COMPSIC 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.
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 $X \in \mathbb{R}^{n \times d}$ for $k \ll d$.

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

$O(nd^2)$
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 $X \in \mathbb{R}^{n \times k}$ for $k \ll d$.

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

Sparse (iterative) vs. Direct Method. \texttt{svd} vs. \texttt{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:** 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 $X \in \mathbb{R}^{n \times d}$, with SVD $X = U\Sigma V^\top$, find $\tilde{z} \approx \tilde{v}_1$. 


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

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

Return $\tilde{Z}_t$
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 \( X \in \mathbb{R}^{n \times d} \), with SVD \( X = U \Sigma V \), find \( \tilde{z} \approx v_1 \).

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

Return \( \tilde{z}_t \)

**Total Runtime:** \( O(ndt) \sim O(nd^2) \)

\[ X^T X = O(nd^2) \]
POWER METHOD

- $\vec{v}_1$ is the top singular vector.
- $\vec{Z}(0)$
- Unit circle
POWER METHOD

unit circle
POWER METHOD

![Diagram of eigenvalues and eigenvectors](image)

- $\mathbf{v}_1$
- $\mathbf{z}^{(0)}$
- $\mathbf{z}^{(1)}$
- $\mathbf{z}^{(2)}$

unit circle
POWER METHOD

unit circle
POWER METHOD

\[ \vec{v}_1, \vec{z}^{(1)}, \vec{z}^{(2)}, \vec{z}^{(3)} \]

unit circle
POWER METHOD

Why is it converging towards \( \vec{v}_1 \)?
Write $\tilde{z}^{(0)}$ in the right singular vector basis:

$$\tilde{z}^{(0)} = c_1\tilde{v}_1 + c_2\tilde{v}_2 + \ldots + c_d\tilde{v}_d.$$
Write $\tilde{z}^{(0)}$ in the right singular vector basis:

$$X = \mathcal{U} \mathcal{E} \mathcal{V}^T$$

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

Update step: $\tilde{z}^{(i)} = X \tilde{x}^0 \cdot \tilde{z}^{(i-1)} = \mathbf{V} \mathbf{\Sigma}^2 \mathbf{V}^T \cdot \tilde{z}^{(i-1)}$ (then normalize)
Write $\mathbf{z}^{(0)}$ in the right singular vector basis:

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

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

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

$$\sqrt{\mathbf{V}^T \mathbf{z}^{(0)}} = \sqrt{\mathbf{V}^T (c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 \ldots)}$$

$$\sqrt{c_1 \mathbf{v}_1} = c_1 \mathbf{v}_1$$

$$\frac{1}{\sqrt{c_1}} \begin{bmatrix} 1 \\ 0 \\ \frac{1}{\sqrt{c_2}} \end{bmatrix}$$

$\mathbf{X} \in \mathbb{R}^{n \times d}$: input matrix with SVD $\mathbf{X} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T$. $\mathbf{v}_1$: top right singular vector, being computed, $\mathbf{z}^{(i)}$: iterate at step $i$, converging to $\mathbf{v}_1$. 
Write $\bar{z}^{(0)}$ in the right singular vector basis:

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

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

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

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

$\mathbf{X} \in \mathbb{R}^{n \times d}$: input matrix with SVD $\mathbf{X} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$. $\vec{v}_1$: top right singular vector, being computed, $\bar{z}^{(i)}$: iterate at step $i$, converging to $\vec{v}_1$. 
Write $\tilde{z}^{(0)}$ in the right singular vector basis:

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

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

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

$$
\Sigma^2 V^T \tilde{z}^{(0)} =
$$

$$
\tilde{z}^{(1)} = V (\Sigma^2 V^T \cdot \tilde{z}^{(0)}) = V \begin{bmatrix}
0 \\
\vdots \\
0 \\
g_1^2 c_1 \\
\vdots \\
g_d^2 c_d
\end{bmatrix} = g_1^2 c_1 \tilde{v}_1 + g_2^2 c_2 \tilde{v}_2 + \ldots + g_d^2 c_d \tilde{v}_d
$$

$X \in \mathbb{R}^{n \times d}$: input matrix with SVD $X = U \Sigma V^T$. $\tilde{v}_1$: top right singular vector, being computed, $\tilde{z}^{(i)}$: iterate at step $i$, converging to $\tilde{v}_1$. 
Claim 1: Writing \( \vec{z}^{(0)} = c_1 \vec{v}_1 + c_2 \vec{v}_2 + \ldots + c_d \vec{v}_d \),

\[
\vec{z}^{(1)} = c_1 \cdot \sigma_1^2 \vec{v}_1 + c_2 \cdot \sigma_2^2 \vec{v}_2 + \ldots + c_d \cdot \sigma_d^2 \vec{v}_d.
\]
Claim 1: Writing $\tilde{z}^{(0)} = c_1 \tilde{v}_1 + c_2 \tilde{v}_2 + \ldots + c_d \tilde{v}_d$,

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

$$\tilde{z}^{(2)} = X^T X \tilde{z}^{(1)} = V \Sigma^2 V^T \tilde{z}^{(1)} = c_1 \cdot \gamma_1 \tilde{v}_1 + \ldots + c_d \cdot \gamma_d \tilde{v}_d$$

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

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

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

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

Claim 2:

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

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

$$z^{(0)} = c_1 \vec{v}_1 + c_2 \vec{v}_2 + \ldots + c_d \vec{v}_d \implies z^{(t)} = c_1 \sigma_1^{2t} \vec{v}_1 + c_2 \sigma_2^{2t} \vec{v}_2 + \ldots + c_d \sigma_d^{2t} \vec{v}_d$$
After $t$ iterations, we have ‘powered’ up the singular values, making the component in the direction of $\nu_1$ much larger, relative to the other components.

$$\mathbf{z}^{(0)} = c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + \ldots + c_d \mathbf{v}_d \quad \Rightarrow \quad \mathbf{z}^{(t)} = c_1 \sigma_1^{2t} \mathbf{v}_1 + c_2 \sigma_2^{2t} \mathbf{v}_2 + \ldots + c_d \sigma_d^{2t} \mathbf{v}_d$$
After $t$ iterations, we have ‘powered’ up the singular values, making the component in the direction of $\nu_1$ much larger, relative to the other components.

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

$$\vec{z}^{(0)} = c_1 \vec{v}_1 + c_2 \vec{v}_2 + \ldots + c_d \vec{v}_d \quad \Rightarrow \quad \vec{z}^{(t)} = c_1 \sigma_1^{2t} \vec{v}_1 + c_2 \sigma_2^{2t} \vec{v}_2 + \ldots + c_d \sigma_d^{2t} \vec{v}_d$$
After $t$ iterations, we have ‘powered’ up the singular values, making the component in the direction of $\nu_1$ much larger, relative to the other components.

\[
\mathbf{z}^{(0)} = c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + \ldots + c_d \mathbf{v}_d \quad \Rightarrow \quad \mathbf{z}^{(t)} = c_1 \sigma_1^{2t} \mathbf{v}_1 + c_2 \sigma_2^{2t} \mathbf{v}_2 + \ldots + c_d \sigma_d^{2t} \mathbf{v}_d
\]
After $t$ iterations, we have ‘powered’ up the singular values, making the component in the direction of $\nu_1$ much larger, relative to the other components.

$$\tilde{z}^{(0)} = c_1 \tilde{v}_1 + c_2 \tilde{v}_2 + \ldots + c_d \tilde{v}_d \implies \tilde{z}^{(t)} = c_1 \sigma_1^{2t} \tilde{v}_1 + c_2 \sigma_2^{2t} \tilde{v}_2 + \ldots + c_d \sigma_d^{2t} \tilde{v}_d$$
After \( t \) iterations, we have ‘powered’ up the singular values, making the component in the direction of \( \nu_1 \) much larger, relative to the other components.

\[
\begin{align*}
\mathbf{z}^{(0)} &= c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + \ldots + c_d \mathbf{v}_d \\
\implies \mathbf{z}^{(t)} &= c_1 \sigma_1^{2t} \mathbf{v}_1 + c_2 \sigma_2^{2t} \mathbf{v}_2 + \ldots + c_d \sigma_d^{2t} \mathbf{v}_d
\end{align*}
\]
After $t$ iterations, we have ‘powered’ up the singular values, making the component in the direction of $\nu_1$ much larger, relative to the other components.

$$\tilde{z}^{(0)} = c_1 \vec{v}_1 + c_2 \vec{v}_2 + \ldots + c_d \vec{v}_d \quad \Rightarrow \quad \tilde{z}^{(t)} = c_1 \sigma_1^{2t} \vec{v}_1 + c_2 \sigma_2^{2t} \vec{v}_2 + \ldots + c_d \sigma_d^{2t} \vec{v}_d$$
After $t$ iterations, we have ‘powered’ up the singular values, making the component in the direction of $\nu_1$ much larger, relative to the other components.

\[ \mathbf{z}^{(0)} = c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + \ldots + c_d \mathbf{v}_d \quad \Rightarrow \quad \mathbf{z}^{(t)} = c_1 \sigma_1^{2t} \mathbf{v}_1 + c_2 \sigma_2^{2t} \mathbf{v}_2 + \ldots + c_d \sigma_d^{2t} \mathbf{v}_d \]
After \( t \) iterations, we have ‘powered’ up the singular values, making the component in the direction of \( \nu_1 \) much larger, relative to the other components.

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

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

\[ \mathbf{z}^{(0)} = c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + \ldots + c_d \mathbf{v}_d \implies \mathbf{z}^{(t)} = c_1 \sigma_1^{2t} \mathbf{v}_1 + c_2 \sigma_2^{2t} \mathbf{v}_2 + \ldots + c_d \sigma_d^{2t} \mathbf{v}_d \]
After $t$ iterations, we have ‘powered’ up the singular values, making the component in the direction of $\nu_1$ much larger, relative to the other components.

$$\bar{z}^{(0)} = c_1 \vec{v}_1 + c_2 \vec{v}_2 + \ldots + c_d \vec{v}_d \quad \Rightarrow \quad \bar{z}^{(t)} = c_1 \sigma_1^{2t} \vec{v}_1 + c_2 \sigma_2^{2t} \vec{v}_2 + \ldots + c_d \sigma_d^{2t} \vec{v}_d$$
After $t$ iterations, we have ‘powered’ up the singular values, making the component in the direction of $\nu_1$ much larger, relative to the other components.

$$z^{(0)} = c_1 \nu_1 + c_2 \nu_2 + \ldots + c_d \nu_d \quad \Rightarrow \quad z^{(t)} = c_1 \sigma_1^{2t} \nu_1 + c_2 \sigma_2^{2t} \nu_2 + \ldots + c_d \sigma_d^{2t} \nu_d$$

### POWER METHOD CONVERGENCE

#### Iteration 12

<table>
<thead>
<tr>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
<th>$c_5$</th>
<th>$c_6$</th>
<th>$c_7$</th>
<th>$c_8$</th>
<th>$c_9$</th>
<th>$c_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
After \( t \) iterations, we have ‘powered’ up the singular values, making the component in the direction of \( \nu_1 \) much larger, relative to the other components.

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

When will convergence be slow?
Slow Case: X has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

$$\tilde{Z}^{(0)} = c_1 \tilde{v}_1 + c_2 \tilde{v}_2 + \ldots + c_d \tilde{v}_d \implies \tilde{Z}^{(t)} = c_1 \sigma_1^{2^t} \tilde{v}_1 + c_2 \sigma_2^{2^t} \tilde{v}_2 + \ldots + c_d \sigma_d^{2^t} \tilde{v}_d$$
Slow Case: $X$ has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

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

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

\[
\mathbf{z}^{(0)} = c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + \ldots + c_d \mathbf{v}_d \iff \mathbf{z}^{(t)} = c_1 \sigma_1^{2t} \mathbf{v}_1 + c_2 \sigma_2^{2t} \mathbf{v}_2 + \ldots + c_d \sigma_d^{2t} \mathbf{v}_d
\]
Slow Case: X has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

$$\mathbf{Z}^{(0)} = c_1 \mathbf{V}_1 + c_2 \mathbf{V}_2 + \ldots + c_d \mathbf{V}_d \implies \mathbf{Z}^{(t)} = c_1 \sigma_1^{2t} \mathbf{V}_1 + c_2 \sigma_2^{2t} \mathbf{V}_2 + \ldots + c_d \sigma_d^{2t} \mathbf{V}_d$$
**POWER METHOD SLOW CONVERGENCE**

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

$$\tilde{z}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \ldots + c_d\vec{v}_d \implies \tilde{z}^{(t)} = c_1\sigma_1^{2t}\vec{v}_1 + c_2\sigma_2^{2t}\vec{v}_2 + \ldots + c_d\sigma_d^{2t}\vec{v}_d$$
Slow Case: $X$ has singular values: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

$$\mathbf{Z}^{(0)} = c_1 \mathbf{V}_1 + c_2 \mathbf{V}_2 + \ldots + c_d \mathbf{V}_d \implies \mathbf{Z}^{(t)} = c_1 \sigma_1^{2t} \mathbf{V}_1 + c_2 \sigma_2^{2t} \mathbf{V}_2 + \ldots + c_d \sigma_d^{2t} \mathbf{V}_d$$
**POWER METHOD SLOW CONVERGENCE**

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

\[
\vec{x}^{(0)} = c_1 \vec{v}_1 + c_2 \vec{v}_2 + \ldots + c_d \vec{v}_d \implies \vec{x}^{(t)} = c_1 \sigma_1^{2t} \vec{v}_1 + c_2 \sigma_2^{2t} \vec{v}_2 + \ldots + c_d \sigma_d^{2t} \vec{v}_d
\]
**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: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

$$\tilde{z}^{(0)} = c_1 \vec{v}_1 + c_2 \vec{v}_2 + \ldots + c_d \vec{v}_d \implies \tilde{z}^{(t)} = c_1 \sigma_1^{2t} \vec{v}_1 + c_2 \sigma_2^{2t} \vec{v}_2 + \ldots + c_d \sigma_d^{2t} \vec{v}_d$$
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: \( \sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots \)

\[
\vec{z}^{(0)} = c_1 \vec{v}_1 + c_2 \vec{v}_2 + \ldots + c_d \vec{v}_d \quad \Rightarrow \quad \vec{z}^{(t)} = c_1 \sigma_1^{2t} \vec{v}_1 + c_2 \sigma_2^{2t} \vec{v}_2 + \ldots + c_d \sigma_d^{2t} \vec{v}_d
\]
**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: $\sigma_1 = 1, \sigma_2 = .99, \sigma_3 = .9, \sigma_4 = .8, \ldots$

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

$\bar{z}^{t} \neq \vec{v}_1$
Rule of 70

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

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 \( \sigma_2^{2t} \leq \frac{1}{2} \cdot \sigma_1^{2t} \)?

\[ \sigma_2^{2t} \leq \frac{1}{2} \sigma_1^{2t} \Rightarrow \left( \frac{\sigma_2}{\sigma_1} \right)^{2t} \leq \frac{1}{2} \]

\[ (1 - \gamma)^{2t} \leq \frac{1}{2} \]

\[ (1 - \gamma)^{1/2} = \frac{1}{e} \]

\[ t = O(1/\gamma) \implies (1 - \gamma)^{2t} \leq \frac{1}{2} \]

\[ X \in \mathbb{R}^{n \times d} : \text{matrix with SVD} \ X = U \Sigma V^T. \text{ Singular values } \sigma_1, \sigma_2, \ldots, \sigma_d. \ \vec{v}_1: \text{top right singular vector, being computed, } z^{(i)}: \text{iterate at step } i, \text{ converging to } \vec{v}_1. \]
\[ \tilde{z}^{(0)} = c_1 \tilde{v}_1 + c_2 \tilde{v}_2 + \ldots + c_d \tilde{v}_d \implies \tilde{z}^{(t)} = c_1 \sigma_1^{2t} \tilde{v}_1 + c_2 \sigma_2^{2t} \tilde{v}_2 + \ldots + c_d \sigma_d^{2t} \tilde{v}_d \]

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 \( \sigma_2^{2t} \leq \frac{1}{2} \cdot \sigma_1^{2t} \)? \( \mathcal{O}(1/\gamma) \cdot \mathcal{O}(\log_{1/\gamma} 1/2) \)

\[ X \in \mathbb{R}^{n \times d} : \text{matrix with SVD } X = U \Sigma V^T. \text{ Singular values } \sigma_1, \sigma_2, \ldots, \sigma_d. \text{ } \tilde{v}_1 : \text{top right singular vector}, \text{being computed}, \tilde{z}^{(i)} : \text{iterate at step } i, \text{converging to } \tilde{v}_1. \]
\[ \bar{z}^{(0)} = c_1 \bar{v}_1 + c_2 \bar{v}_2 + \ldots + c_d \bar{v}_d \implies \bar{z}^{(t)} = c_1 \sigma_1^{2t} \bar{v}_1 + c_2 \sigma_2^{2t} \bar{v}_2 + \ldots + c_d \sigma_d^{2t} \bar{v}_d \]

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 \( \sigma_2^{2t} \leq \delta \cdot \sigma_1^{2t} \)?

\[
\begin{align*}
O(1/\gamma) & \quad 2^t \leq \frac{1}{\delta} \cdot 6 \cdot 6^t \\
O(1/\gamma) \cdot O(\log(1/\delta)) & \quad m = \log \left( \frac{1}{1/d} \right) \\
\end{align*}
\]

\[ X \in \mathbb{R}^{n \times d} : \text{matrix with SVD } X = U \Sigma V^T. \text{ Singular values } \sigma_1, \sigma_2, \ldots, \sigma_d. \text{ } \bar{v}_1: \text{top right singular vector, being computed, } \bar{z}^{(i)}: \text{iterate at step } i, \text{converging to } \bar{v}_1. \]
\[ \tilde{Z}^{(0)} = c_1 \tilde{v}_1 + c_2 \tilde{v}_2 + \ldots + c_d \tilde{v}_d \implies \tilde{Z}^{(t)} = c_1 \sigma_1^{2t} \tilde{v}_1 + c_2 \sigma_2^{2t} \tilde{v}_2 + \ldots + c_d \sigma_d^{2t} \tilde{v}_d \]

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 \( \sigma_2^{2t} \leq \delta \cdot \sigma_1^{2t} \)? \( O \left( \frac{\log(1/\delta)}{\gamma} \right) \).

**X \in \mathbb{R}^{n \times d}**: matrix with SVD \( X = U \Sigma V^T \). Singular values \( \sigma_1, \sigma_2, \ldots, \sigma_d \). \( \tilde{v}_1 \): top right singular vector, being computed, \( \tilde{z}^{(i)} \): iterate at step \( i \), converging to \( \tilde{v}_1 \). 
\( \mathbf{z}(0) = c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + \ldots + c_d \mathbf{v}_d \implies \mathbf{z}(t) = c_1 \sigma_1^{2t} \mathbf{v}_1 + c_2 \sigma_2^{2t} \mathbf{v}_2 + \ldots + c_d \sigma_d^{2t} \mathbf{v}_d \)

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 \( \sigma_2^{2t} \leq \frac{1}{2} \cdot \sigma_1^{2t} \)? \( \mathcal{O}(1/\gamma) \).

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

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

\[ X \in \mathbb{R}^{n \times d} \]: matrix with SVD \( X = U \Sigma V^T \). Singular values \( \sigma_1, \sigma_2, \ldots, \sigma_d \). \( \mathbf{v}_1 \): top right singular vector, being computed, \( \mathbf{z}^{(i)} \): iterate at step \( i \), converging to \( \mathbf{v}_1 \).
**Claim:** When $z^{(0)}$ is chosen with random Gaussian entries, writing $z^{(0)} = c_1 v_1 + c_2 v_2 + \ldots + c_d v_d$, with very high probability, for all $i$:

$$0(1/d^2) \leq c_i \leq O(\log d)$$

How is $c_i$ distributed?

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

$$c_i = \langle v_i, z^{(0)} \rangle$$

$$c_i \sim N(0, 1)$$

$$c_i \sim N(0, \sigma_i^2)$$

Rotation invariance of Gaussian

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

11
**Claim:** When $z^{(0)}$ is chosen with random Gaussian entries, writing $z^{(0)} = c_1 v_1 + c_2 v_2 + \ldots + c_d v_d$, with very high probability, for all $i$:

$$\log d \geq 1 \quad O(1/d^2) \leq |c_i| \leq O(\log d)$$

**Corollary:**

$$\max_j \left| \frac{c_j}{c_1} \right| \leq O(d^2 \log d), \quad \left| \frac{c_i}{c_1} \right| \leq \frac{\log(d)}{c_1} \leq \frac{\log(d)}{1/d^2} \leq d^2 \log d$$

$X \in \mathbb{R}^{n \times d}$: matrix with SVD $X = U \Sigma V^T$. Singular values $\sigma_1, \sigma_2, \ldots, \sigma_d$. $v_1$: top right singular vector, being computed, $z^{(i)}$: iterate at step $i$, converging to $v_1$. 
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 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:

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

**X**: matrix with SVD $X = U \Sigma V^T$. Singular values $\sigma_1, \sigma_2, \ldots, \sigma_d$. $\tilde{v}_1$: top right singular vector, being computed, $\tilde{x}^{(i)}$: iterate at step $i$, converging to $\tilde{v}_1$. 


**RANDOM INITIALIZATION**

**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:  
\[
\mathbf{z}^{(t)} = c_1 \sigma_1^{2t} \tilde{\mathbf{v}}_1 + c_2 \sigma_2^{2t} \tilde{\mathbf{v}}_2 + \ldots + c_d \sigma_d^{2t} \tilde{\mathbf{v}}_d \propto c_1 \tilde{\mathbf{v}}_1 + c_2 \delta \tilde{\mathbf{v}}_2 + \ldots + c_d \delta \tilde{\mathbf{v}}_d
\]

If we set \( \delta = O \left( \frac{\epsilon}{d^3 \log d} \right) \) by Claim 1 will have:  
\[
\mathbf{z}^{(t)} \propto \tilde{\mathbf{v}}_1 + \frac{\epsilon}{d} (\tilde{\mathbf{v}}_2 + \ldots + \tilde{\mathbf{v}}_d).
\]

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

\[ c_1 = 0 \]

**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, 
\[ \max_j \frac{c_j}{c_1} \leq O(d^2 \log d). \]

**Claim 2:** For gap \( \gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1} \), after \( t = O\left(\frac{\log(1/\delta)}{\gamma}\right) \) iterations:
\[ \tilde{z}^{(t)} = c_1 \sigma_1^2 \mathbf{v}_1 + c_2 \sigma_2^2 \mathbf{v}_2 + \ldots + c_d \sigma_d^2 \mathbf{v}_d \propto c_1 \mathbf{v}_1 + c_2 \delta \mathbf{v}_2 + \ldots + c_d \delta \mathbf{v}_d \]
If we set \( \delta = O\left(\frac{\epsilon}{d^3 \log d}\right) \) by Claim 1 will have:
\[ \tilde{z}^{(t)} \propto \mathbf{v}_1 + \frac{\epsilon}{d} (\mathbf{v}_2 + \ldots + \mathbf{v}_d) \]

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

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

$$O\left(\text{nnz}(X) \cdot \frac{\log(d/\epsilon)}{\gamma}\right) = O\left(nd \cdot \frac{\log(d/\epsilon)}{\gamma}\right). \quad \Theta(n^2)$$
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 $\tilde{v}^{(0)}$ then, with high probability, after $t = O \left( \frac{\log d/\epsilon}{\gamma} \right)$ steps:

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

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

$O \left( \text{nnz}(X) \cdot \frac{\log(d/\epsilon)}{\gamma} \right) = O \left( nd \cdot \frac{\log(d/\epsilon)}{\gamma} \right) \cdot \frac{\epsilon}{\epsilon + \log(1/\epsilon)} \cdot \log(1/\epsilon) + \log(1/\epsilon)$

How is $\epsilon$ dependence?

"linearly convergent"

How is $\gamma$ 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 $\sigma_1$ from $\sigma_i$ 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 $\sigma_1$ from $\sigma_i$ for $i \geq 2$.

- Power method: power up to $\sigma_1^{2t}$ and $\sigma_i^{2t}$. 
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 $\sigma_1$ from $\sigma_i$ 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 $T_t(\sigma_i^2)$. 
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 \( \sigma_1 \) from \( \sigma_i \) 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?

\[
\begin{align*}
X^T X v_0 & \quad (X^T X) v_1 & \quad (X^T X)^2 v_2 \\
X^T X v_0 & \quad (X^T X)^2 v_0 & \quad (X^T X)^3 v_0 \\
\end{align*}
\]
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.
GENERALIZATIONS TO LARGER $k$

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

**Runtime:** $O\left( n dk \cdot \frac{\log d/\epsilon}{\sqrt{\gamma}} \right)$

6_k 6_{k+1}

to accurately compute the top $k$ singular vectors.
GENERALIZATIONS TO LARGER $k$

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

**Runtime:** $O\left(n dk \cdot \frac{\log d/\epsilon}{\sqrt{\gamma}}\right)$

to accurately compute the top $k$ singular vectors.

**‘Gapless’ Runtime:** $O\left(n dk \cdot \frac{\log d/\epsilon}{\sqrt{\epsilon}}\right)$

if you just want a set of vectors that gives an $\epsilon$-optimal low-rank approximation when you project onto them.
Consider a random walk on a graph $G$ with adjacency matrix $A$. 
Consider a random walk on a graph $G$ with adjacency matrix $A$.

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 $A$. 
Consider a random walk on a graph $G$ with adjacency matrix $A$. 
Consider a random walk on a graph $G$ with adjacency matrix $A$. 
Let $\vec{p}^{(t)} \in \mathbb{R}^n$ have $i^{th}$ entry $\vec{p}_i^{(t)} = \Pr(\text{walk at node } i \text{ at step } t)$. 
Let $\vec{p}^{(t)} \in \mathbb{R}^n$ have $i^{th}$ entry $p_i^{(t)} = \Pr(\text{walk at node } i \text{ at step } t)$.

- **Initialize:** $\vec{p}^{(0)} = [1, 0, 0, \ldots, 0]$. 
Let $\vec{p}^{(t)} \in \mathbb{R}^n$ have $i^{th}$ entry $\vec{p}^{(t)}_i = \Pr(\text{walk at node } i \text{ at step } t)$.

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

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

$$
\exists_j \Pr_i^{(t-1)}
$$

where $\exists_j = \frac{1}{\text{degree}(j)}$ for all $j \in \text{neigh}(i)$.
Let $\tilde{p}^{(t)} \in \mathbb{R}^n$ have $i^{th}$ entry $\tilde{p}^{(t)}_i = \Pr(\text{walk at node } i \text{ at step } t)$.

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

- **Update:**

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

$$= \tilde{Z}^T \tilde{p}^{(t-1)}$$

where $\tilde{Z}_j = \frac{1}{\text{degree}(j)}$ for all $j \in \text{neigh}(i)$. 
Let $\mathbf{p}^{(t)} \in \mathbb{R}^n$ have $i^{th}$ entry $p_i^{(t)} = \text{Pr}(\text{walk at node } i \text{ at step } t)$.

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

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

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

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

\[
\begin{align*}
\mathbf{p}^{(t)} &= \mathbf{AD}^{-1} \mathbf{p}^{(t-1)} \\
\mathbf{p}^{(t)} &= \lim_{t \to \infty} \mathbf{AD}^{-1} \mathbf{p}^{(0)}
\end{align*}
\]
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:**

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

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

- \( \vec{z} \) is just the \( i^{th} \) row of the right normalized adjacency matrix \( A \mathbf{D}^{-1} \).
- \( \vec{p}^{(t)} \) = \( A \mathbf{D}^{-1} \vec{p}^{(t-1)} \)
Let $\vec{p}^{(t)} \in \mathbb{R}^n$ have $i^{th}$ entry $p_i^{(t)} = \Pr(\text{walk at node } i \text{ at step } t)$.

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

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

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

- $\vec{z}$ is just the $i^{th}$ row of the right normalized adjacency matrix $\mathbf{A} \mathbf{D}^{-1}$.
- $\vec{p}^{(t)} = \mathbf{A} \mathbf{D}^{-1} \vec{p}^{(t-1)} = \underbrace{\mathbf{A} \mathbf{D}^{-1} \mathbf{A} \mathbf{D}^{-1} \ldots \mathbf{A} \mathbf{D}^{-1}}_{t \text{ times}} \vec{p}^{(0)}$
Claim: After $t$ steps, the probability that a random walk is at node $i$ is given by the $i^{th}$ entry of

$$
\bar{p}(t) = \underbrace{AD^{-1}AD^{-1} \ldots AD^{-1}}_{t \text{ times}} \bar{p}(0).
$$
**Claim:** After $t$ steps, the probability that a random walk is at node $i$ is given by the $i^{th}$ entry of

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

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

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

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

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

- $D^{-1/2}\vec{p}(t)$ 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

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

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

- $D^{-1/2} \tilde{p}(t)$ is exactly what would be obtained by applying $t/2$ iterations of power method to $D^{-1/2} \tilde{p}(0)$!
- Will converge to the top singular vector (eigenvector) of the normalized adjacency matrix $D^{-1/2}AD^{-1/2}$. Stationary distribution.
**Claim:** After $t$ steps, the probability that a random walk is at node $i$ is given by the $i^{th}$ entry of

$$
\bar{p}^{(t)} = AD^{-1}AD^{-1} \ldots AD^{-1} \bar{p}^{(0)}.
$$

$$
D^{-1/2} \bar{p}^{(t)} = (D^{-1/2}AD^{-1/2})(D^{-1/2}AD^{-1/2}) \ldots (D^{-1/2}AD^{-1/2})(D^{-1/2} \bar{p}^{(0)}).
$$

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

- Will converge to the top singular vector (eigenvector) of the normalized adjacency matrix $D^{-1/2}AD^{-1/2}$. **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 $AD^{-1}$. The **spectral gap.**