The Noisy Power Method

Moritz Hardt **Eric Price**

IBM IBM → UT Austin

2014-10-31

Problem

- Common problem: find *low rank* approximation to a matrix A
  - PCA: apply to covariance matrix
  - Spectral analysis: PageRank, Cheever's inequality for cuts, etc.

Simple algorithm: the power method

AKA subspace iteration, subspace power iteration

- Choose random $X_0 \in \mathbb{R}^{n \times k}$.

Simple algorithm: the power method

AKA subspace iteration, subspace power iteration

- Choose random $X_0 \in \mathbb{R}^{n \times k}$.
- Repeat:

$$Y_{t+1} = AX_t$$
$$X_{t+1} = \text{orthonormalize}(Y_{t+1})$$

Simple algorithm: the power method

AKA subspace iteration, subspace power iteration

- Choose random $X_0 \in \mathbb{R}^{n \times k}$.
- Repeat:

$$\begin{aligned} Y_{t+1} &= AX_t \\ X_{t+1} &= \text{orthonormalize}(Y_{t+1}) \end{aligned}$$

- Converges towards U, the space of the top k eigenvalues.

Simple algorithm: the power method

AKA subspace iteration, subspace power iteration

- Choose random $X_0 \in \mathbb{R}^{n \times k}$.
- Repeat:

$$
\begin{aligned}
Y_{t+1} &= AX_t \\
X_{t+1} &= \text{orthonormalize}(Y_{t+1})
\end{aligned}
$$

- Converges towards U, the space of the top k eigenvalues.
- Question 1: how quickly?

Simple algorithm: the power method

AKA subspace iteration, subspace power iteration

- Choose random $X_0 \in \mathbb{R}^{n \times k}$.
- Repeat:

$$\begin{aligned} Y_{t+1} &= AX_t \\ X_{t+1} &= \text{orthonormalize}(Y_{t+1}) \end{aligned}$$

- Converges towards U, the space of the top k eigenvalues.
- Question 1: how quickly?
  - [Stewart '69, ..., Halko-Martinsson-Tropp '10]

Simple algorithm: the power method

AKA subspace iteration, subspace power iteration

- Choose random $X_0 \in \mathbb{R}^{n \times k}$.
- Repeat:

$$
\begin{aligned}
Y_{t+1} &= AX_t + G \\
X_{t+1} &= \text{orthonormalize}(Y_{t+1})
\end{aligned}
$$

- Converges towards U, the space of the top k eigenvalues.
- Question 1: how quickly?
  - [Stewart '69, ..., Halko-Martinsson-Tropp '10]
- Question 2: how robust to noise?

Simple algorithm: the power method

AKA subspace iteration, subspace power iteration

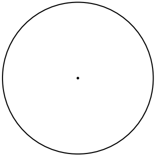
- Choose random $X_0 \in \mathbb{R}^{n \times k}$.
- Repeat:

$$Y_{t+1} = AX_t + G$$
$$X_{t+1} = \text{orthonormalize}(Y_{t+1})$$

- Converges towards U, the space of the top k eigenvalues.
- Question 1: how quickly?
  - [Stewart '69, ..., Halko-Martinsson-Tropp '10]
- Question 2: how robust to noise?
  - Application-specific bounds: [Hardt-Roth '13, Mitliagkas-Caramanis-Jain '13, Jain-Netrapalli-Sanghavi '13]

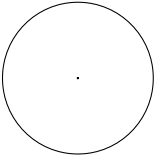
Basic power method, $k = 1$

- Choose a random unit vector $x \in \mathbb{R}^n$.



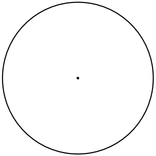
Basic power method, $k = 1$

- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.

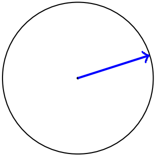


Basic power method, $k = 1$

- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.



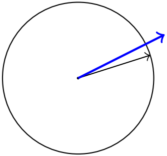
Basic power method, $k = 1$



- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
    - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.

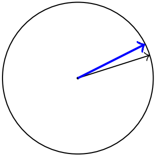
Basic power method, $k = 1$

- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q - 1$.



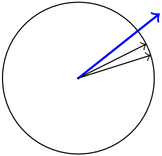
Basic power method, $k = 1$

- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.

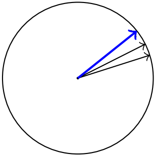


Basic power method, $k = 1$

- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.

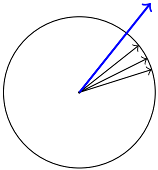


Basic power method, $k = 1$



- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q - 1$.

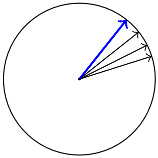
Basic power method, $k = 1$



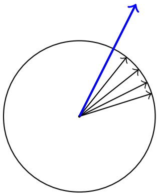
- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.

Basic power method, $k = 1$

- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.



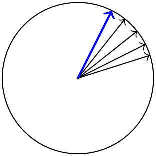
Basic power method, $k = 1$



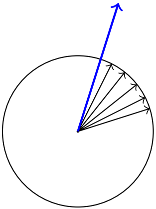
- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.

Basic power method, $k = 1$

- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.

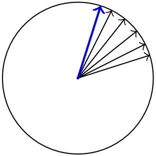


Basic power method, $k = 1$



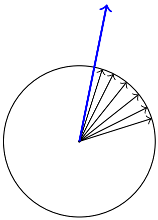
- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.

Basic power method, $k = 1$



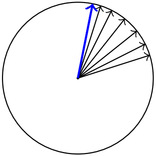
- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q - 1$.

Basic power method, $k = 1$



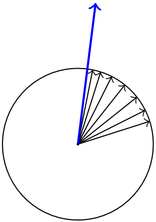
- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.

Basic power method, $k = 1$



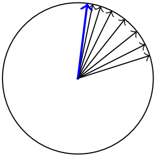
- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.

Basic power method, $k = 1$



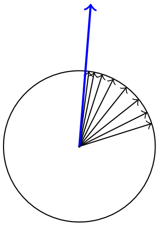
- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.

Basic power method, $k = 1$



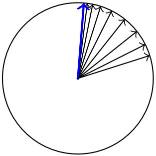
- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.

Basic power method, $k = 1$



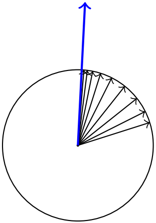
- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.

Basic power method, $k = 1$



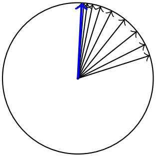
- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q - 1$.

Basic power method, $k = 1$



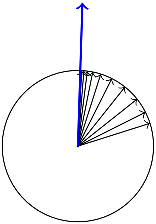
- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.

Basic power method, $k = 1$



- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.

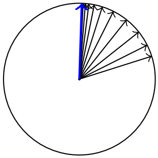
Basic power method, $k = 1$



- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.

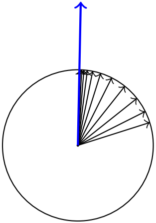
Basic power method, $k = 1$

- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.



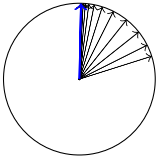
Basic power method, $k = 1$

- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q - 1$.

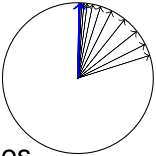


Basic power method, $k = 1$

- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.

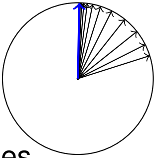


Basic power method, $k = 1$



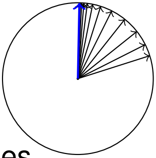
- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.
- Suppose A has eigenvectors $v_1, \ldots, v_n$, eigenvalues $\lambda_1 > \lambda_2 \geq \cdots \lambda_n \geq 0$.

Basic power method, $k = 1$



- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.
- Suppose A has eigenvectors $v_1, \ldots, v_n$, eigenvalues $\lambda_1 > \lambda_2 \geq \cdots \lambda_n \geq 0$.
- Start with $x_0 = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n$.

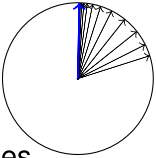
Basic power method, $k = 1$



- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.
- Suppose A has eigenvectors $v_1, \ldots, v_n$, eigenvalues $\lambda_1 > \lambda_2 \geq \cdots \lambda_n \geq 0$.
- Start with $x_0 = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n$.
- After q iterations,

$$A^q x_0 = \sum_i \lambda_i^q \alpha_i v_i$$

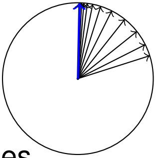
Basic power method, $k = 1$



- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.
- Suppose A has eigenvectors $v_1, \ldots, v_n$, eigenvalues $\lambda_1 > \lambda_2 \geq \cdots \lambda_n \geq 0$.
- Start with $x_0 = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n$.
- After q iterations,

$$A^q x_0 = \sum_i \lambda_i^q \alpha_i v_i \propto v_1 + \sum_{i \geq 2} \left(\frac{\lambda_i}{\lambda_1} \right)^q \frac{\alpha_i}{\alpha_1} v_i$$

Basic power method, $k = 1$

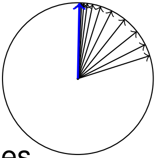


- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.
- Suppose A has eigenvectors $v_1, \ldots, v_n$, eigenvalues $\lambda_1 > \lambda_2 \geq \cdots \lambda_n \geq 0$.
- Start with $x_0 = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n$.
- After q iterations,

$$A^q x_0 = \sum_i \lambda_i^q \alpha_i v_i \propto v_1 + \sum_{i \geq 2} \left(\frac{\lambda_i}{\lambda_1} \right)^q \frac{\alpha_i}{\alpha_1} v_i$$

- For $q \geq \log_{\lambda_1/\lambda_2} \frac{d}{\epsilon \alpha_1}$, have $A^q x$ proportional to $v_1 \pm O(\epsilon)$.

Basic power method, $k = 1$



- Choose a random unit vector $x \in \mathbb{R}^n$.
- Output $A^q x$, renormalized to unit vector.
  - $x_{t+1} = Ax_t / \|Ax_t\|$ for $t = 0, \ldots, q-1$.
- Suppose A has eigenvectors $v_1, \ldots, v_n$, eigenvalues $\lambda_1 > \lambda_2 \geq \cdots \lambda_n \geq 0$.
- Start with $x_0 = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n$.
- After q iterations,

$$A^q x_0 = \sum_i \lambda_i^q \alpha_i v_i \propto v_1 + \sum_{i \geq 2} \left(\frac{\lambda_i}{\lambda_1}\right)^q \frac{\alpha_i}{\alpha_1} v_i$$

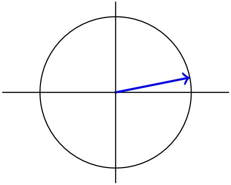
- For $q \geq \log_{\lambda_1/\lambda_2} \frac{d}{\epsilon \alpha_1}$, have $A^q x$ proportional to $v_1 \pm O(\epsilon)$.

$$q = O(\frac{\lambda_1}{\lambda_1 - \lambda_2} \log n)$$

Handling noise, $k = 1$

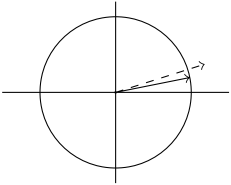
- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$



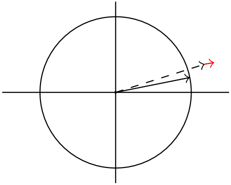
Handling noise, $k = 1$

- Consider the iteration



$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

Handling noise, $k = 1$



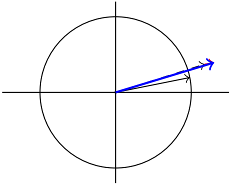
- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

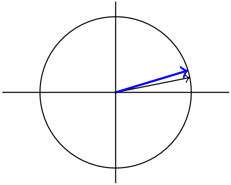
Handling noise, $k = 1$

- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$



Handling noise, $k = 1$

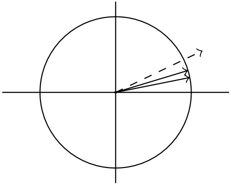


- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

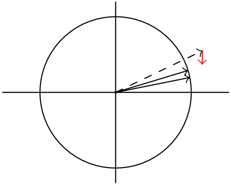
Handling noise, $k = 1$

- Consider the iteration



$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

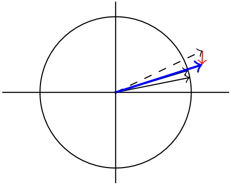
Handling noise, $k = 1$



- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

Handling noise, $k = 1$

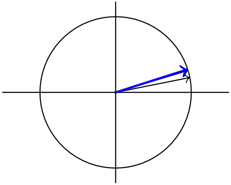


- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

Handling noise, $k = 1$

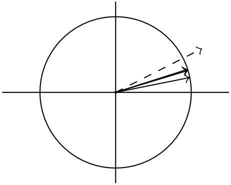
- Consider the iteration



$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

Handling noise, $k = 1$

- Consider the iteration

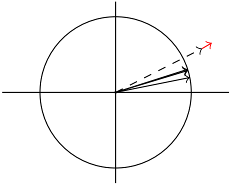


$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

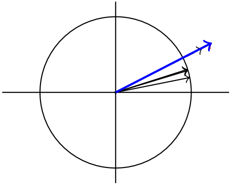
Handling noise, $k = 1$

- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1} / \|y_{t+1}\|$$



Handling noise, $k = 1$



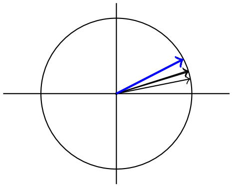
- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

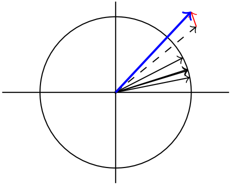
Handling noise, $k = 1$

- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$



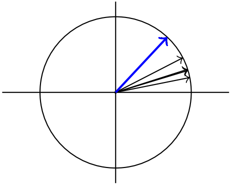
Handling noise, $k = 1$



- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

Handling noise, $k = 1$



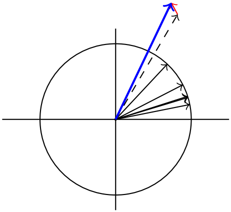
- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

Handling noise, $k = 1$

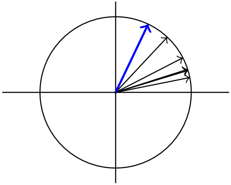
- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1} / \|y_{t+1}\|$$



Handling noise, $k = 1$

- Consider the iteration

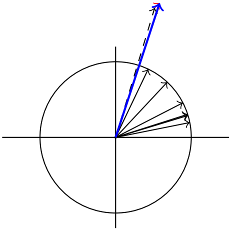


$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

Handling noise, $k = 1$

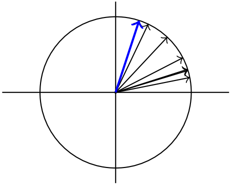
- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$



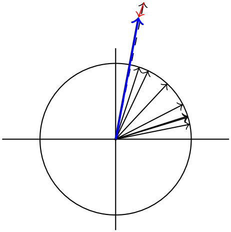
Handling noise, $k = 1$

- Consider the iteration



$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

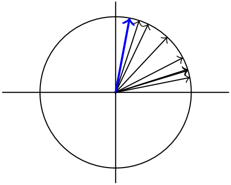
Handling noise, $k = 1$



- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

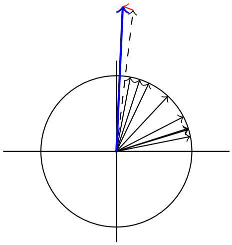
Handling noise, $k = 1$



- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

Handling noise, $k = 1$



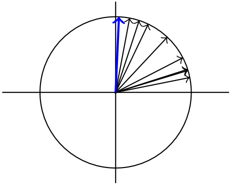
- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

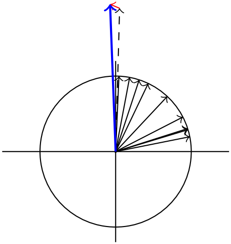
Handling noise, $k = 1$

- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1} / \|y_{t+1}\|$$



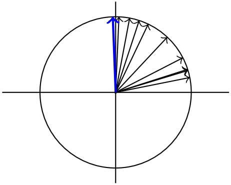
Handling noise, $k = 1$



- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

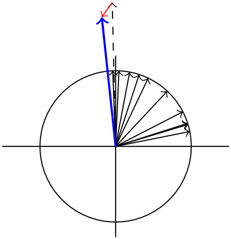
Handling noise, $k = 1$



- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

Handling noise, $k = 1$

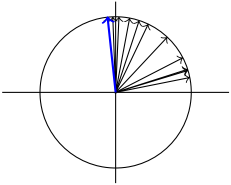


- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

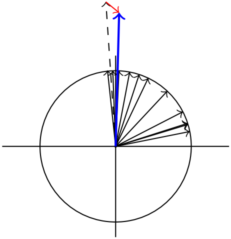
Handling noise, $k = 1$

- Consider the iteration



$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

Handling noise, $k = 1$

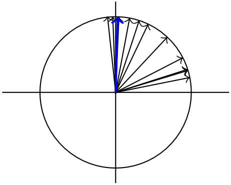


- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

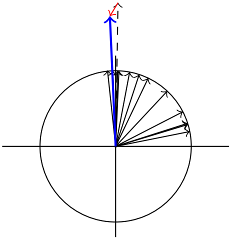
Handling noise, $k = 1$

- Consider the iteration



$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

Handling noise, $k = 1$



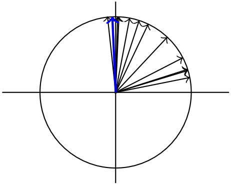
- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

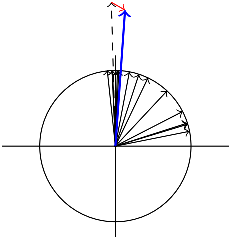
Handling noise, $k = 1$

- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$



Handling noise, $k = 1$



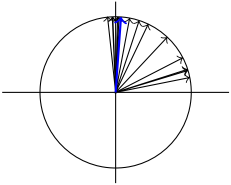
- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1} / \|y_{t+1}\|$$

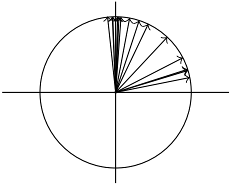
Handling noise, $k = 1$

- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$



Handling noise, $k = 1$

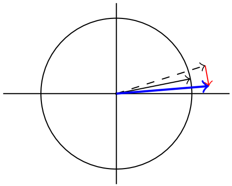


- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

- What conditions on G will cause this to converge to within ϵ?

Handling noise, $k = 1$



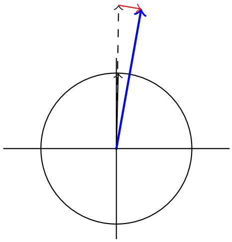
- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

- What conditions on G will cause this to converge to within ϵ?
  - G must make progress at the beginning

$$|G_1| \le (\lambda_1 - \lambda_2)\frac{1}{\sqrt{d}}$$

Handling noise, $k = 1$



- Consider the iteration

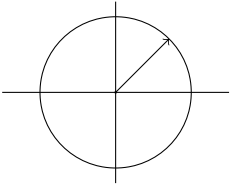
$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

- What conditions on G will cause this to converge to within ϵ?
  - G must make progress at the beginning
  - G must not perturb by ϵ at the end.

$$|G_1| \leq (\lambda_1 - \lambda_2)\frac{1}{\sqrt{d}} \qquad \|G\| \leq \epsilon(\lambda_1 - \lambda_2)$$

Handling noise, $k = 1$

- Consider the iteration

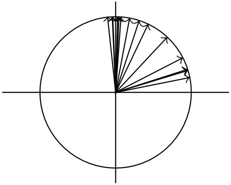


$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

- What conditions on G will cause this to converge to within ϵ?
  - G must make progress at the beginning
  - G must not perturb by ϵ at the end.
  - Looser requirements in the middle.

$$|G_1| \leq (\lambda_1 - \lambda_2)\frac{1}{\sqrt{d}} \qquad \|G\| \leq \epsilon(\lambda_1 - \lambda_2)$$

Handling noise, $k = 1$



- Consider the iteration

$$y_{t+1} = Ax_t + G$$
$$x_{t+1} = y_{t+1}/\|y_{t+1}\|$$

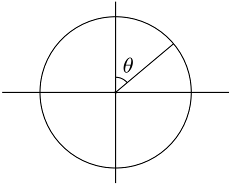
- What conditions on G will cause this to converge to within ϵ?
  - G must make progress at the beginning
  - G must not perturb by ϵ at the end.
  - Looser requirements in the middle.
- **Theorem:** Converges to $v_1 \pm O(\epsilon)$ if all the G satisfy

$$|G_1| \leq (\lambda_1 - \lambda_2)\frac{1}{\sqrt{d}} \qquad \|G\| \leq \epsilon(\lambda_1 - \lambda_2)$$

in $O(\frac{\lambda_1}{\lambda_2 - \lambda_1} \log(d/\epsilon))$ iterations.

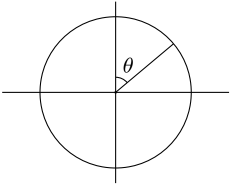
Noisy convergence proof ($k = 1$)

- Use a potential-based argument to show progress at each step. Potential:



$$\tan\theta = \frac{\sqrt{\sum_{j>1}\alpha_j^2}}{\alpha_1}$$

Noisy convergence proof ($k = 1$)



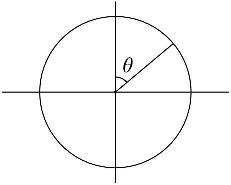
- Use a potential-based argument to show progress at each step. Potential:

$$\tan\theta = \frac{\sqrt{\sum_{j>1} \alpha_j^2}}{\alpha_1}$$

- With no noise:

$$\tan\theta_{t+1} = \frac{\sqrt{\sum_{j>1} \lambda_j^2 \alpha_j^2}}{\lambda_1 \alpha_1} \leq \frac{\lambda_2}{\lambda_1} \tan\theta_t$$

Noisy convergence proof ($k = 1$)



- Use a potential-based argument to show progress at each step. Potential:

$$\tan\theta = \frac{\sqrt{\sum_{j>1}\alpha_j^2}}{\alpha_1}$$

- With no noise:

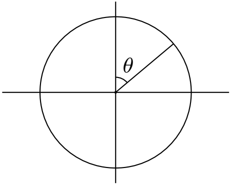
$$\tan\theta_{t+1} = \frac{\sqrt{\sum_{j>1}\lambda_j^2\alpha_j^2}}{\lambda_1\alpha_1} \leq \frac{\lambda_2}{\lambda_1}\tan\theta_t$$

- With noise G satisfying the conditions ($|G_1|$, $\|G\|$ small enough),

$$\tan\theta_{t+1} \leq \frac{\lambda_2\sqrt{\sum_{j>1}\alpha_j^2} + \|G\|}{\lambda_1\alpha_1 - |G_1|}$$

Noisy convergence proof ($k = 1$)

- Use a potential-based argument to show progress at each step. Potential:



$$\tan\theta = \frac{\sqrt{\sum_{j>1}\alpha_j^2}}{\alpha_1}$$

- With no noise:

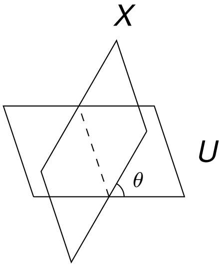
$$\tan\theta_{t+1} = \frac{\sqrt{\sum_{j>1}\lambda_j^2\alpha_j^2}}{\lambda_1\alpha_1} \le \frac{\lambda_2}{\lambda_1}\tan\theta_t$$

- With noise G satisfying the conditions ($|G_1|$, $\|G\|$ small enough),

$$\tan\theta_{t+1} \le \frac{\lambda_2\sqrt{\sum_{j>1}\alpha_j^2} + \|G\|}{\lambda_1\alpha_1 - |G_1|} \le \epsilon + (\frac{\lambda_2}{\lambda_1})^{1/4}\tan\theta_t$$

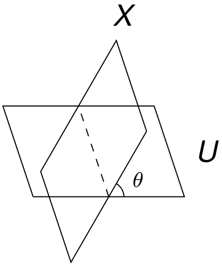
Noisy convergence proof (general k)

- Use "principal angle" θ from X to U
  let $U \in \mathbb{R}^{d \times k}$ have top k eigenvectors, $V = U^{\perp}$.



$$\tan\theta := \frac{\|V^T X\|}{\|U^T X\|} = \sqrt{\frac{\sum_{j>k} \alpha_j^2}{\sum_{j\le k} \alpha_j^2}}$$

Noisy convergence proof (general k)



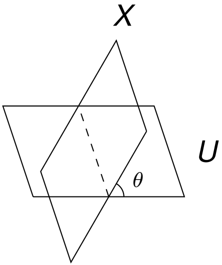
- Use "principal angle" θ from X to U
  let $U \in \mathbb{R}^{d \times k}$ have top k eigenvectors, $V = U^{\perp}$.

$$\tan\theta := \frac{\|V^T X\|}{\|U^T X\|} = \sqrt{\frac{\sum_{j>k} \alpha_j^2}{\sum_{j \le k} \alpha_j^2}}$$

- With no noise:

$$\tan\theta_{t+1} = \sqrt{\frac{\sum_{j>k} \lambda_j^2 \alpha_j^2}{\sum_{j \le k} \lambda_j^2 \alpha_j^2}}$$

Noisy convergence proof (general k)



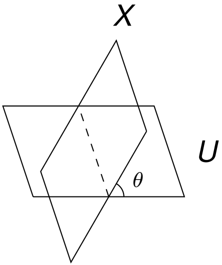
- Use "principal angle" θ from X to U
  let $U \in \mathbb{R}^{d \times k}$ have top k eigenvectors, $V = U^{\perp}$.

$$\tan\theta := \frac{\|V^T X\|}{\|U^T X\|} = \sqrt{\frac{\sum_{j>k} \alpha_j^2}{\sum_{j\le k} \alpha_j^2}}$$

- With no noise:

$$\tan\theta_{t+1} = \sqrt{\frac{\sum_{j>k} \lambda_j^2 \alpha_j^2}{\sum_{j\le k} \lambda_j^2 \alpha_j^2}} \le \frac{\lambda_{k+1}}{\lambda_k} \tan\theta_t$$

Noisy convergence proof (general k)



- Use "principal angle" θ from X to U
  let $U \in \mathbb{R}^{d\times k}$ have top k eigenvectors, $V = U^\perp$.

$$\tan\theta := \frac{\|V^T X\|}{\|U^T X\|} = \sqrt{\frac{\sum_{j>k}\alpha_j^2}{\sum_{j\leq k}\alpha_j^2}}$$

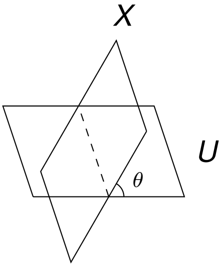
- With no noise:

$$\tan\theta_{t+1} = \sqrt{\frac{\sum_{j>k}\lambda_j^2\alpha_j^2}{\sum_{j\leq k}\lambda_j^2\alpha_j^2}} \leq \frac{\lambda_{k+1}}{\lambda_k}\tan\theta_t$$

- With noise G "small enough" we will have

$$\tan\theta_{t+1} \leq \frac{\lambda_{k+1}\|V^T X\| + \|G\|}{\lambda_k\|U^T X\| - \|U^T G\|}$$

Noisy convergence proof (general k)



- Use "principal angle" θ from X to U
  let $U \in \mathbb{R}^{d \times k}$ have top k eigenvectors, $V = U^{\perp}$.

$$\tan \theta := \frac{\|V^T X\|}{\|U^T X\|} = \sqrt{\frac{\sum_{j>k} \alpha_j^2}{\sum_{j \le k} \alpha_j^2}}$$

- With no noise:

$$\tan \theta_{t+1} = \sqrt{\frac{\sum_{j>k} \lambda_j^2 \alpha_j^2}{\sum_{j \le k} \lambda_j^2 \alpha_j^2}} \le \frac{\lambda_{k+1}}{\lambda_k} \tan \theta_t$$

- With noise G "small enough" we will have

$$\tan \theta_{t+1} \le \frac{\lambda_{k+1}\|V^T X\| + \|G\|}{\lambda_k \|U^T X\| - \|U^T G\|} \le \epsilon + (\frac{\lambda_{k+1}}{\lambda_k})^{1/4} \tan \theta_t$$

Noisy power method lemma

Theorem

Consider running the noisy power method on a random starting space $X_0 \in \mathbb{R}^{d \times k}$. Let $U \in \mathbb{R}^{d \times k}$ have top k eigenvectors of A. If

$$5\|G\| \leq \epsilon(\lambda_k - \lambda_{k+1}) \qquad 5\|U^T G\| \leq (\lambda_k - \lambda_{k+1})\frac{1}{\sqrt{kd}}$$

then after $L = O(\frac{\lambda_k}{\lambda_k - \lambda_{k+1}} \log(d/\epsilon))$ iterations,

$$\tan \Theta(X_L, U) \lesssim \epsilon \iff \|(I - X_L X_L^T)U\| \lesssim \epsilon$$

Noisy power method lemma

Theorem

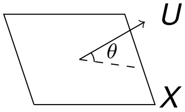
Consider running the noisy power method on a random starting space $X_0 \in \mathbb{R}^{d \times k}$. Let $U \in \mathbb{R}^{d \times k}$ have top k eigenvectors of A. If

$$5\|G\| \leq \epsilon(\lambda_k - \lambda_{k+1}) \qquad 5\|U^T G\| \leq (\lambda_k - \lambda_{k+1})\frac{1}{\sqrt{kd}}$$

then after $L = O(\frac{\lambda_k}{\lambda_k - \lambda_{k+1}} \log(d/\epsilon))$ iterations,

$$\tan \Theta(X_L, U) \lesssim \epsilon \iff \|(I - X_L X_L^T)U\| \lesssim \epsilon$$

- Can also iterate on a $p > k$ dimensional subspace.



Noisy power method lemma

Theorem

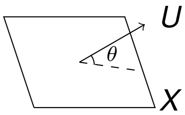
Consider running the noisy power method on a random starting space $X_0 \in \mathbb{R}^{d \times p}$. Let $U \in \mathbb{R}^{d \times k}$ have top k eigenvectors of A. If

$$5\|G\| \le \epsilon(\lambda_k - \lambda_{k+1}) \qquad 5\|U^T G\| \le (\lambda_k - \lambda_{k+1})\frac{\sqrt{p} - \sqrt{k-1}}{\sqrt{d}}$$

then after $L = O(\frac{\lambda_k}{\lambda_k - \lambda_{k+1}} \log(d/\epsilon))$ iterations,

$$\tan \Theta(X_L, U) \lesssim \epsilon \iff \|(I - X_L X_L^T)U\| \lesssim \epsilon$$

- Can also iterate on a $p > k$ dimensional subspace.
  - kth singular value of X is typically $\frac{\sqrt{p}-\sqrt{k-1}}{\sqrt{d}}$.



Noisy power method lemma

Theorem

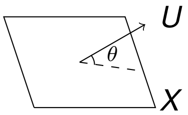
Consider running the noisy power method on a random starting space $X_0 \in \mathbb{R}^{d \times 2k}$. Let $U \in \mathbb{R}^{d \times k}$ have top k eigenvectors of A. If

$$5\|G\| \leq \epsilon(\lambda_k - \lambda_{k+1}) \qquad 5\|U^T G\| \leq (\lambda_k - \lambda_{k+1})\sqrt{\frac{k}{d}}$$

then after $L = O(\frac{\lambda_k}{\lambda_k - \lambda_{k+1}} \log(d/\epsilon))$ iterations,

$$\tan \Theta(X_L, U) \lesssim \epsilon \iff \|(I - X_L X_L^T)U\| \lesssim \epsilon$$

- Can also iterate on a $p > k$ dimensional subspace.
  - kth singular value of X is typically $\frac{\sqrt{p} - \sqrt{k-1}}{\sqrt{d}}$.



Noisy power method lemma

Theorem

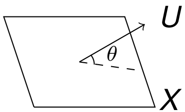
Consider running the noisy power method on a random starting space $X_0 \in \mathbb{R}^{d \times 2k}$. Let $U \in \mathbb{R}^{d \times k}$ have top k eigenvectors of A. If

$$5\|G\| \leq \epsilon(\lambda_k - \lambda_{k+1}) \qquad 5\|U^T G\| \leq (\lambda_k - \lambda_{k+1})\sqrt{\frac{k}{d}}$$

then after $L = O(\frac{\lambda_k}{\lambda_k - \lambda_{k+1}} \log(d/\epsilon))$ iterations,

$$\tan \Theta(X_L, U) \lesssim \epsilon \iff \|(I - X_L X_L^T)U\| \lesssim \epsilon$$

- Can also iterate on a $p > k$ dimensional subspace.
  - kth singular value of X is typically $\frac{\sqrt{p} - \sqrt{k-1}}{\sqrt{d}}$.
- If G is fairly uniform, expect $\|U^T G\| \approx \|G\|\sqrt{\frac{k}{d}}$



Noisy power method lemma

Theorem

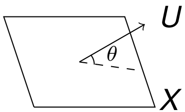
Consider running the noisy power method on a random starting space $X_0 \in \mathbb{R}^{d \times 2k}$. Let $U \in \mathbb{R}^{d \times k}$ have top k eigenvectors of A. If

$$5\|G\| \leq \epsilon(\lambda_k - \lambda_{k+1}) \qquad 5\|U^T G\| \leq (\lambda_k - \lambda_{k+1})\sqrt{\frac{k}{d}}$$

then after $L = O(\frac{\lambda_k}{\lambda_k - \lambda_{k+1}} \log(d/\epsilon))$ iterations,

$$\tan \Theta(X_L, U) \lesssim \epsilon \iff \|(I - X_L X_L^T)U\| \lesssim \epsilon$$

- Can also iterate on a $p > k$ dimensional subspace.
  - kth singular value of X is typically $\frac{\sqrt{p} - \sqrt{k-1}}{\sqrt{d}}$.
- If G is fairly uniform, expect $\|U^T G\| \approx \|G\| \sqrt{\frac{k}{d}}$
  - First condition is the main one, iteration will converge to to $\frac{\|G\|}{\lambda_k - \lambda_{k+1}}$.



Conjectures to remove eigengap

Theorem

Consider running the noisy power method on a random starting space $X_0 \in \mathbb{R}^{d \times 2k}$. Let $U \in \mathbb{R}^{d \times k}$ have top k eigenvectors of A. If

$$5\|G\| \leq \epsilon(\lambda_k - \lambda_{k+1}) \qquad 5\|U^T G\| \leq (\lambda_k - \lambda_{k+1})\sqrt{\frac{k}{d}}$$

at each iteration then after $L = O(\frac{\lambda_k}{\lambda_k - \lambda_{k+1}} \log(d/\epsilon))$ iterations,

$$\tan \Theta(X_L, U) \lesssim \epsilon \iff \|(I - X_L X_L^T)U\| \lesssim \epsilon$$

- If $\lambda_k = \lambda_{k+1}$, our theorem is useless.

Conjectures to remove eigengap

Conjecture (Can depend on $\lambda_k - \lambda_{2k+1}$ eigengap)

Consider running the noisy power method on a random starting space $X_0 \in \mathbb{R}^{d \times 2k}$. Let $U \in \mathbb{R}^{d \times k}$ have top k eigenvectors of A. If

$$5\|G\| \leq \epsilon(\lambda_k - \lambda_{2k+1}) \qquad 5\|U^T G\| \leq (\lambda_k - \lambda_{2k+1})\sqrt{\frac{k}{d}}$$

at each iteration then after $L = O(\frac{\lambda_k}{\lambda_k - \lambda_{2k+1}} \log(d/\epsilon))$ iterations,

$$\tan \Theta(X_L, U) \lesssim \epsilon \iff \|(I - X_L X_L^T)U\| \lesssim \epsilon$$

- If $\lambda_k = \lambda_{k+1}$, our theorem is useless.
- If X is $n \times p$, maybe the relevant eigengap is $\lambda_k - \lambda_{p+1}$?

Conjectures to remove eigengap

Conjecture (Can depend on $\lambda_k - \lambda_{2k+1}$ eigengap)

Consider running the noisy power method on a random starting space $X_0 \in \mathbb{R}^{d \times 2k}$. Let $U \in \mathbb{R}^{d \times k}$ have top k eigenvectors of A. If

$$5\|G\| \leq \epsilon(\lambda_k - \lambda_{2k+1}) \qquad 5\|U^T G\| \leq (\lambda_k - \lambda_{2k+1})\sqrt{\frac{k}{d}}$$

at each iteration then after $L = O(\frac{\lambda_k}{\lambda_k - \lambda_{2k+1}} \log(d/\epsilon))$ iterations,

$$\tan \Theta(X_L, U) \lesssim \epsilon \iff \|(I - X_L X_L^T)U\| \lesssim \epsilon$$

- If $\lambda_k = \lambda_{k+1}$, our theorem is useless.
- If X is $n \times p$, maybe the relevant eigengap is $\lambda_k - \lambda_{p+1}$?
- But do we need any eigengap at all?

Conjectures to remove eigengap

- Do we need any eigengap at all?

Conjectures to remove eigengap

- Do we need any eigengap at all?
- Yes for X to approximate U: $\|(I - XX^T)U\| \leq \epsilon$.

Conjectures to remove eigengap

- Do we need any eigengap at all?
- Yes for X to approximate U: $\|(I - XX^T)U\| \leq \epsilon$.
- Not clear for X to approximate A: $\|(I - XX^T)A\| \leq \lambda_{k+1} + \epsilon$.

Conjectures to remove eigengap

- Do we need any eigengap at all?
- Yes for X to approximate U: $\|(I - XX^T)U\| \leq \epsilon$.
- Not clear for X to approximate A: $\|(I - XX^T)A\| \leq \lambda_{k+1} + \epsilon$.
  - This is weaker: doesn't imply Frobenius approximation.

Conjectures to remove eigengap

- Do we need any eigengap at all?
- Yes for X to approximate U: $\|(I - XX^T)U\| \leq \epsilon$.
- Not clear for X to approximate A: $\|(I - XX^T)A\| \leq \lambda_{k+1} + \epsilon$.
  - This is weaker: doesn't imply Frobenius approximation.

Conjecture

Consider running the noisy power method on a random starting space $X_0 \in \mathbb{R}^{d \times 2k}$. Let $U \in \mathbb{R}^{d \times k}$ have top k eigenvectors of A. If

$$\|G\| \leq \epsilon \qquad \|U^T G\| \leq \epsilon\sqrt{\frac{k}{d}}$$

at each iteration then after $L = O(\frac{\lambda_{k+1}}{\epsilon}\log(d/\epsilon))$ iterations,

$$\|(I - X_L X_L^T)A\| \leq \lambda_{k+1} + O(\epsilon)$$

Review of our theorem

Theorem

Consider running the noisy power method on a random starting space $X_0 \in \mathbb{R}^{d\times 2k}$. Let $U \in \mathbb{R}^{d\times k}$ have the top k eigenvectors of A. If

$$5\|G\| \leq \epsilon(\lambda_k - \lambda_{k+1}) \qquad 5\|U^T G\| \leq (\lambda_k - \lambda_{k+1})\sqrt{\frac{k}{d}}$$

at each iteration then after $L = O(\frac{\lambda_k}{\lambda_k - \lambda_{k+1}} \log(d/\epsilon))$ iterations,

$$\tan\Theta(X_L, U) \lesssim \epsilon \iff \|(I - X_L X_L^T)U\| \lesssim \epsilon$$

- Gaussian G: if $G_{i,j} \sim N(0, \sigma^2)$ then $\|G\| \lesssim \sqrt{d}\sigma$, $\|U^T G\| \lesssim \sqrt{k}\sigma$ with high probability. Hence $\sigma = \epsilon(\lambda_k - \lambda_{k+1})/\sqrt{d}$ is tolerable.

Outline

1. Applications

Applications of the Noisy Power Method

- Will discuss two applications of our theorem:
  - Privacy-preserving spectral analysis [Hardt-Roth ’13]
  - Streaming PCA [Mitliagkas-Caramanis-Jain ’13]
- Both cases, get improved bound.

Privacy-preserving spectral analysis

- Can we find *differentially private* approximations to the top eigenvectors?

Privacy-preserving spectral analysis

- Can we find *differentially private* approximations to the top eigenvectors?
- Think of A as related to adjacency matrix for graph (web links, social network, etc.)

Privacy-preserving spectral analysis

- Can we find *differentially private* approximations to the top eigenvectors?
- Think of A as related to adjacency matrix for graph (web links, social network, etc.)
  - Top eigenvectors are useful to study and reveal (e.g. PageRank, Cheever cuts)

Privacy-preserving spectral analysis

- Can we find *differentially private* approximations to the top eigenvectors?
- Think of A as related to adjacency matrix for graph (web links, social network, etc.)
  - Top eigenvectors are useful to study and reveal (e.g. PageRank, Cheever cuts)
  - Don't want to reveal whether x and y are friends.

Privacy-preserving spectral analysis

- Can we find *differentially private* approximations to the top eigenvectors?
- Think of A as related to adjacency matrix for graph (web links, social network, etc.)
  - Top eigenvectors are useful to study and reveal (e.g. PageRank, Cheever cuts)
  - Don't want to reveal whether x and y are friends.
- Randomized algorithm f is (ϵ, δ) differentially private if: for any A, A' with $\|A - A'\| \leq 1$, and for any subset S of the range,

$$\Pr[f(A) \in S] \leq e^{\epsilon} \Pr[f(A') \in S] + \delta$$

Privacy-preserving spectral analysis

- Can we find *differentially private* approximations to the top eigenvectors?
- Think of A as related to adjacency matrix for graph (web links, social network, etc.)
  - Top eigenvectors are useful to study and reveal (e.g. PageRank, Cheever cuts)
  - Don't want to reveal whether x and y are friends.
- Randomized algorithm f is (ϵ, δ) differentially private if: for any A, A' with $\|A - A'\| \leq 1$, and for any subset S of the range,

$$\Pr[f(A) \in S] \leq e^{\epsilon} \Pr[f(A') \in S] + \delta$$

- Typical dependence is $\text{poly}(\frac{1}{\epsilon} \log(1/\delta))$.

Privacy-preserving spectral analysis

- Hardt-Roth '12: Noisy power method $X \to AX + G$ preserves privacy if $G_{i,j} \sim N(0, \sigma^2)$ for large enough σ at each stage.

Privacy-preserving spectral analysis

- Hardt-Roth '12: Noisy power method $X \to AX + G$ preserves privacy if $G_{i,j} \sim N(0, \sigma^2)$ for large enough σ at each stage.
- Apply our theorem to see how well the result approximates U.

Privacy-preserving spectral analysis

- Hardt-Roth '12: Noisy power method $X \to AX + G$ preserves privacy if $G_{i,j} \sim N(0, \sigma^2)$ for large enough σ at each stage.
- Apply our theorem to see how well the result approximates U.
- Complicated expression, but for example:

Privacy-preserving spectral analysis

- Hardt-Roth '12: Noisy power method $X \to AX + G$ preserves privacy if $G_{i,j} \sim N(0, \sigma^2)$ for large enough σ at each stage.
- Apply our theorem to see how well the result approximates U.
- Complicated expression, but for example:
  - Suppose A represents random graph with a planted sparse cut.

Privacy-preserving spectral analysis

- Hardt-Roth '12: Noisy power method $X \to AX + G$ preserves privacy if $G_{i,j} \sim N(0, \sigma^2)$ for large enough σ at each stage.
- Apply our theorem to see how well the result approximates U.
- Complicated expression, but for example:
  - Suppose A represents random graph with a planted sparse cut.
  - Then the (ϵ, δ)-differentially private result is very close to U.

Privacy-preserving spectral analysis

- Hardt-Roth '12: Noisy power method $X \to AX + G$ preserves privacy if $G_{i,j} \sim N(0, \sigma^2)$ for large enough σ at each stage.
- Apply our theorem to see how well the result approximates U.
- Complicated expression, but for example:
  - Suppose A represents random graph with a planted sparse cut.
  - Then the (ϵ, δ)-differentially private result is very close to U.
  - [Hardt-Roth: $k = 1$ case.]

Privacy-preserving spectral analysis

- Hardt-Roth '12: Noisy power method $X \to AX + G$ preserves privacy if $G_{i,j} \sim N(0, \sigma^2)$ for large enough σ at each stage.
- Apply our theorem to see how well the result approximates U.
- Complicated expression, but for example:
  - Suppose A represents random graph with a planted sparse cut.
  - Then the (ϵ, δ)-differentially private result is very close to U.
  - [Hardt-Roth: $k = 1$ case.]
- Added bonus: algorithm uses sparsity of A.

Streaming PCA

- Can take samples $x_1, x_2, \ldots \sim \mathcal{D}$ in $\mathbb{R}^d$.

Streaming PCA

- Can take samples $x_1, x_2, \ldots \sim \mathcal{D}$ in $\mathbb{R}^d$.
- Want to estimate covariance matrix $\Sigma = \mathbb{E}[xx^T]$.

Streaming PCA

- Can take samples $x_1, x_2, \ldots \sim \mathcal{D}$ in $\mathbb{R}^d$.
- Want to estimate covariance matrix $\Sigma = \mathbb{E}[xx^T]$.
- Easy answer is the empirical covariance:

$$\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T.$$

Streaming PCA

- Can take samples $x_1, x_2, \ldots \sim \mathcal{D}$ in $\mathbb{R}^d$.
- Want to estimate covariance matrix $\Sigma = \mathbb{E}[xx^T]$.
- Easy answer is the empirical covariance:

$$\widehat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T.$$

- But uses n^2 space. Can we use $O(nk)$ space if Σ is nearly low rank?

Streaming PCA

- Can take samples $x_1, x_2, \ldots \sim \mathcal{D}$ in $\mathbb{R}^d$.
- Want to estimate covariance matrix $\Sigma = \mathbb{E}[xx^T]$.
- Easy answer is the empirical covariance:

$$\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T.$$

- But uses n^2 space. Can we use $O(nk)$ space if Σ is nearly low rank?
- [Mitliagkas-Caramanis-Jain '13] Yes, using more samples. Can do one iteration of the power method in small space:

$$X_{t+1} = \widehat{\Sigma} X = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T X$$

Streaming PCA

- Algorithm is: in every iteration, take a bunch of samples to move in correct direction.

Streaming PCA

- Algorithm is: in every iteration, take a bunch of samples to move in correct direction.
- How many iterations, and how many samples per iteration?

Streaming PCA

- Algorithm is: in every iteration, take a bunch of samples to move in correct direction.
- How many iterations, and how many samples per iteration?
- This is then a noisy power method problem:

$$X_{t+1} = \Sigma x + (\widehat{\Sigma} - \Sigma)X$$

Streaming PCA

- Algorithm is: in every iteration, take a bunch of samples to move in correct direction.
- How many iterations, and how many samples per iteration?
- This is then a noisy power method problem:

$$X_{t+1} = \Sigma x + (\widehat{\Sigma} - \Sigma)X$$

- Just need to bound norm of $(\widehat{\Sigma} - \Sigma)X$ in each iteration.

Streaming PCA

- Algorithm is: in every iteration, take a bunch of samples to move in correct direction.
- How many iterations, and how many samples per iteration?
- This is then a noisy power method problem:

$$X_{t+1} = \Sigma x + (\widehat{\Sigma} - \Sigma)X$$

- Just need to bound norm of $(\widehat{\Sigma} - \Sigma)X$ in each iteration.
- Nice case: spiked covariance model

Streaming PCA

- Algorithm is: in every iteration, take a bunch of samples to move in correct direction.
- How many iterations, and how many samples per iteration?
- This is then a noisy power method problem:

$$X_{t+1} = \Sigma x + (\widehat{\Sigma} - \Sigma)X$$

- Just need to bound norm of $(\widehat{\Sigma} - \Sigma)X$ in each iteration.
- Nice case: spiked covariance model
  - Gaussian, where Σ has k eigenvalues $\lambda_1, \ldots, \lambda_k = \Theta(1)$ perturbed by Gaussian noise $N(0, \sigma^2)$ in each coordinate.

Streaming PCA

- Algorithm is: in every iteration, take a bunch of samples to move in correct direction.
- How many iterations, and how many samples per iteration?
- This is then a noisy power method problem:

$$X_{t+1} = \Sigma x + (\widehat{\Sigma} - \Sigma)X$$

- Just need to bound norm of $(\widehat{\Sigma} - \Sigma)X$ in each iteration.
- Nice case: spiked covariance model
  - Gaussian, where Σ has k eigenvalues $\lambda_1, \ldots, \lambda_k = \Theta(1)$ perturbed by Gaussian noise $N(0, \sigma^2)$ in each coordinate.
  - $\widetilde{O}(\frac{1+\sigma^6}{\epsilon^2} dk)$ samples suffice.

Streaming PCA

- Algorithm is: in every iteration, take a bunch of samples to move in correct direction.
- How many iterations, and how many samples per iteration?
- This is then a noisy power method problem:

$$X_{t+1} = \Sigma x + (\widehat{\Sigma} - \Sigma)X$$

- Just need to bound norm of $(\widehat{\Sigma} - \Sigma)X$ in each iteration.
- Nice case: spiked covariance model
  - Gaussian, where Σ has k eigenvalues $\lambda_1, \ldots, \lambda_k = \Theta(1)$ perturbed by Gaussian noise $N(0, \sigma^2)$ in each coordinate.
  - $\widetilde{O}(\frac{1+\sigma^6}{\epsilon^2} dk)$ samples suffice.
  - Factor k improvement on [Mitliagkas-Caramanis-Jain '13]

Streaming PCA

- Algorithm is: in every iteration, take a bunch of samples to move in correct direction.
- How many iterations, and how many samples per iteration?
- This is then a noisy power method problem:

$$X_{t+1} = \Sigma x + (\widehat{\Sigma} - \Sigma)X$$

- Just need to bound norm of $(\widehat{\Sigma} - \Sigma)X$ in each iteration.
- Nice case: spiked covariance model
  - Gaussian, where Σ has k eigenvalues $\lambda_1, \ldots, \lambda_k = \Theta(1)$ perturbed by Gaussian noise $N(0, \sigma^2)$ in each coordinate.
  - $\widetilde{O}(\frac{1+\sigma^6}{\epsilon^2} dk)$ samples suffice.
  - Factor k improvement on [Mitliagkas-Caramanis-Jain '13]
- But also applies to less nice cases:

Streaming PCA

- Algorithm is: in every iteration, take a bunch of samples to move in correct direction.
- How many iterations, and how many samples per iteration?
- This is then a noisy power method problem:

$$X_{t+1} = \Sigma x + (\widehat{\Sigma} - \Sigma)X$$

- Just need to bound norm of $(\widehat{\Sigma} - \Sigma)X$ in each iteration.
- Nice case: spiked covariance model
  - Gaussian, where Σ has k eigenvalues $\lambda_1, \ldots, \lambda_k = \Theta(1)$ perturbed by Gaussian noise $N(0, \sigma^2)$ in each coordinate.
  - $\widetilde{O}(\frac{1+\sigma^6}{\epsilon^2} dk)$ samples suffice.
  - Factor k improvement on [Mitliagkas-Caramanis-Jain '13]
- But also applies to less nice cases:
  - Strong whenever $\mathcal{D}$ has exponential concentration.

Streaming PCA

- Algorithm is: in every iteration, take a bunch of samples to move in correct direction.
- How many iterations, and how many samples per iteration?
- This is then a noisy power method problem:

$$X_{t+1} = \Sigma x + (\widehat{\Sigma} - \Sigma)X$$

- Just need to bound norm of $(\widehat{\Sigma} - \Sigma)X$ in each iteration.
- Nice case: spiked covariance model
  - Gaussian, where Σ has k eigenvalues $\lambda_1, \ldots, \lambda_k = \Theta(1)$ perturbed by Gaussian noise $N(0, \sigma^2)$ in each coordinate.
  - $\widetilde{O}(\frac{1+\sigma^6}{\epsilon^2} dk)$ samples suffice.
  - Factor k improvement on [Mitliagkas-Caramanis-Jain '13]
- But also applies to less nice cases:
  - Strong whenever $\mathcal{D}$ has exponential concentration.
  - Nontrivial result for general distributions.

Recap and open questions

- Noisy power method is a useful tool.

Recap and open questions

- Noisy power method is a useful tool.
- Can show $\tan \Theta(X_L, U) \leq \epsilon$ if

$$5\|G\| \leq \epsilon(\lambda_k - \lambda_{k+1}) \qquad 5\|U^T G\| \leq (\lambda_k - \lambda_{k+1})\sqrt{\frac{k}{d}}$$

Recap and open questions

- Noisy power method is a useful tool.
- Can show $\tan \Theta(X_L, U) \leq \epsilon$ if

$$5\|G\| \leq \epsilon(\lambda_k - \lambda_{k+1}) \qquad 5\|U^T G\| \leq (\lambda_k - \lambda_{k+1})\sqrt{\frac{k}{d}}$$

- Can we apply it to more problems?

Recap and open questions

- Noisy power method is a useful tool.
- Can show $\tan \Theta(X_L, U) \leq \epsilon$ if

$$5\|G\| \leq \epsilon(\lambda_k - \lambda_{k+1}) \qquad 5\|U^T G\| \leq (\lambda_k - \lambda_{k+1})\sqrt{\frac{k}{d}}$$

- Can we apply it to more problems?
- Can we prove a theorem without the eigengap?