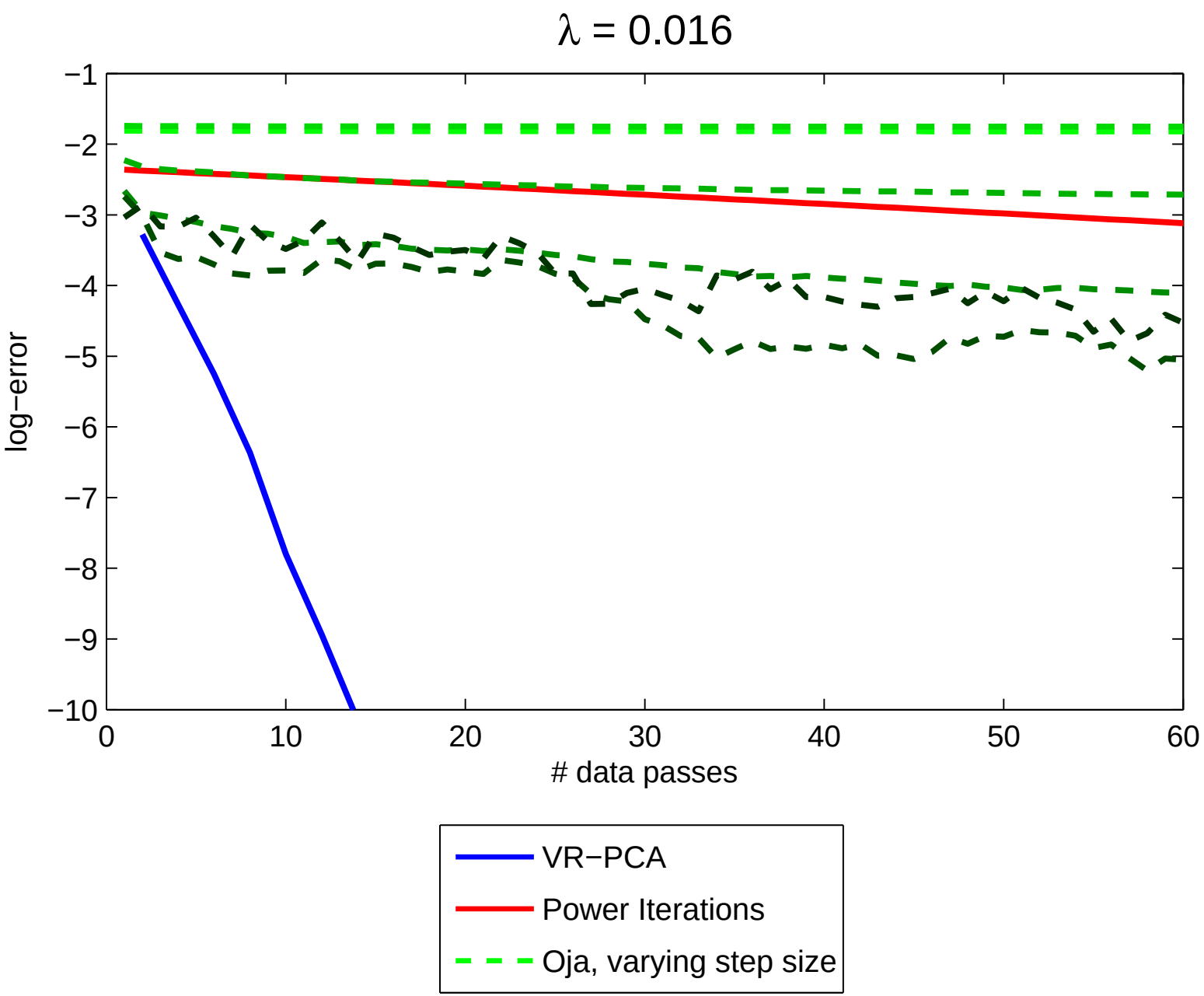
Experiments



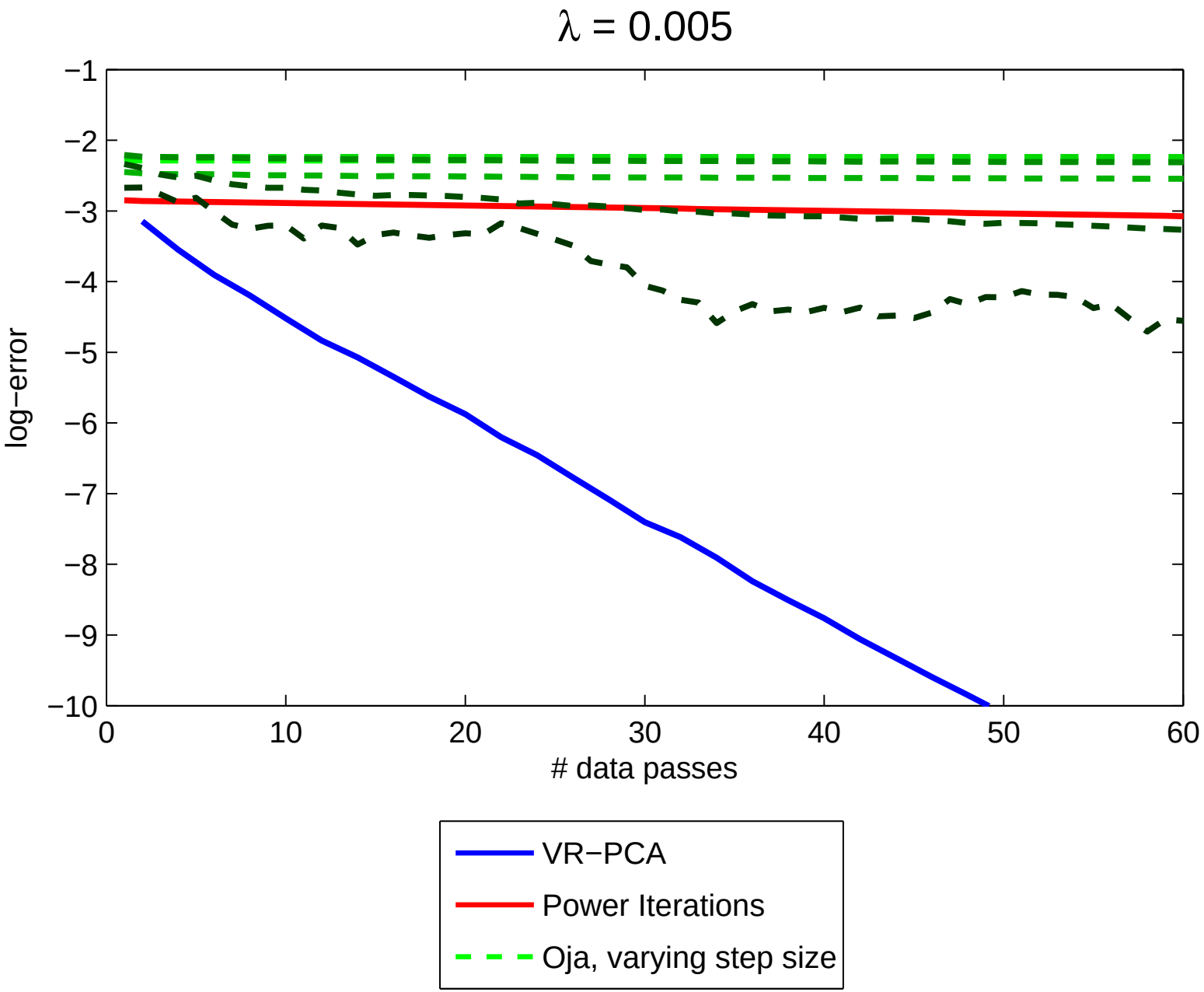
Experiments



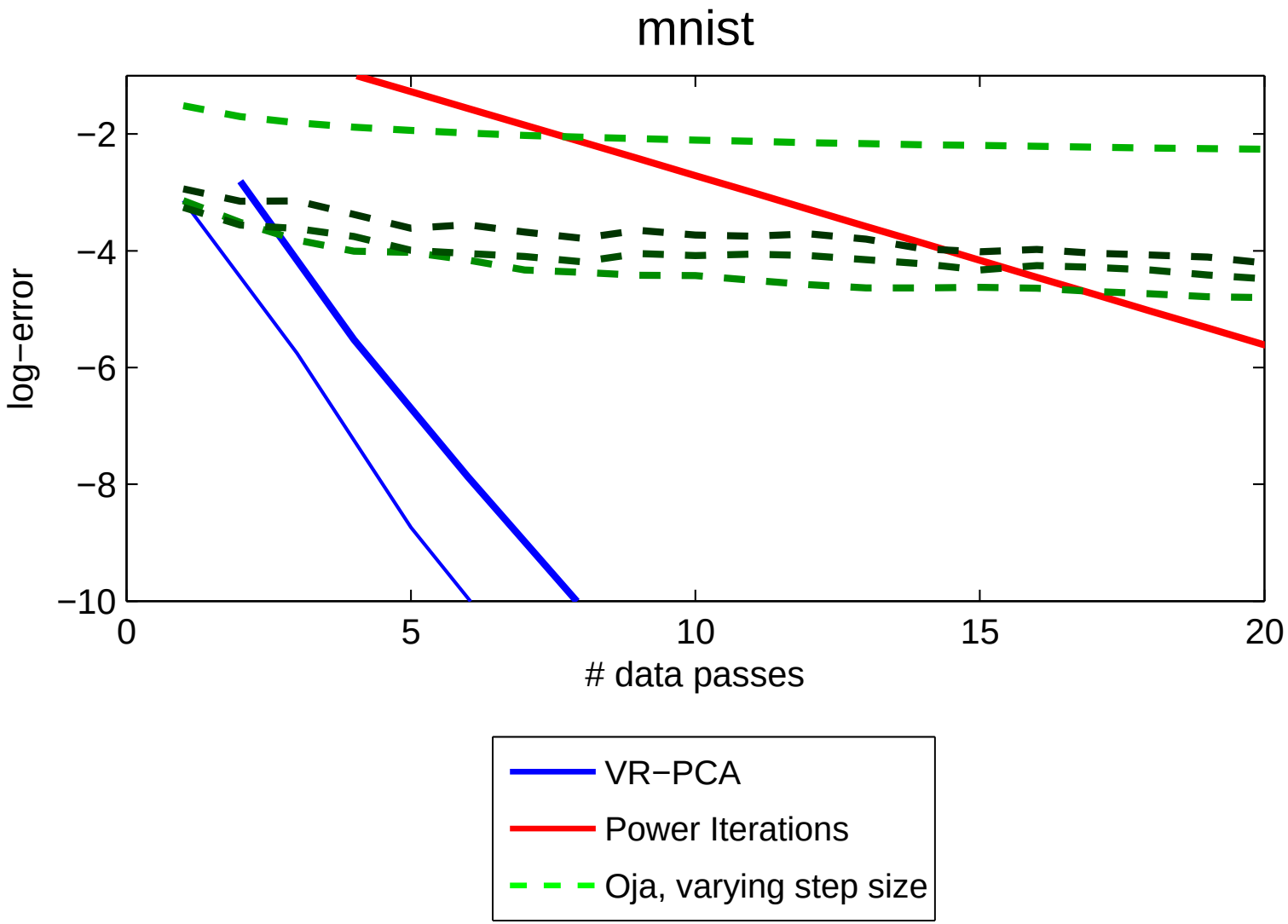
Experiments



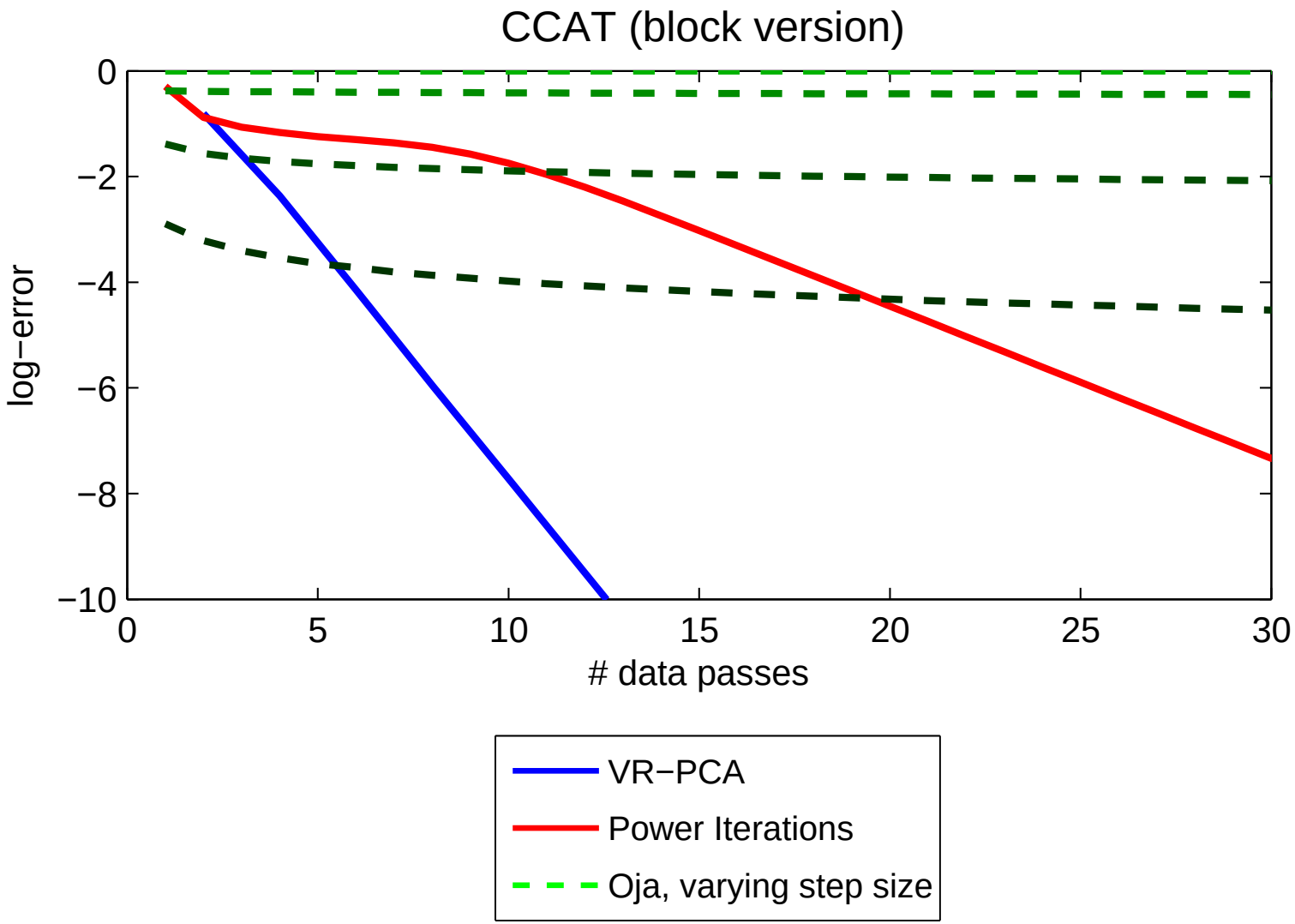
Experiments



Experiments



Experiments



Work-in-progress and Open Questions

Work-in-progress and Open Questions

- Generalizing analysis to block version and random initialization
- Runtime is $\mathcal{O}(d(n+\frac{1}{\lambda^2})\log\left(\frac{1}{\epsilon}\right))$ can we get $\mathcal{O}(d(n+\frac{1}{\lambda})\log\left(\frac{1}{\epsilon}\right)$ or better, analogous to convex case?
- Analyze step-size independent of λ?
- Using other "fast-stochastic" approaches?
- Other non-convex problems?

Work-in-progress and Open Questions

- Generalizing analysis to block version and random initialization
- Runtime is $\mathcal{O}(d(n+\frac{1}{\lambda^2})\log\left(\frac{1}{\epsilon}\right))$ can we get $\mathcal{O}(d(n+\frac{1}{\lambda})\log\left(\frac{1}{\epsilon}\right)$ or better, analogous to convex case?
- Analyze step-size independent of λ?
- Using other "fast-stochastic" approaches?
- Other non-convex problems?

Thanks!