COMPSCI 514: ALGORITHMS FOR DATA SCIENCE

Cameron Musco

University of Massachusetts Amherst. Fall 2019.

Lecture 17

- Problem Set 3 was released on Saturday. Due next Friday 11/15 at 8pm.
- I will hold office hours after class until 12:30pm today.

Last Class:

- FInished up spectral clustering and stochastic block model.
- Started discussion of efficient algorithms for SVD/eigendecomposition.

This Class:

- Finish efficient algorithms for SVD/eigendecomposition.
- Iterative methods: power method, Krylov subspace methods.
- ~~Start optimization unit. Gradient Descent~~.

FASTER ALGORITHMS

$O(nd^2)$

To speed up SVD computation we will take advantage of the fact that we typically only care about computing the top (or bottom) k singular vectors of a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ for $k \ll d$.

- Suffices to compute $\mathbf{V}_k \in \mathbb{R}^{d\times k}$ and then compute $\mathbf{U}_k\mathbf{\Sigma}_k = \mathbf{X}\mathbf{V}_k$.
  $X = U\Sigma V^T$ $\quad XV_k \quad U\Sigma V^T V_k = U_k\Sigma_k$
- Use an *iterative algorithm* to compute an *approximation* to the top k singular vectors $\mathbf{V}_k$.
- Runtime will be roughly $O(ndk)$ instead of $O(nd^2)$.

To speed up SVD computation we will take advantage of the fact that we typically only care about computing the top (or bottom) k singular vectors of a matrix $\mathbf{X} \in \mathbb{R}^{n \times k}$ for $k \ll d$.

- Suffices to compute $\mathbf{V}_k \in \mathbb{R}^{d \times k}$ and then compute $\mathbf{U}_k \boldsymbol{\Sigma}_k = \mathbf{X}\mathbf{V}_k$.
- Use an *iterative algorithm* to compute an *approximation* to the top k singular vectors $\mathbf{V}_k$.
- Runtime will be roughly $O(ndk)$ instead of $O(nd^2)$.

Sparse (iterative) vs. Direct Method. `svd` vs. `svds`.

Power Method: The most fundamental iterative method for approximate SVD. Applies to computing $k = 1$ singular vectors, but can easily be generalized to larger k.

POWER METHOD

Power Method: The most fundamental iterative method for approximate SVD. Applies to computing $k = 1$ singular vectors, but can easily be generalized to larger k.

Goal: Given $\mathbf{X} \in \mathbb{R}^{n \times d}$, with SVD $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, find $\vec{z} \approx \vec{v}_1$.

Power Method: The most fundamental iterative method for approximate SVD. Applies to computing $k = 1$ singular vectors, but can easily be generalized to larger k.

Goal: Given $\mathbf{X} \in \mathbb{R}^{n\times d}$, with SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$, find $\vec{z} \approx \vec{v}_1$.

- **Initialize:** Choose $\vec{z}^{(0)}$ randomly. E.g. $\vec{z}^{(0)}(i) \sim \mathcal{N}(0, 1)$. $\in \mathbb{R}^d$
- For $i = 1, \ldots, t$
  - $\vec{z}^{(i)} = (\mathbf{X}^T\mathbf{X}) \cdot \vec{z}^{(i-1)}$
  - $n_i = \|\vec{z}^{(i)}\|_2$
  - $\vec{z}^{(i)} = \vec{z}^{(i)}/n_i$

  Return $\vec{z}_t$

Power Method: The most fundamental iterative method for approximate SVD. Applies to computing $k = 1$ singular vectors, but can easily be generalized to larger k.

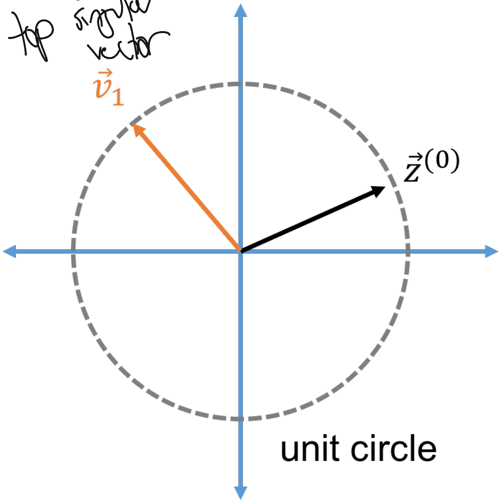
Goal: Given $\mathbf{X} \in \mathbb{R}^{n \times d}$, with SVD $\mathbf{X} = \mathbf{U\Sigma V}$, find $\vec{z} \approx \vec{v}_1$.

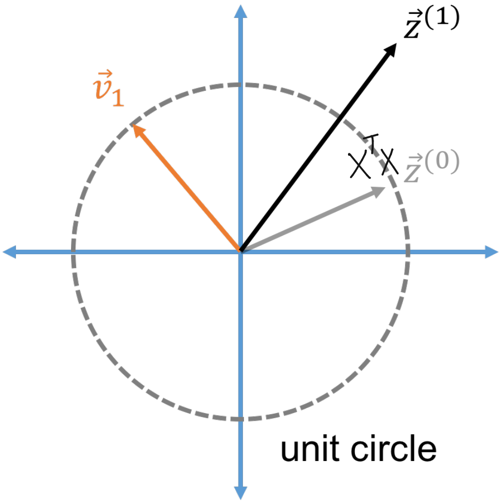
- **Initialize:** Choose $\vec{z}^{(0)}$ randomly. E.g. $\vec{z}^{(0)}(i) \sim \mathcal{N}(0, 1)$.
- For $i = 1, \ldots, t$ $\quad X^T(Xz^{i-1})$
  - $\vec{z}^{(i)} = (\mathbf{X}^T\mathbf{X}) \cdot \vec{z}^{(i-1)}$ $\quad$ Runtime: $2 \cdot nd$ $\quad X^TX = O(nd^2)$
  - $n_i = \|\vec{z}^{(i)}\|_2$ $\quad$ Runtime: d
  - $\vec{z}^{(i)} = \vec{z}^{(i)}/n_i$ $\quad$ Runtime: d $\quad O(nd)$

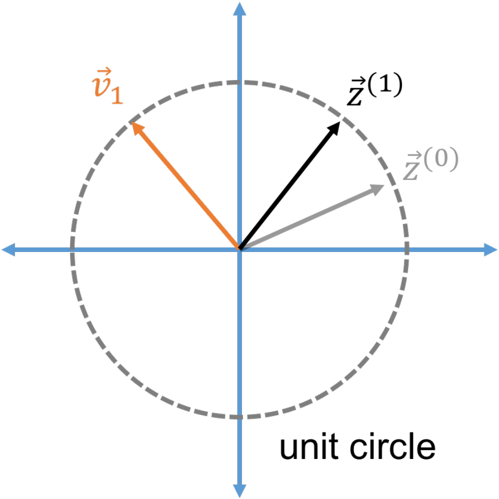
  Return $\vec{z}_t$

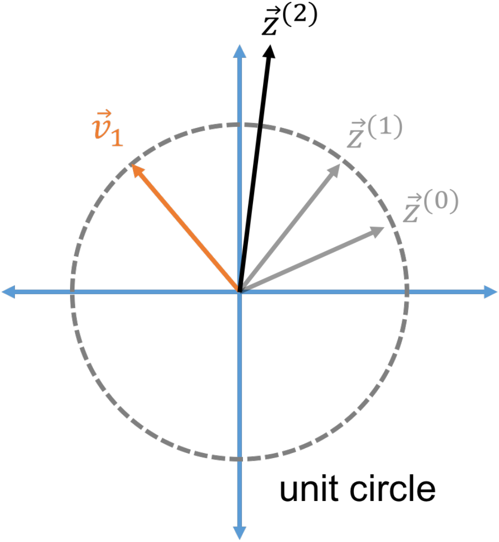
Total Runtime: $O(ndt)$ $\simeq O(nd)$

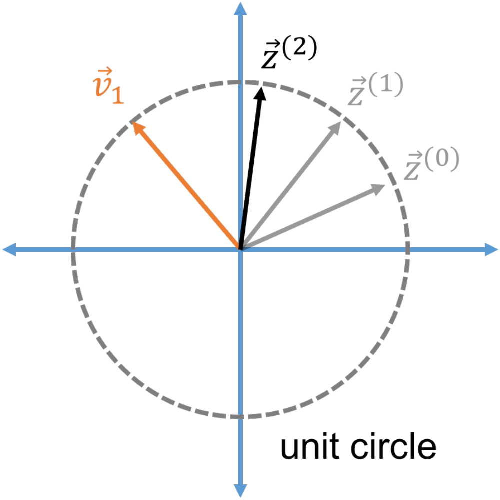
top singular
vector

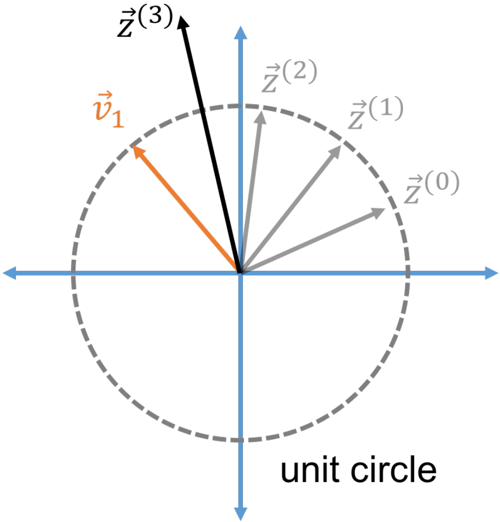


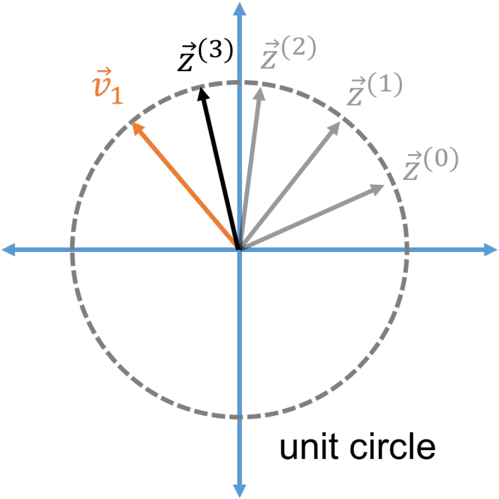

$\vec{z}^{(1)}$
$\vec{v}_1$
$X^T X$
$\vec{z}^{(0)}$
unit circle

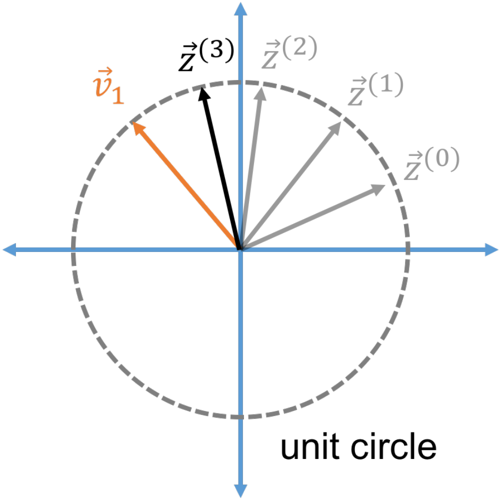

$\vec{v}_1$
$\vec{z}^{(1)}$
$\vec{z}^{(0)}$
unit circle


$\vec{z}^{(2)}$
$\vec{v}_1$
$\vec{z}^{(1)}$
$\vec{z}^{(0)}$
unit circle


$\vec{z}^{(2)}$
$\vec{v}_1$
$\vec{z}^{(1)}$
$\vec{z}^{(0)}$
unit circle


$\vec{z}^{(3)}$
$\vec{z}^{(2)}$
$\vec{v}_1$
$\vec{z}^{(1)}$
$\vec{z}^{(0)}$
unit circle


$\vec{z}^{(3)}$
$\vec{z}^{(2)}$
$\vec{v}_1$
$\vec{z}^{(1)}$
$\vec{z}^{(0)}$
unit circle



Why is it converging towards $\vec{v}_1$?

random

Write $\vec{z}^{(0)}$ in the right singular vector basis:

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d.$$

$\mathbf{X} \in \mathbb{R}^{n \times d}$: input matrix with SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

POWER METHOD INTUITION

Write $\vec{z}^{(0)}$ in the right singular vector basis:

$X = U\Sigma V^T$

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d.$$

Update step: $\vec{z}^{(i)} = \overset{T}{\mathbf{X}}\mathbf{X}^{\cancel{T}} \cdot \vec{z}^{(i-1)} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T \cdot \vec{z}^{(i-1)}$ (then normalize)

$\mathbf{X} \in \mathbb{R}^{n\times d}$: input matrix with SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

Write $\vec{z}^{(0)}$ in the right singular vector basis:

$$* \quad \vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d.$$

Update step: $\vec{z}^{(i)} = \mathbf{X}\mathbf{X}^T \cdot \vec{z}^{(i-1)} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T \cdot \vec{z}^{(i-1)}$ (then normalize)

$$\mathbf{V}^T\vec{z}^{(0)} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_d \end{bmatrix}$$

$$V^T z^0 = V^T(c_1 v_1 + c_2 v_2 \ldots)$$

$$V^T c_1 v_1 = c_1 V^T v_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$c_1 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + c_2 \begin{bmatrix} 0 \\ 1 \\ - \\ 0 \end{bmatrix}$$

$\mathbf{X} \in \mathbb{R}^{n \times d}$: input matrix with SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

Write $\vec{z}^{(0)}$ in the right singular vector basis:

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d.$$

Update step: $\vec{z}^{(i)} = \mathbf{X}\mathbf{X}^T \cdot \vec{z}^{(i-1)} = \mathbf{V}\left(\boldsymbol{\Sigma}^2\mathbf{V}^T \cdot \vec{z}^{(i-1)}\right)$ (then normalize)

$$\mathbf{V}^T\vec{z}^{(0)} = \begin{bmatrix} c_1 \\ \vdots \\ c_d \end{bmatrix}$$

$$\boldsymbol{\Sigma}^2\mathbf{V}^T\vec{z}^{(0)} = \begin{bmatrix} \sigma_1^2 \cdot c_1 \\ \sigma_2^2 \cdot c_2 \\ \vdots \\ \sigma_d^2 \cdot c_d \end{bmatrix}$$

$\mathbf{X} \in \mathbb{R}^{n \times d}$: input matrix with SVD $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

Write $\vec{z}^{(0)}$ in the right singular vector basis:

$$\vec{z}^{(0)} = \underline{c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d}.$$

Update step: $\vec{z}^{(i)} = \mathbf{X}^T\mathbf{X} \cdot \vec{z}^{(i-1)} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T \cdot \vec{z}^{(i-1)}$ (then normalize)

$$\mathbf{V}^T\vec{z}^{(0)} =$$

$$\mathbf{\Sigma}^2\mathbf{V}^T\vec{z}^{(0)} = \begin{bmatrix} \sigma_1^2 c_1 \\ \vdots \\ \sigma_d^2 c_d \end{bmatrix}$$

$$\vec{z}^{(1)} = \mathbf{V}(\mathbf{\Sigma}^2\mathbf{V}^T \cdot \vec{z}^{(0)}) = V\begin{bmatrix} \sigma_1^2 c_1 \\ \sigma_2^2 c_d \end{bmatrix} = \underline{\sigma_1^2 c_1 v_1 + \sigma_2^2 c_2 v_2 + \ldots \sigma_d^2 c_d v_d}$$

$\mathbf{X} \in \mathbb{R}^{n\times d}$: input matrix with SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

Claim 1 : Writing $\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d$,

$$\vec{z}^{(1)} = c_1 \cdot \sigma_1^2\vec{v}_1 + c_2 \cdot \sigma_2^2\vec{v}_2 + \ldots + c_d \cdot \sigma_d^2\vec{v}_d.$$

$\mathbf{X} \in \mathbb{R}^{n\times d}$: input matrix with SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

Claim 1: Writing $\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d$,

$$\vec{z}^{(1)} = c_1 \cdot \sigma_1^2\vec{v}_1 + c_2 \cdot \sigma_2^2\vec{v}_2 + \ldots + c_d \cdot \sigma_d^2\vec{v}_d.$$

$$\vec{z}^{(2)} = \mathbf{X}^T\mathbf{X}\vec{z}^{(1)} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T\vec{z}^{(1)} = c_1 \sigma_1^4 v_1 + \ldots \; c_d \sigma_d^4 v_d$$

$\mathbf{X} \in \mathbb{R}^{n\times d}$: input matrix with SVD $\mathbf{X} = \mathbf{U\Sigma V}^T$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

POWER METHOD INTUITION

Claim 1 : Writing $\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d$,

$$\vec{z}^{(1)} = c_1 \cdot \sigma_1^2\vec{v}_1 + c_2 \cdot \sigma_2^2\vec{v}_2 + \ldots + c_d \cdot \sigma_d^2\vec{v}_d.$$

$$\vec{z}^{(2)} = \mathbf{X}^T\mathbf{X}\vec{z}^{(1)} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T\vec{z}^{(1)} =$$

Claim 2:

$$\vec{z}^{(t)} = c_1 \cdot \sigma_1^{2t}\vec{v}_1 + c_2 \cdot \sigma_2^{2t}\vec{v}_2 + \ldots + c_d \cdot \sigma_d^{2t}\vec{v}_d.$$

$\|\vec{z}^{(t)}\|$

$\mathbf{X} \in \mathbb{R}^{n\times d}$: input matrix with SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

After t iterations, we have 'powered' up the singular values, making the component in the direction of v_1 much larger, relative to the other components.

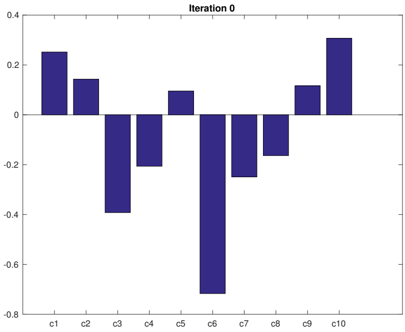
$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$

POWER METHOD CONVERGENCE

After t iterations, we have 'powered' up the singular values, making the component in the direction of v_1 much larger, relative to the other components.

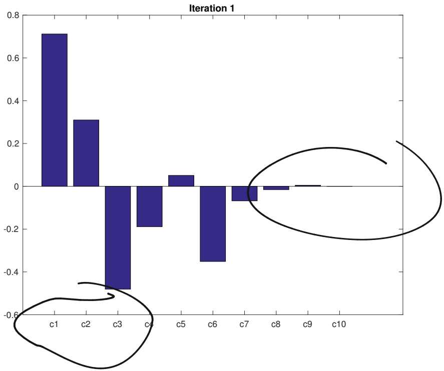
$c_1^t \qquad c_2^t \qquad c_d^t$

$$\boxed{\vec{z}^{(0)}} = \underline{c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d} \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



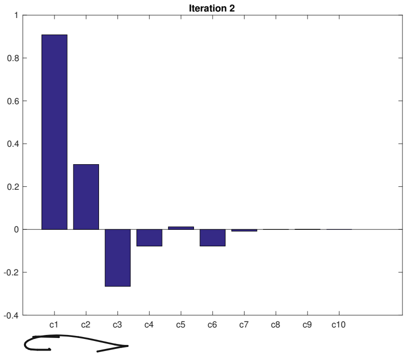
After t iterations, we have 'powered' up the singular values, making the component in the direction of v_1 much larger, relative to the other components.

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



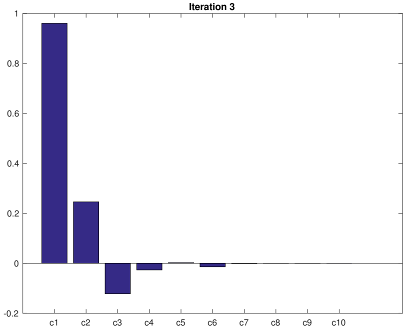
After t iterations, we have 'powered' up the singular values, making the component in the direction of v_1 much larger, relative to the other components.

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



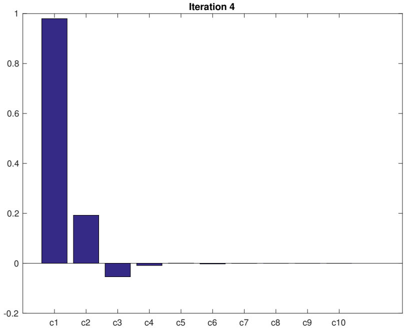
After t iterations, we have 'powered' up the singular values, making the component in the direction of v_1 much larger, relative to the other components.

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



After t iterations, we have 'powered' up the singular values, making the component in the direction of v_1 much larger, relative to the other components.

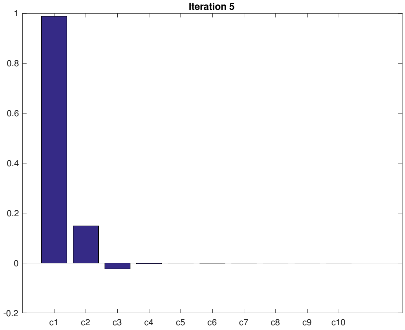
$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



POWER METHOD CONVERGENCE

After t iterations, we have 'powered' up the singular values, making the component in the direction of v_1 much larger, relative to the other components.

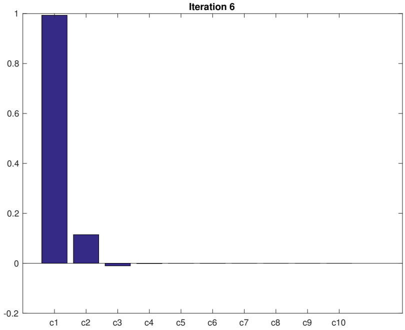
$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



POWER METHOD CONVERGENCE

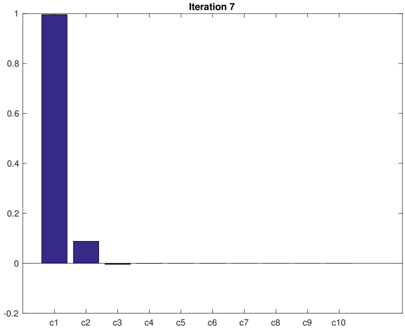
After t iterations, we have 'powered' up the singular values, making the component in the direction of v_1 much larger, relative to the other components.

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



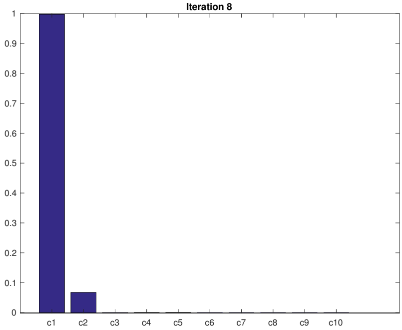
After t iterations, we have 'powered' up the singular values, making the component in the direction of v_1 much larger, relative to the other components.

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



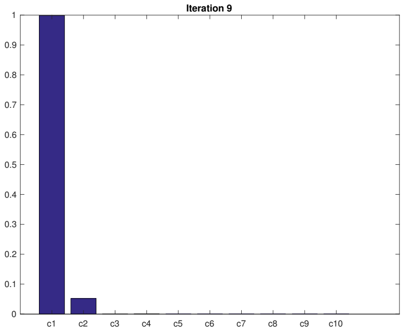
After t iterations, we have 'powered' up the singular values, making the component in the direction of v_1 much larger, relative to the other components.

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



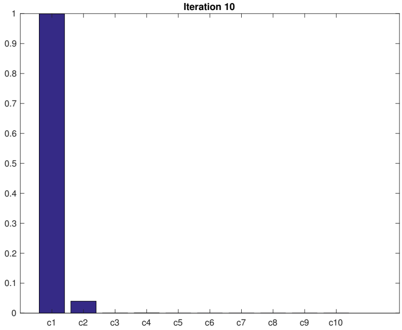
After t iterations, we have 'powered' up the singular values, making the component in the direction of v_1 much larger, relative to the other components.

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



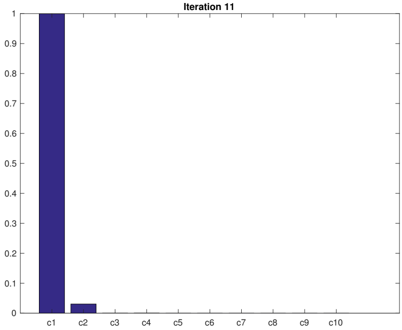
After t iterations, we have 'powered' up the singular values, making the component in the direction of v_1 much larger, relative to the other components.

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



After t iterations, we have 'powered' up the singular values, making the component in the direction of v_1 much larger, relative to the other components.

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$

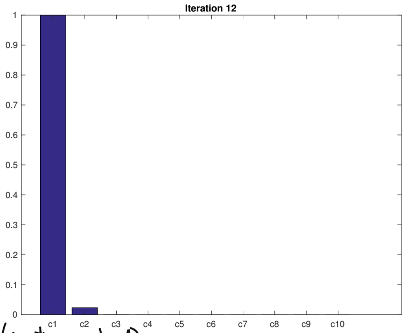


After t iterations, we have 'powered' up the singular values, making the component in the direction of v_1 much larger, relative to the other components.

c_1 positive or negative

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$

$z^{(t)} \approx v_1 \cdot c\sigma_1^{2t}$

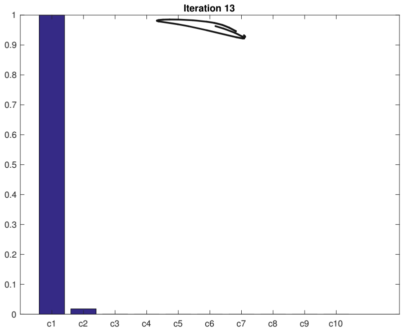


$z^{(0)} = 1 \cdot v_1 + 0 \cdot v_2 + \ldots + 0 \cdot v_d$

POWER METHOD CONVERGENCE

After t iterations, we have 'powered' up the singular values, making the component in the direction of v_1 much larger, relative to the other components.

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



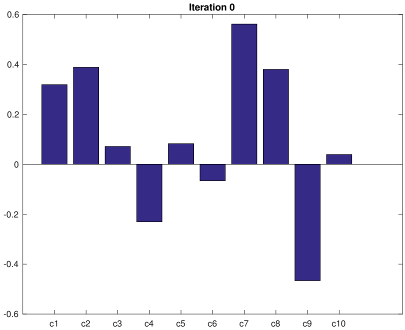
When will convergence be slow?

Slow Case: X has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$

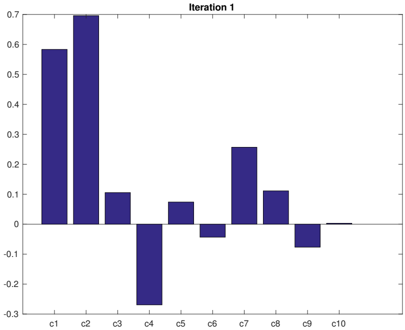
Slow Case: X has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



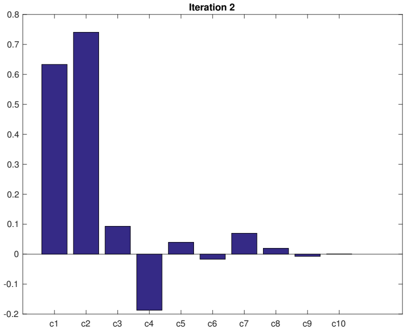
Slow Case: X has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



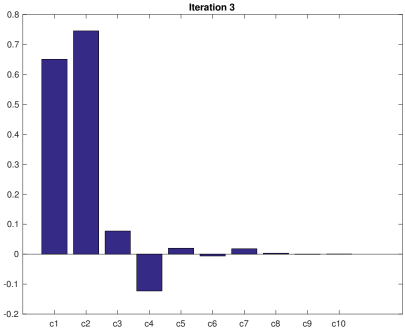
Slow Case: X has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



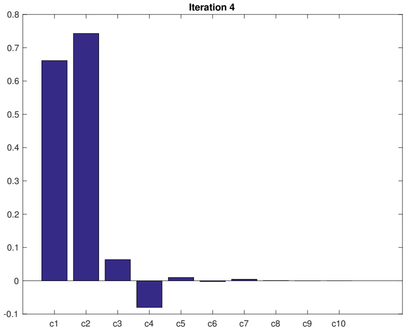
Slow Case: X has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



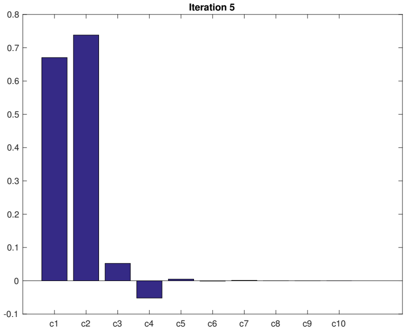
Slow Case: X has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



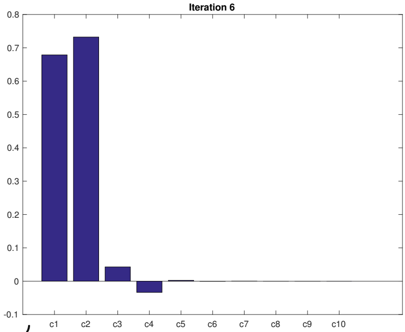
Slow Case: X has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



Slow Case: X has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

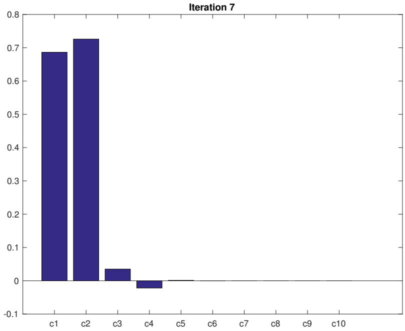
$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



$c_i \sigma_i^{2t}$

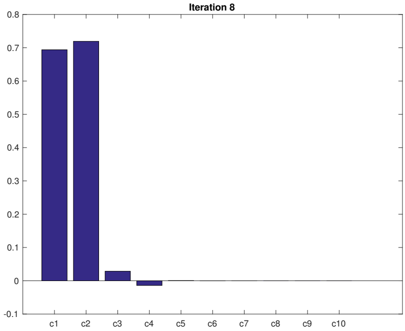
Slow Case: X has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



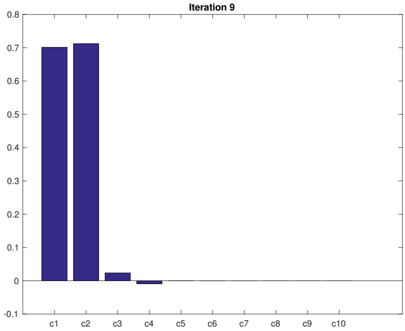
Slow Case: X has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



Slow Case: X has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

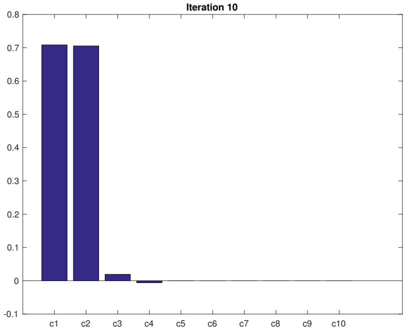
$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



POWER METHOD SLOW CONVERGENCE

Slow Case: X has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

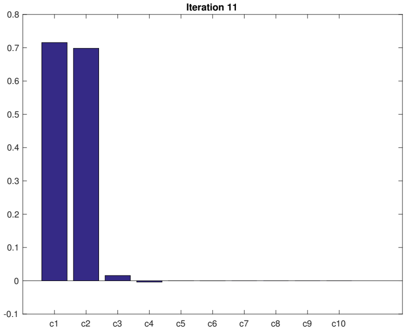
$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



POWER METHOD SLOW CONVERGENCE

Slow Case: X has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

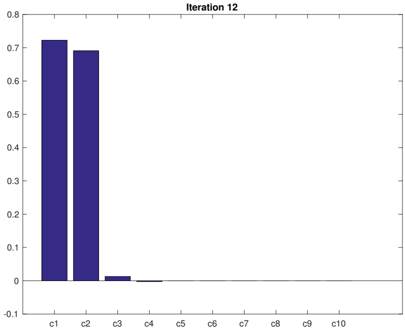
$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



POWER METHOD SLOW CONVERGENCE

Slow Case: X has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

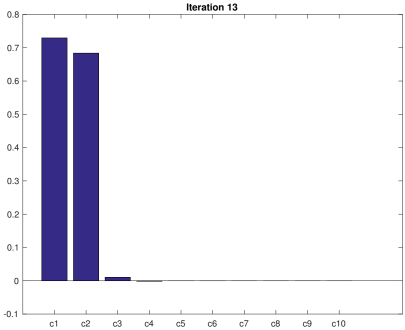
$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$



Slow Case: X has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$

$z^t \neq v_1$



$\gamma = .01$ "Rule of 70"

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$

Write $\underline{\sigma_2 = (1-\gamma)}\sigma_1$ for 'gap' $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$. How many iterations t does it take to have $\sigma_2^{2t} \leq \frac{1}{2} \cdot \sigma_1^{2t}$?

$$\sigma_2^{2t} \leq \frac{1}{2}\sigma_1^{2t} \rightarrow \left(\frac{\sigma_2}{\sigma_1}\right)^{2t} \leq \frac{1}{2}$$

$$(1-\gamma)^{2t} \leq \frac{1}{2} \qquad \underline{2t = \log_{1-\gamma}(1/2)}$$

$$(1-\gamma)^{1/\gamma} = 1/e$$

$$t = O(1/\gamma) \rightarrow (1-\gamma)^{2t} \leq \frac{1}{2}$$

$\mathbf{X} \in \mathbb{R}^{n\times d}$: matrix with SVD $\mathbf{X} = \mathbf{U\Sigma V}^T$. Singular values $\sigma_1, \sigma_2, \ldots, \sigma_d$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$

Write $\sigma_2 = (1-\gamma)\sigma_1$ for 'gap' $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$. How many iterations t does it take to have $\sigma_2^{2t} \le \frac{1}{2} \cdot \sigma_1^{2t}$? $O(1/\gamma)$. $O(\log_{1-\gamma} 1/2)$

$\mathbf{X} \in \mathbb{R}^{n \times d}$: matrix with SVD $\mathbf{X} = \mathbf{U\Sigma V}^T$. Singular values $\sigma_1, \sigma_2, \ldots, \sigma_d$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$

Write $\sigma_2 = (1-\gamma)\sigma_1$ for 'gap' $\gamma = \frac{\sigma_1-\sigma_2}{\sigma_1}$. How many iterations t does it take to have $\sigma_2^{2t} \le \frac{1}{2}\cdot\sigma_1^{2t}$? $O(1/\gamma)$.

How many iterations t does it take to have $\sigma_2^{2t} \le \delta\cdot\sigma_1^{2t}$?

$\log_{1-\gamma}\delta$

$O(1/\gamma)$

$\sigma_2^{2t} \le \frac{1}{2}\sigma_1^{2t}$

$\sigma_2^{2t\cdot m} \le \frac{1}{2^m}\cdot\sigma_1^{2t}$ $\quad m = \log(1/\delta)$

$O(1/\gamma)\cdot O(\log(1/\delta)) = \frac{\log(1/\delta)}{\gamma}$

$\mathbf{X} \in \mathbb{R}^{n\times d}$: matrix with SVD $\mathbf{X} = \mathbf{U\Sigma V}^T$. Singular values $\sigma_1, \sigma_2, \ldots, \sigma_d$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

POWER METHOD CONVERGENCE RATE

$$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$

Write $\sigma_2 = (1-\gamma)\sigma_1$ for 'gap' $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$. How many iterations t does it take to have $\sigma_2^{2t} \leq \frac{1}{2} \cdot \sigma_1^{2t}$? $O(1/\gamma)$.

How many iterations t does it take to have $\sigma_2^{2t} \leq \delta \cdot \sigma_1^{2t}$? $O\left(\frac{\log(1/\delta)}{\gamma}\right)$.

$\mathbf{X} \in \mathbb{R}^{n \times d}$: matrix with SVD $\mathbf{X} = \mathbf{U\Sigma V}^T$. Singular values $\sigma_1, \sigma_2, \ldots, \sigma_d$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

$\vec{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$

Write $\sigma_2 = (1-\gamma)\sigma_1$ for 'gap' $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$. How many iterations t does it take to have $\sigma_2^{2t} \leq \frac{1}{2} \cdot \sigma_1^{2t}$? $O(1/\gamma)$.

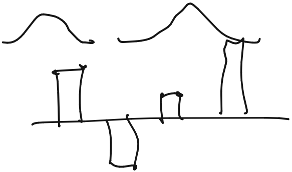
How many iterations t does it take to have $\sigma_2^{2t} \leq \delta \cdot \sigma_1^{2t}$? $O\left(\frac{\log(1/\delta)}{\gamma}\right)$.

How small must we set δ to ensure that $c_1\sigma_1^{2t}$ dominates all other components and so $\vec{z}^{(t)}$ is very close to $\vec{v}_1$?

$\mathbf{X} \in \mathbb{R}^{n\times d}$: matrix with SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Singular values $\sigma_1, \sigma_2, \ldots, \sigma_d$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

Claim: When $z^{(0)}$ is chosen with random Gaussian entries, writing $z^{(0)} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \ldots + c_d\mathbf{v}_d$, with very high probability, for all i:

$$\underline{O(1/d^2) \leq c_i \leq O(\sqrt{\log d})}$$



How is c_i distributed?

$$V^T z^{(0)} = \begin{bmatrix} c_1 \\ \vdots \\ c_d \end{bmatrix}$$

$$c_i = \langle v_i, z^{(0)} \rangle = \underbrace{\sum v_i(j) \cdot z^{(0)}(j)}$$

$N(0,1)$

$N(0, v_i(j)^2)$

$$e^{-x^2} \quad e^{-(\sqrt{\log})^2} = \frac{1}{d}$$

$c_i \sim N(0,1)$

"rotation invariance of Gaussian"

$c_i \sim N(0, \sum v_i(j)^2) = N(0, \|v_i\|_2^2)$

$\mathbf{X} \in \mathbb{R}^{n \times d}$: matrix with SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Singular values $\sigma_1, \sigma_2, \ldots, \sigma_d$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

Claim: When $z^{(0)}$ is chosen with random Gaussian entries, writing $z^{(0)} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \ldots + c_d\mathbf{v}_d$, with very high probability, for all i:

$\log d \geq 1$

$$\underline{O(1/d^2)} \leq |c_i| \leq \underline{O(\log d)}$$

Corollary:

$$\max_j \left|\frac{c_j}{c_1}\right| \leq O(d^2 \log d).$$

$\left|\frac{c_j}{c_1}\right| \leq \frac{\log(d)}{c_1} \leq \frac{\log d}{1/d^2} \leq d^2 \log d$

$\mathbf{X} \in \mathbb{R}^{n\times d}$: matrix with SVD $\mathbf{X} = \mathbf{U\Sigma V}^T$. Singular values $\sigma_1, \sigma_2, \ldots, \sigma_d$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

Claim 1: When $z^{(0)}$ is chosen with random Gaussian entries, writing $z^{(0)} = c_1\mathbf{v}_1 + \mathbf{c}_2\mathbf{v}_2 + \ldots + c_d\mathbf{v}_d$, with very high probability, $\max_j \frac{c_j}{c_1} \leq O(d^2 \log d)$.

$\mathbf{X} \in \mathbb{R}^{n\times d}$: matrix with SVD $\mathbf{X} = \mathbf{U\Sigma V}^T$. Singular values $\sigma_1, \sigma_2, \ldots, \sigma_d$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

Claim 1: When $z^{(0)}$ is chosen with random Gaussian entries, writing $z^{(0)} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \ldots + c_d\mathbf{v}_d$, with very high probability, $\max_j \frac{c_j}{c_1} \leq O(d^2 \log d)$.

Claim 2: For gap $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$, after $t = O\left(\frac{\log(1/\delta)}{\gamma}\right)$ iterations:

$$\vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d \propto \underline{c_1}\vec{v}_1 + \underline{c_2\delta}\vec{v}_2 + \ldots + \underline{c_d\delta}\vec{v}_d$$

$\mathbf{X} \in \mathbb{R}^{n\times d}$: matrix with SVD $\mathbf{X} = \mathbf{U\Sigma V}^T$. Singular values $\sigma_1, \sigma_2, \ldots, \sigma_d$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

Claim 1: When $z^{(0)}$ is chosen with random Gaussian entries, writing $z^{(0)} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \ldots + c_d\mathbf{v}_d$, with very high probability, $\max_j \frac{c_j}{c_1} \leq O(d^2 \log d)$.

Claim 2: For gap $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$, after $t = O\left(\frac{\log(1/\delta)}{\gamma}\right)$ iterations:

$\vec{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d \propto c_1\vec{v}_1 + c_2\delta\vec{v}_2 + \ldots + c_d\delta\vec{v}_d$

If we set $\delta = O\left(\frac{\epsilon}{d^3 \log d}\right)$ by Claim 1 will have:

$\frac{\epsilon}{d^3 \log d} \cdot d^2 \log d = \frac{\epsilon}{d}$

$$\vec{z}^{(t)} \propto \vec{v}_1 + \frac{\epsilon}{d}\left(\vec{v}_2 + \ldots + \vec{v}_d\right).$$

$\mathbf{X} \in \mathbb{R}^{n \times d}$: matrix with SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Singular values $\sigma_1, \sigma_2, \ldots, \sigma_d$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

$c_1 \neq 0$

Claim 1: When $z^{(0)}$ is chosen with random Gaussian entries, writing $z^{(0)} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \ldots + c_d\mathbf{v}_d$, with very high probability, $\max_j \frac{c_j}{c_1} \leq O(d^2 \log d)$.

$\log(1/\delta)$ $\qquad \log\left(\frac{d^3 \log d}{\varepsilon}\right) = O(\log \frac{d}{\varepsilon})$

Claim 2: For gap $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$, after $t = O\left(\frac{\log(1/\delta)}{\gamma}\right)$ iterations:

$$\vec{z}^{(t)} = c_1\cancel{\sigma_1^{2t}}^{\,1}\vec{v}_1 + c_2\cancel{\sigma_2^{2t}}^{\,\delta}\vec{v}_2 + \ldots + c_d\cancel{\sigma_d^{2t}}^{\,\delta}\vec{v}_d \propto c_1\vec{v}_1 + c_2\delta\vec{v}_2 + \ldots + c_d\delta\vec{v}_d$$

If we set $\delta = O\left(\frac{\epsilon}{d^3 \log d}\right)$ by Claim 1 will have:

$$\vec{z}^{(t)} \propto \vec{v}_1 + \frac{\epsilon}{d}\left(\vec{v}_2 + \ldots + \vec{v}_d\right).$$

Gives $\|\vec{z}^{(t)} - \vec{v}_1\|_2 \leq O(\epsilon)$.

$\mathbf{X} \in \mathbb{R}^{n \times d}$: matrix with SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Singular values $\sigma_1, \sigma_2, \ldots, \sigma_d$. $\vec{v}_1$: top right singular vector, being computed, $\vec{z}^{(i)}$: iterate at step i, converging to $\vec{v}_1$.

Theorem (Basic Power Method Convergence)

Let $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$ *be the relative gap between the first and second largest singular values. If Power Method is initialized with a random Gaussian vector* $\vec{v}^{(0)}$ *then, with high probability, after* $t = O\left(\frac{\log d/\epsilon}{\gamma}\right)$ *steps:*

$$\|\vec{z}^{(t)} - \vec{v}_1\|_2 \leq \epsilon.$$

Theorem (Basic Power Method Convergence)

Let $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$ be the relative gap between the first and second largest singular values. If Power Method is initialized with a random Gaussian vector $\vec{v}^{(0)}$ then, with high probability, after $t = O\left(\frac{\log d/\epsilon}{\gamma}\right)$ steps:

$$\|\vec{z}^{(t)} - \vec{v}_1\|_2 \leq \epsilon.$$

Total runtime: $O(t)$ matrix-vector multiplications.

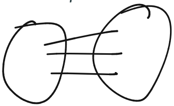
$$O\left(\underline{\mathrm{nnz}(\mathbf{X})} \cdot \frac{\log(d/\epsilon)}{\gamma}\cdot\right) = O\left(nd \cdot \frac{\log(d/\epsilon)}{\gamma}\right). \quad O(n^2)$$

POWER METHOD THEOREM

Theorem (Basic Power Method Convergence)

Let $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$ be the relative gap between the first and second largest singular values. If Power Method is initialized with a random Gaussian vector $\vec{v}^{(0)}$ then, with high probability, after $t = O\left(\frac{\log d/\epsilon}{\gamma}\right)$ steps:

$\|z^{(i)} - z^{(i-1)}\|$



$$\|\vec{z}^{(t)} - \vec{v}_1\|_2 \leq \epsilon.$$

Total runtime: $O(t)$ matrix-vector multiplications.

$\epsilon \to \frac{\epsilon}{\alpha}$

$$O\left(\text{nnz}(\mathbf{X}) \cdot \frac{\log(d/\epsilon)}{\gamma} \cdot\right) = O\left(nd \cdot \frac{\log(d/\epsilon)}{\gamma}\right).$$

$\log(1/\epsilon) \to 1 + \log(1/\epsilon)$

How is ϵ dependence? "linearly convergent"

How is γ dependence?

$\sigma_1 > \sigma_2$

Krylov subspace methods (Lanczos method, Arnoldi method.)

- How `svds`/`eigs` are actually implemented. Only need $t = O\left(\frac{\log d/\epsilon}{\sqrt{\gamma}}\right)$ steps for the same guarantee.

Krylov subspace methods (Lanczos method, Arnoldi method.)

- How `svds`/`eigs` are actually implemented. Only need $t = O\left(\frac{\log d/\epsilon}{\sqrt{\gamma}}\right)$ steps for the same guarantee.

Main Idea: Need to separate σ_1 from σ_i for $i \geq 2$.

Krylov subspace methods (Lanczos method, Arnoldi method.)

- How `svds`/`eigs` are actually implemented. Only need $t = O\left(\frac{\log d/\epsilon}{\sqrt{\gamma}}\right)$ steps for the same guarantee.

Main Idea: Need to separate σ_1 from σ_i for $i \geq 2$.

- Power method: power up to $\sigma_1^{2\cdot t}$ and $\sigma_i^{2\cdot t}$.

Krylov subspace methods (Lanczos method, Arnoldi method.)

- How `svds`/`eigs` are actually implemented. Only need $t = O\left(\frac{\log d/\epsilon}{\sqrt{\gamma}}\right)$ steps for the same guarantee.

Main Idea: Need to separate σ_1 from σ_i for $i \geq 2$.

$p(x) = x^t$

- Power method: power up to $\sigma_1^{2\cdot t}$ and $\sigma_i^{2\cdot t}$. $\rightarrow p(\sigma_1^2) \gg p(\sigma_i^2)$
- Krylov methods: apply a better degree t polynomial $T_t(\sigma_1^2)$ and $T_t(\sigma_i^2)$.

Krylov subspace methods (Lanczos method, Arnoldi method.)

- How `svds`/`eigs` are actually implemented. Only need $t = O\left(\frac{\log d/\epsilon}{\sqrt{\gamma}}\right)$ steps for the same guarantee.

Main Idea: Need to separate σ_1 from σ_i for $i \geq 2$.

- Power method: power up to $\sigma_1^{2\cdot t}$ and $\sigma_i^{2\cdot t}$.
- Krylov methods: apply a better degree t polynomial $T_t(\sigma_1^2)$ and $T_t(\sigma_i^2)$.
- Still requires just $2t$ matrix vector multiplies. Why?

$X^T X v^0$ $\quad (X^T X) v_1 \quad (X^T X) v_2 \quad (a_1 X^T X + a_2 (X^T X)^2 + \dots) z_0$

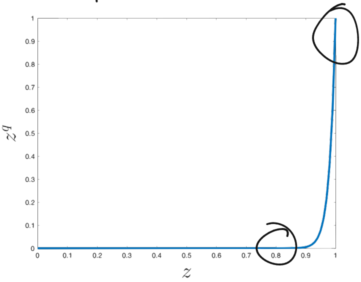
$X^T X v^0 \quad (X^T X)^2 v_0 \quad (X^T X)^3 v_0 \quad X^T X X^T X v_0$

$\qquad$ lynd

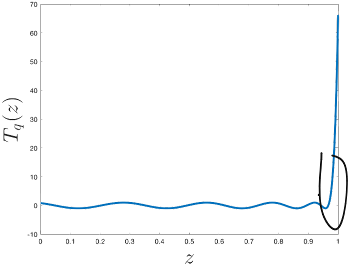
with 2nd

$P(x) = x^t$

$\frac{1}{\gamma} \to \frac{1}{\sqrt{\gamma}}$



vs.



Optimal 'jump' polynomial in general is given by a degree t Chebyshev polynomial. Krylov methods find a polynomial tuned to the input matrix that does at least as well.

- Block Power Method aka Simultaneous Iteration aka Subspace Iteration aka Orthogonal Iteration
- Block Krylov methods

$$\text{Runtime: } O\left(nd\underline{k} \cdot \frac{\log d/\epsilon}{\sqrt{\gamma}}\right)$$

σ_k σ_{k+1}

to accurately compute the top k singular vectors.

- Block Power Method aka Simultaneous Iteration aka Subspace Iteration aka Orthogonal Iteration
- Block Krylov methods

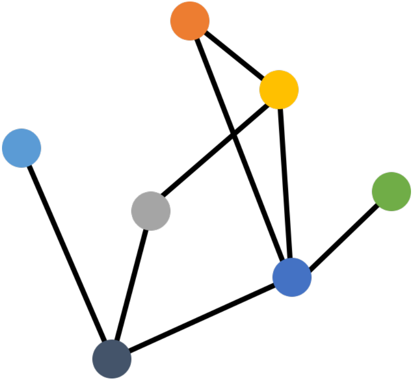
$$\text{Runtime: } O\left(ndk \cdot \frac{\log d/\epsilon}{\sqrt{\gamma}}\right)$$

to accurately compute the top k singular vectors.

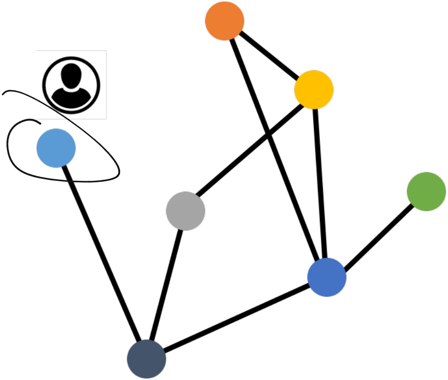
$$\text{'Gapless' Runtime: } O\left(ndk \cdot \frac{\log d/\epsilon}{\sqrt{\epsilon}}\right)$$

if you just want a set of vectors that gives an ϵ-optimal low-rank approximation when you project onto them.

Consider a random walk on a graph G with adjacency matrix $\mathbf{A}$.

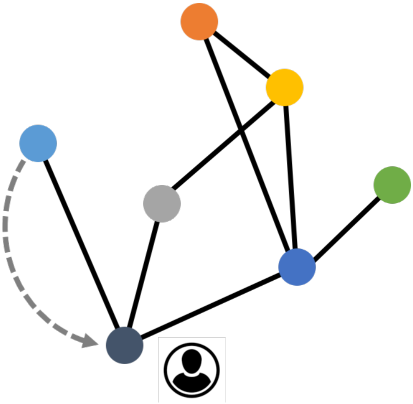


Consider a random walk on a graph G with adjacency matrix $\mathbf{A}$.

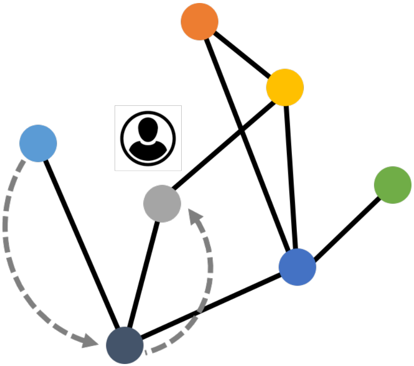


At each step, move to a random vertex, chosen uniformly at random from the neighbors of the current vertex.

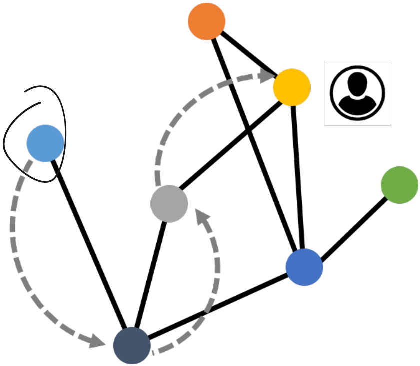
Consider a random walk on a graph G with adjacency matrix $\mathbf{A}$.



Consider a random walk on a graph G with adjacency matrix $\mathbf{A}$.



Consider a random walk on a graph G with adjacency matrix $\mathbf{A}$.



Let $\vec{p}^{(t)} \in \mathbb{R}^n$ have i^{th} entry $\vec{p}_i^{(t)} = \Pr(\text{walk at node i at step t})$.

Let $\vec{p}^{(t)} \in \mathbb{R}^n$ have i^{th} entry $\vec{p}_i^{(t)} = \Pr(\text{walk at node i at step t})$.

- **Initialize:** $\vec{p}^{(0)} = [1, 0, 0, \ldots, 0]$.

Let $\vec{p}^{(t)} \in \mathbb{R}^n$ have i^{th} entry $\vec{p}_i^{(t)} = \Pr(\text{walk at node i at step t})$.

- **Initialize:** $\vec{p}^{(0)} = [1, 0, 0, \ldots, 0]$.
- **Update:**

$$\Pr(\text{walk at i at step t}) = \sum_{j \in neigh(i)} \Pr(\text{walk at j at step t-1}) \cdot \frac{1}{degree(j)}$$

$z_j \; p_i^{(t-1)}$

where $\vec{z}_j = \frac{1}{degree(j)}$ for all $j \in neigh(i)$.

CONNECTION TO RANDOM WALKS

Let $\vec{p}^{(t)} \in \mathbb{R}^n$ have i^{th} entry $\vec{p}_i^{(t)} = \Pr(\text{walk at node i at step t})$.

- **Initialize:** $\vec{p}^{(0)} = [1, 0, 0, \ldots, 0]$.
- **Update:**

$$\begin{aligned}\Pr(\text{walk at i at step t}) &= \sum_{j \in neigh(i)} \Pr(\text{walk at j at step t-1}) \cdot \frac{1}{degree(j)} \\ &= \vec{z}^T \vec{p}^{(t-1)}\end{aligned}$$

where $\vec{z}_j = \frac{1}{degree(j)}$ for all $j \in neigh(i)$.

Let $\vec{p}^{(t)} \in \mathbb{R}^n$ have i^{th} entry $\vec{p}_i^{(t)} = \Pr(\text{walk at node i at step t})$.

- **Initialize:** $\vec{p}^{(0)} = [1, 0, 0, \ldots, 0]$.
- **Update:**

$$\Pr(\text{walk at i at step t}) = \sum_{j \in neigh(i)} \Pr(\text{walk at j at step t-1}) \cdot \frac{1}{degree(j)}$$

where $\vec{z}_j = \frac{1}{degree(j)}$ for all $j \in neigh(i)$.

- $\vec{z}$ is just the i^{th} row of the right normalized adjacency matrix $\mathbf{A}\mathbf{D}^{-1}$.

$$p^{(t)} = AD^{-1} p^{(t-1)}$$

$$p^{(t)} = \underbrace{AD^{-1} AD^{-1} \cdots}_{t \text{ times}} \; p^{(0)}$$

CONNECTION TO RANDOM WALKS

Let $\vec{p}^{(t)} \in \mathbb{R}^n$ have i^{th} entry $\vec{p}_i^{(t)} = \Pr(\text{walk at node i at step t})$.

- **Initialize:** $\vec{p}^{(0)} = [1, 0, 0, \ldots, 0]$.
- **Update:**

$$\Pr(\text{walk at i at step t}) = \sum_{j \in neigh(i)} \Pr(\text{walk at j at step t-1}) \cdot \frac{1}{degree(j)}$$

where $\vec{z}_j = \frac{1}{degree(j)}$ for all $j \in neigh(i)$.

- $\vec{z}$ is just the i^{th} row of the right normalized adjacency matrix $\mathbf{A}\mathbf{D}^{-1}$.
- $\vec{p}^{(t)} = \mathbf{A}\mathbf{D}^{-1}\vec{p}^{(t-1)}$

Let $\vec{p}^{(t)} \in \mathbb{R}^n$ have i^{th} entry $\vec{p}_i^{(t)} = \Pr(\text{walk at node i at step t})$.

- **Initialize:** $\vec{p}^{(0)} = [1, 0, 0, \ldots, 0]$.
- **Update:**

$$\Pr(\text{walk at i at step t}) = \sum_{j \in neigh(i)} \Pr(\text{walk at j at step t-1}) \cdot \frac{1}{degree(j)}$$

where $\vec{z}_j = \frac{1}{degree(j)}$ for all $j \in neigh(i)$.

- $\vec{z}$ is just the i^{th} row of the right normalized adjacency matrix $\mathbf{AD}^{-1}$.
- $\vec{p}^{(t)} = \mathbf{AD}^{-1}\vec{p}^{(t-1)} = \underbrace{\mathbf{AD}^{-1}\mathbf{AD}^{-1}\ldots\mathbf{AD}^{-1}}_{t \text{ times}}\vec{p}^{(0)}$

Claim: After t steps, the probability that a random walk is at node i is given by the i^{th} entry of

$$\vec{p}^{(t)} = \underbrace{\mathbf{A}\mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1}\ldots\mathbf{A}\mathbf{D}^{-1}}_{t \text{ times}}\vec{p}^{(0)}.$$

Claim: After t steps, the probability that a random walk is at node i is given by the i^{th} entry of

$$\vec{p}^{(t)} = \underbrace{\mathbf{A}\mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1}\ldots\mathbf{A}\mathbf{D}^{-1}}_{t \text{ times}}\vec{p}^{(0)}.$$

$$\mathbf{D}^{-1/2}\vec{p}^{(t)} = \underbrace{(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})\ldots(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})}_{t \text{ times}}(D^{-1/2}\vec{p}^{(0)}).$$

RANDOM WALKING AS POWER METHOD

Claim: After t steps, the probability that a random walk is at node i is given by the i^{th} entry of

$$\vec{p}^{(t)} = \underbrace{\mathbf{A}\mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1}\ldots\mathbf{A}\mathbf{D}^{-1}}_{t \text{ times}}\vec{p}^{(0)}.$$

$$\mathbf{D}^{-1/2}\vec{p}^{(t)} = \underbrace{(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})\ldots(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})}_{t \text{ times}}(D^{-1/2}\vec{p}^{(0)}).$$

- $\mathbf{D}^{-1/2}\vec{p}^{(t)}$ is exactly what would obtained by applying $t/2$ iterations of power method to $D^{-1/2}\vec{p}^{(0)}$!

Claim: After t steps, the probability that a random walk is at node i is given by the i^{th} entry of

$$\vec{p}^{(t)} = \underbrace{\mathbf{AD}^{-1}\mathbf{AD}^{-1}\ldots\mathbf{AD}^{-1}}_{t \text{ times}}\vec{p}^{(0)}.$$

$$\mathbf{D}^{-1/2}\vec{p}^{(t)} = \underbrace{(\mathbf{D}^{-1/2}\mathbf{AD}^{-1/2})(\mathbf{D}^{-1/2}\mathbf{AD}^{-1/2})\ldots(\mathbf{D}^{-1/2}\mathbf{AD}^{-1/2})}_{t \text{ times}}(D^{-1/2}\vec{p}^{(0)}).$$

- $\mathbf{D}^{-1/2}\vec{p}^{(t)}$ is exactly what would obtained by applying $t/2$ iterations of power method to $D^{-1/2}\vec{p}^{(0)}$!
- Will converge to the top singular vector (eigenvector) of the normalized adjacency matrix $\mathbf{D}^{-1/2}\mathbf{AD}^{-1/2}$. Stationary distribution.

RANDOM WALKING AS POWER METHOD

Claim: After t steps, the probability that a random walk is at node i is given by the i^{th} entry of

$$\vec{p}^{(t)} = \underbrace{\mathbf{A}\mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1}\ldots\mathbf{A}\mathbf{D}^{-1}}_{t \text{ times}}\vec{p}^{(0)}.$$

$$\mathbf{D}^{-1/2}\vec{p}^{(t)} = \underbrace{(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})\ldots(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})}_{t \text{ times}}(D^{-1/2}\vec{p}^{(0)}).$$

- $\mathbf{D}^{-1/2}\vec{p}^{(t)}$ is exactly what would obtained by applying $t/2$ iterations of power method to $D^{-1/2}\vec{p}^{(0)}$!
- Will converge to the top singular vector (eigenvector) of the normalized adjacency matrix $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$. Stationary distribution.
- Like the power method, the time a random walk takes to converge to its stationary distribution (mixing time) is dependent on the gap between the top two eigenvalues of $\mathbf{A}\mathbf{D}^{-1}$. The spectral gap.