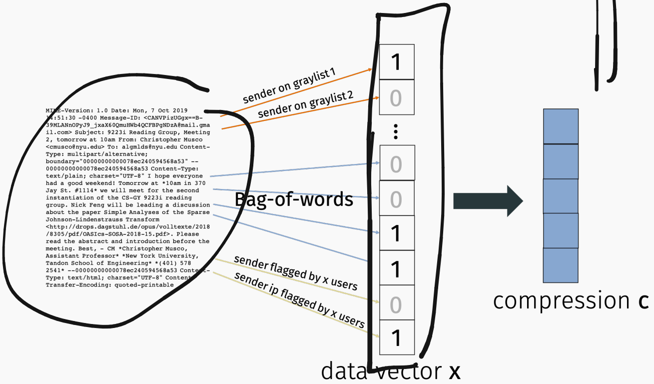
CS-GY 6763: Lecture 9
Low-rank approximation and singular value decomposition

NYU Tandon School of Engineering, Prof. Christopher Musco

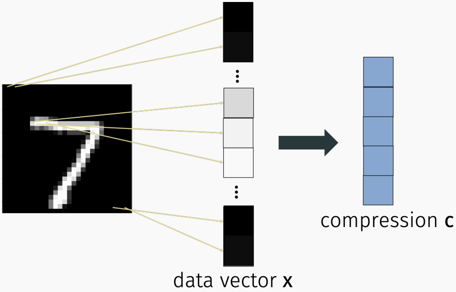
ADMINISTRATIVE

- Reading group tomorrow at 9:30am. Pat and Hogyeong are presenting on a method for the Frequent Items problem that improve on the CountMin method we learned in class in may scenarios.
- Midterms will be returned at the end of class.
- Problem Set 2 is being graded.

Return to data compression:

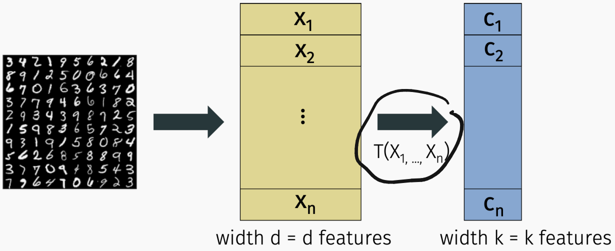


Return to data compression:



SPECTRAL METHODS

Main difference from randomized methods: $X_1 = \Pi x_1$



In this section, we will discuss data dependent transformations. Johnson-Lindenstrauss, MinHash, SimHash were all data oblivious.

Advantages of data independent methods:

- computationally more "light weight"
- don't require "full view" of data
  - streaming
  - distributed

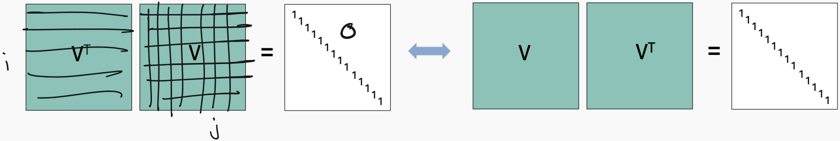
Advantages of data dependent methods:

- better compression when there's underlying structure in the data

If a square matrix has orthonormal rows, it also have orthonormal columns:

columns in A

$\|A\|_F^2 = \sum_{ij} A_{ij}^2 = \sum_i \|a_i\|_2^2$



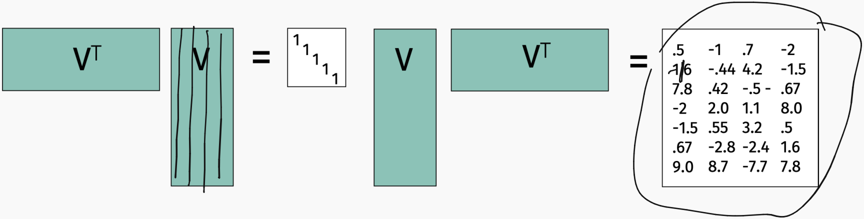
$$\mathbf{V}^T\mathbf{V} = \mathbf{I} = \mathbf{V}\mathbf{V}^T$$

Implies that for any vector $\mathbf{x}$, $\|\mathbf{V}\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2$ and $\|\mathbf{V}^T\mathbf{x}\|_2^2$.

$x^T V^T V x = x^T x = \|x\|_2^2$ $\quad$ $x^T V V^T x = x^T I x = \|x\|_2^2$

Same thing goes for Frobenius norm: for any matrix $\mathbf{X}$, $\|\mathbf{V}\mathbf{X}\|_F^2 = \|\mathbf{X}\|_F^2$ and $\|\mathbf{V}^T\mathbf{X}\|_F^2 = \|\mathbf{X}\|_F^2$.

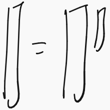
The same is not true for rectangular matrices:



projection matrix

$V^T V = I$ but $VV^T \neq I$

For any $\mathbf{x}$, $\|\mathbf{V}\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2$ but $\|\mathbf{V}^T\mathbf{x}\|_2^2 \neq \|\mathbf{x}\|_2^2$ in general.

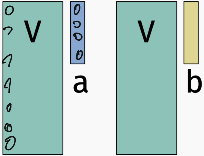


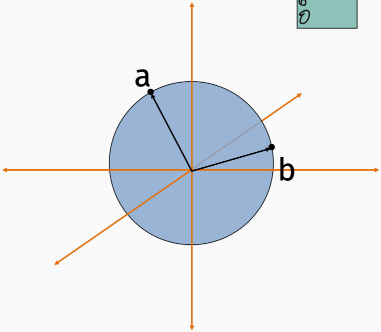
LINEAR ALGEBRA REMINDER

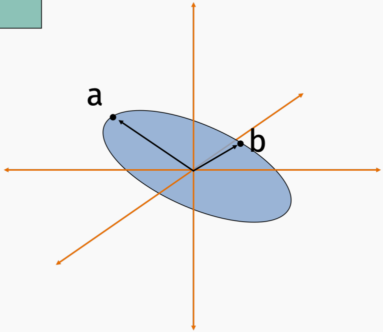
Multiplying a vector by $\mathbf{V}$ with orthonormal columns rotates and/or reflects the vector.

$$\|V(a-b)\|_2 = \|a-b\|_2$$

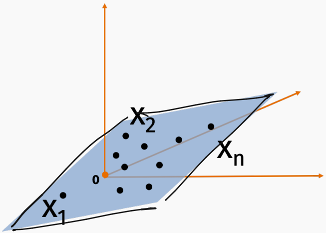
$$\|V(a+b)\|_2 = \|a+b\|_2$$



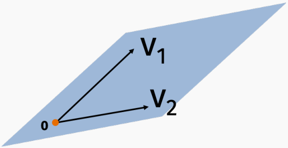




Suppose $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ lie on a low-dimensional subspace S through the origin. I.e. our data set is rank k for $k < d$.

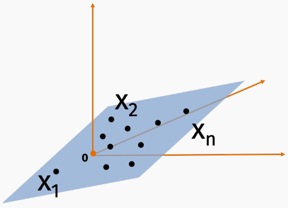


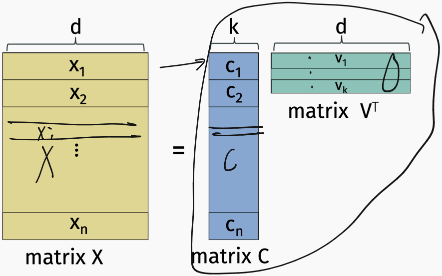
Let $\mathbf{v}_1, \ldots, \mathbf{v}_k$ be orthogonal unit vectors spanning S.



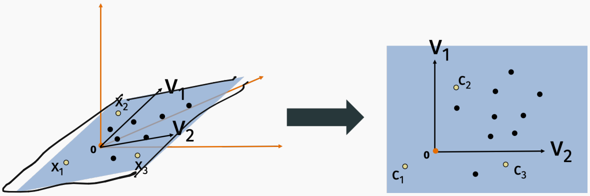
For all i, we can write:

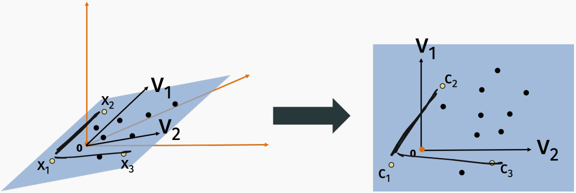
$$\mathbf{x}_i = c_{i,1}\mathbf{v}_1 + \ldots + c_{i,k}\mathbf{v}_k.$$





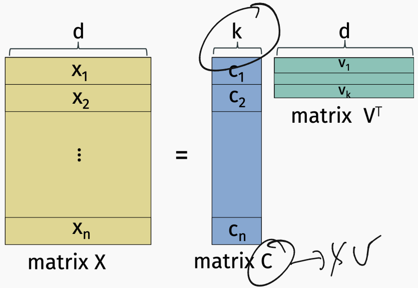
What are $\mathbf{c}_1, \ldots, \mathbf{c}_n$?





Lots of information preserved:

- $\|\mathbf{x}_i - \mathbf{x}_j\|_2 = \|\mathbf{c}_i - \mathbf{c}_j\|_2$ for all i, j.
- $\mathbf{x}_i^T \mathbf{x}_j = \mathbf{c}_i^T \mathbf{c}_j$ for all i, j.
- Norms preserved, linear separability preserved, $\min \|\mathbf{X}\mathbf{y} - \mathbf{b}\| = \min \|\mathbf{C}\mathbf{z} - \mathbf{b}\|$, etc., etc.



Formally, $\mathbf{C} = \mathbf{X}\mathbf{V}$:

$$\mathbf{X} = \mathbf{C}\mathbf{V}^T \Rightarrow \mathbf{X}\mathbf{V} = \mathbf{C}\underbrace{\mathbf{V}^T\mathbf{V}}_{\to I} \qquad XV = C$$

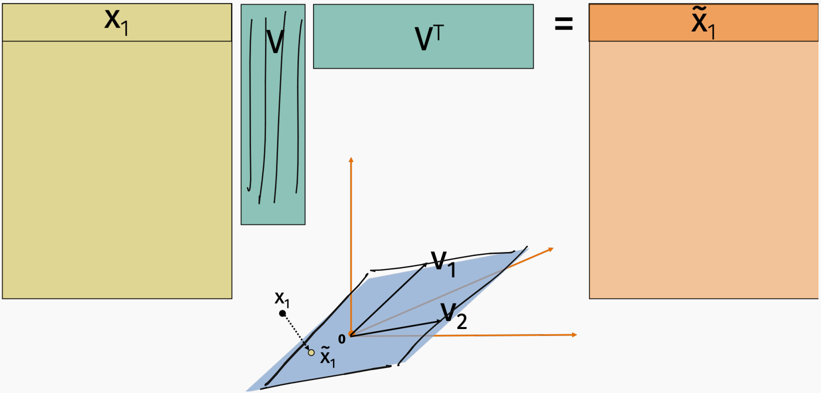
Since $\mathbf{V}$'s columns are an orthonormal basis, $\mathbf{V}^T\mathbf{V} = \mathbf{I}$.

So $\mathbf{X} = \mathbf{X}\mathbf{V}\mathbf{V}^T$.

for some V with k columns

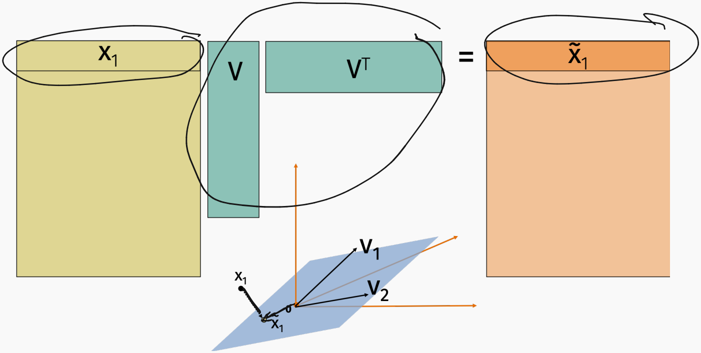
PROJECTION MATRICES

$\mathbf{V}\mathbf{V}^T$ is a symmetric projection matrix. $V^TV = I$



When all data points already lie in the subspace spanned by $\mathbf{V}$'s columns, projection doesn't do anything. So $\mathbf{X} = \mathbf{X}\mathbf{V}\mathbf{V}^T$.

VV^T is a symmetric projection matrix.



$\tilde{x}_i$: $x_1^T VV^T$ is the projection of x_1^T onto the subspace.

By pythagorean theorem, $\|x_1^T - x_1^T VV^T\|_2^2 = \|x_1^T\|_2^2 - \|x_1^T VV^T\|_2^2$ and by apply to all rows, $\|X - XVV^T\|_F^2 = \|X\|_F^2 - \|XVV^T\|_F^2$.

LOW-RANK APPROXIMATION

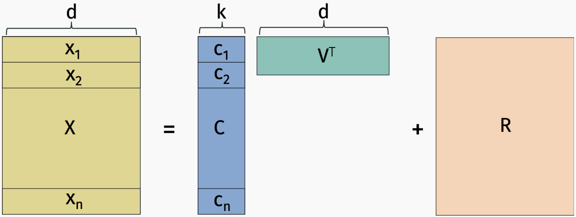
When $\mathbf{X}$'s rows lie close to a k dimensional subspace, we can still approximate

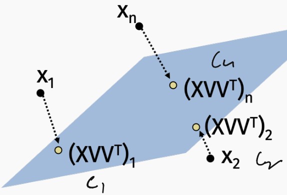
$$\underline{\mathbf{X}} \approx \underline{\mathbf{X}\mathbf{V}\mathbf{V}^T}.$$

$\mathbf{X}\mathbf{V}\mathbf{V}^T$ is a low-rank approximation for $\mathbf{X}$.

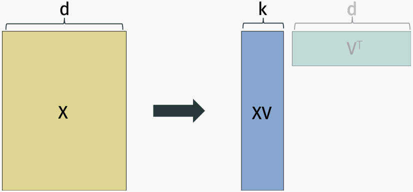
For a given subspace $\mathcal{V}$ spanned by the columns in $\mathbf{V}$,

$$\mathbf{X}\mathbf{V}\mathbf{V}^T = \arg\min_{\mathbf{C}} \|\mathbf{X} - \mathbf{C}\mathbf{V}^T\|_F^2 = \sum_{i,j} (\mathbf{X}_{i,j} - (\mathbf{C}\mathbf{V}^T)_{i,j})^2.$$



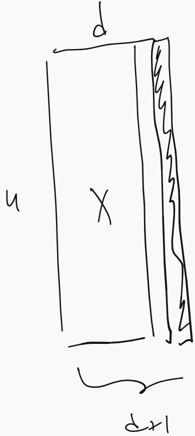


$$\|x_i - x_j\|_2 \approx \|x_i^T VV^T - x_j^T VV^T\|_2 = \|x_i^T V - x_j^T V\|_2$$

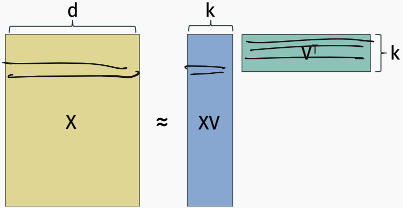


XV can be used as a compressed version of data matrix X.

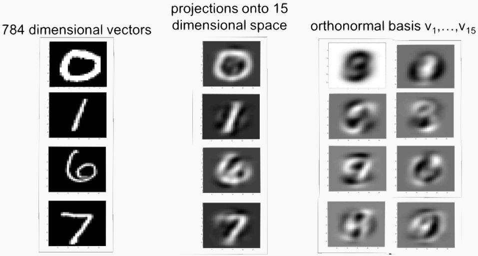
WHY IS DATA APPROXIMATELY LOW-RANK?



Rows of $\mathbf{X}$ (data points) are approximately spanned by k vectors. Columns of $\mathbf{X}$ (data features) are approximately spanned by k vectors.



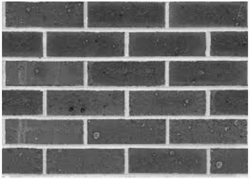
If a data set only had k unique data points, it would be exactly rank k. If it has k "clusters" of data points (e.g. the 10 digits) it's often very close to rank k.

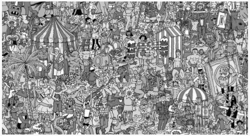


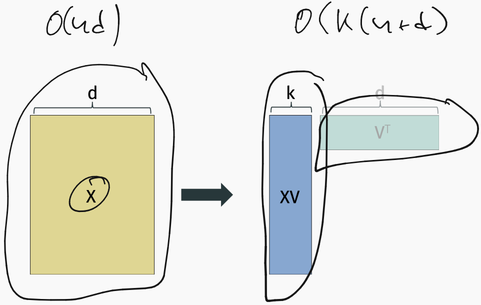
Colinearity/correlation of data features leads to a low-rank data matrix.

	bedrooms	bathrooms	sq.ft.	floors	list price	sale price
home 1	2	2	1800	2	200,000	195,000
home 2	4	2.5	2700	1	300,000	310,000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
home n	5	3.5	3600	3	450,000	450,000

When encoded as a matrix, which image has lower approximate rank?

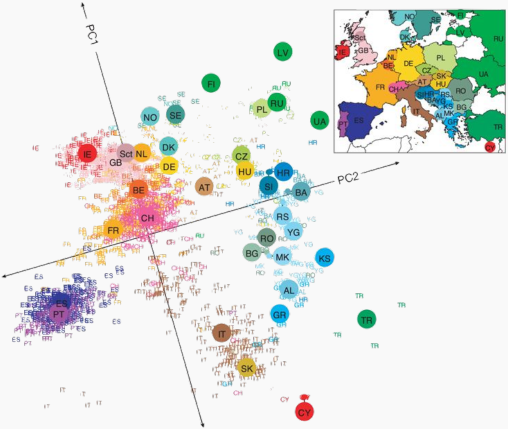






- $\mathbf{XV} \cdot \mathbf{V}^T$ takes $O(k(n+d))$ space to store instead of $O(nd)$.
- Regression problems involving $\mathbf{XV} \cdot \mathbf{V}^T$ can be solved in $O(nk^2)$ instead of $O(nd^2)$ time.
- $\mathbf{XV}$ can be used for visualization when $k = 2, 3$.

"Genes Mirror Geography Within Europe" – Nature, 2008.



Each data vector $\mathbf{x}_i$ contains genetic information for one person in Europe. Set $k = 2$ and plot $(XV)_i$ for each i on a 2-d plane. Color points by what country they are from.

Given a subspace $\mathcal{V}$ spanned by the k columns in $\mathbf{V}$,

$$\|\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 = \cancel{\min_{\mathbf{C}} \|\mathbf{X} - \mathbf{C}\mathbf{V}^T\|_F^2}$$

$\|Ax\|_2^2$

$\max_{x} \frac{\|Ax\|_2^2}{\|x\|_2^2}$

We want to find the best $\mathbf{V} \in \mathbb{R}^{d \times k}$:

$\|X\|_F^2 - \|XVV^T\|_F^2$

$$\min_{\text{orthonormal } \mathbf{V} \in \mathbb{R}^{d \times k}} \|\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 \tag{1}$$

Note that $\|\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2$ for all orthonormal $\mathbf{V}$ (since $\mathbf{V}\mathbf{V}^T$ is a projection). Equivalent form:

$$\max_{\text{orthonormal } \mathbf{V} \in \mathbb{R}^{d \times k}} \|\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 = \|\mathbf{X}\mathbf{V}\|_F^2 \tag{2}$$

If $k = 1$, want to find a single vector $\mathbf{v}_1$ which maximizes:

$$\|\mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2 = \|\mathbf{X}\mathbf{v}_1\|_F^2 = \|\mathbf{X}\mathbf{v}_1\|_2^2 = \mathbf{v}_1^T\mathbf{X}^T\mathbf{X}\mathbf{v}_1.$$

Choose $\mathbf{v}_1$ to be the top eigenvector of $\mathbf{X}^T\mathbf{X}$.

What about higher k?

$$\max_{v_1 : \|v_1\|_2 = 1} v_1^T X^T X v_1$$

$\mathrm{tr}(ABC) = \mathrm{tr}(BCA) = \mathrm{tr}(CAB)$

One-stop shop for computing optimal low-rank approximations.

$X^T = V \Sigma U^T$

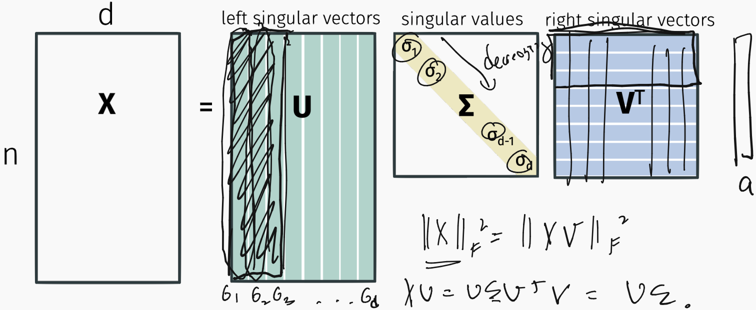
$X^T X = V \Sigma \underbrace{U^T U}_{I} \Sigma V^T = V \Sigma^2 V^T$

$\|X\|_F^2 = \mathrm{tr}(X^T X)$

$\|x\|_2^2 = x^T x$

Any matrix **X** can be written:

$\mathrm{tr}(U \Sigma^2 V^T) = \mathrm{tr}(\Sigma^2 V^T V) = \mathrm{tr}(\Sigma^2)$



$\|X\|_F^2 = \|XV\|_F^2$

$XV = U \Sigma V^T V = U\Sigma.$

Where $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, and $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_d \geq 0$.

Note that $\sum_{i=1}^{d} \sigma_i^2 = \|\mathbf{X}\|_F^2$.

$v_1, \dots v_d$

- $\mathbf{V}_k$'s columns are called the "top right singular vectors of $\mathbf{X}$"
- $\mathbf{U}_k$'s columns are called the "top left singular vectors of $\mathbf{X}$"
- $\sigma_1, \dots, \sigma_k$ are the "top singular values". $\sigma_1, \dots, \sigma_d$ are sometimes called the "spectrum of $\mathbf{X}$" (although this is more typically used to refer to eigenvalues).

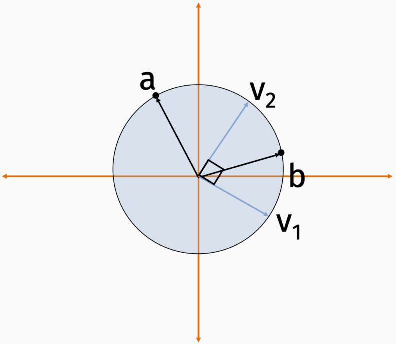
- $\mathbf{U}$ contains the orthonormal eigenvectors of $\mathbf{X}\mathbf{X}^T$.
- $\mathbf{V}$ contains the orthonormal eigenvectors of $\mathbf{X}^T\mathbf{X}$.
- $\sigma_i^2 = \lambda_i(\mathbf{X}\mathbf{X}^T) = \lambda_i(\mathbf{X}^T\mathbf{X})$

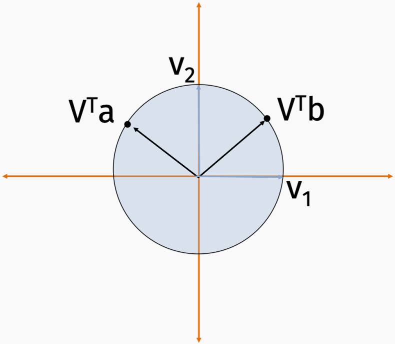
Exercise: Check this can be checked directly.

Important take away from singular value decomposition.

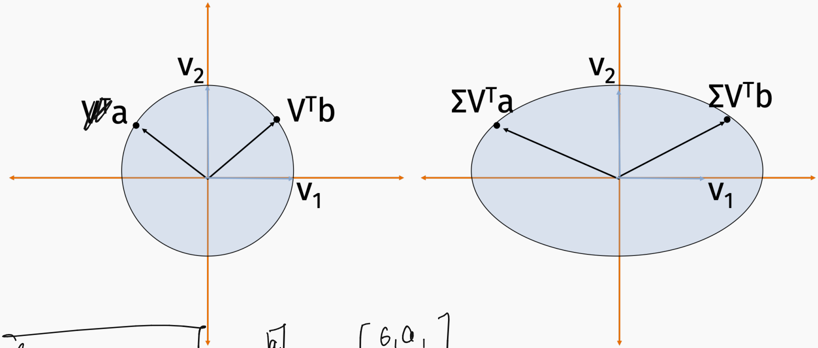
Multiplying any vector $\mathbf{a}$ by a matrix $\mathbf{X}$ to form $\mathbf{X}\mathbf{a}$ can be viewed as a composition of 3 operations:

1. Rotate/reflect the vector (multiplication by to $\mathbf{V}^T$).
2. Scale the coordinates (multiplication by $\mathbf{\Sigma}$.
3. Rotate/reflect the vector again (multiplication by $\mathbf{U}$).


a
v2
b
v1

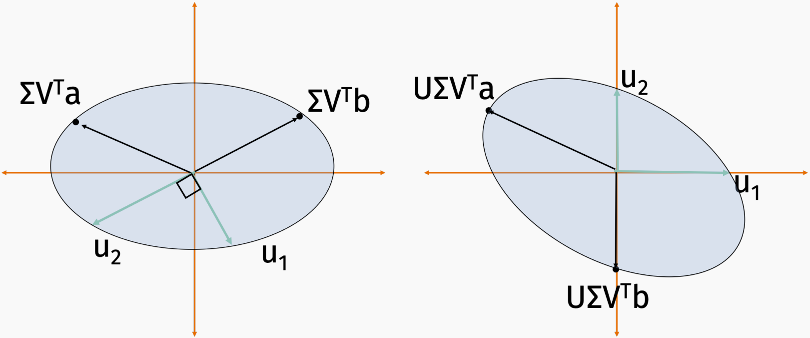

v2
VTa
VTb
v1

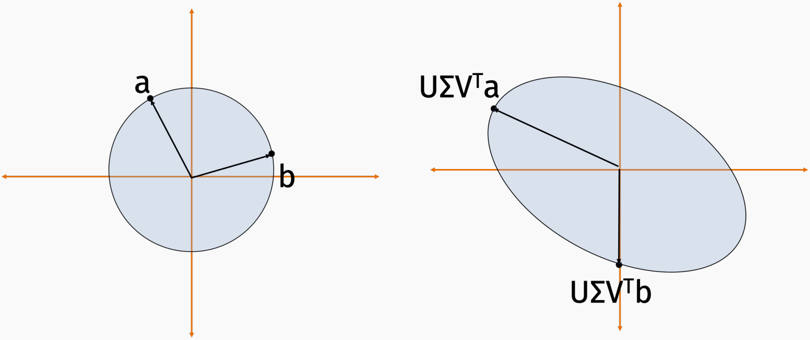
$\sigma_i > 0$



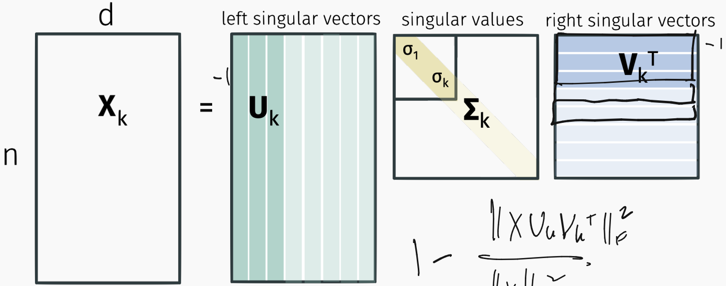
$$\underbrace{\begin{bmatrix}\sigma_1 & & & 0\\ & \sigma_2 & & \\ 0 & & \ddots & \\ & & & \sigma_d\end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix}a_1\\ \\ \\ a_d\end{bmatrix}}_{a} = \begin{bmatrix}\sigma_1 a_1 \\ \\ \\ \sigma_d a_d\end{bmatrix}$$

SINGULAR VALUE DECOMPOSITION: ROTATE/REFLECT




a
b
UΣV^T a
UΣV^T b

Can read off optimal low-rank approximations from the SVD:



$$1 - \frac{\|X U_k V_k^T\|_F^2}{\|X\|_F^2}$$

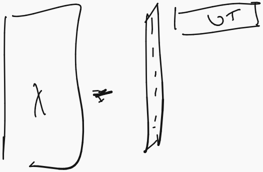
$$X_k = U_k \boldsymbol{\Sigma}_k V_k^T = U_k U_k^T X = X V_k V_k^T.$$

$$V_k = \underset{\text{orthonormal } V \in \mathbb{R}^{d\times k}}{\arg\min} \|X - XVV^T\|_F^2 = \underset{\text{orthonormal } V \in \mathbb{R}^{d\times k}}{\arg\max} \|XVV^T\|_F^2$$

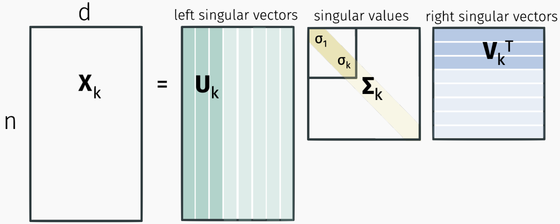
SINGULAR VALUE DECOMPOSITION

Connection to **Principal Component Analysis**:

- Let $\bar{\mathbf{X}} = \mathbf{X} - \mathbf{1}\boldsymbol{\mu}^T$ where $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$. I.e. $\bar{\mathbf{X}}$ is obtained by mean centering $\mathbf{X}$'s rows.
- Let $\bar{\mathbf{U}}\bar{\boldsymbol{\Sigma}}\bar{\mathbf{V}}^T$ be the SVD of $\bar{\mathbf{X}}$. $\bar{\mathbf{U}}$'s first columns are the "top principal components" of $\mathbf{X}$. $\bar{\mathbf{V}}$'s first columns are the "weight vectors" for these principal components.

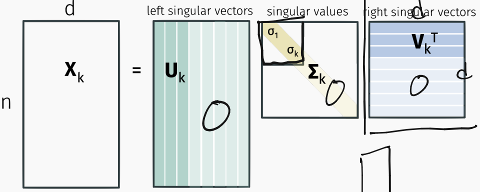


USEFUL OBSERVATIONS



~~**Observation 1:** The optimal compression $\mathbf{X}\mathbf{V}_k$ has orthogonal columns.~~

$V^T V_k = \begin{bmatrix} I \\ 0 \end{bmatrix}$ $\quad XV_kV_k^T = U_k \Sigma_k V_k^T$



Observation 2: The optimal low-rank approximation error $E_k = \|X - XV_kV_k^T\|_F^2 = \|X\|_F^2 - \|XV_kV_k^T\|_F^2$ can be written:

$\sum_{i=1}^{d} \sigma_i^2 \qquad \sum_{i=1}^{k} \sigma_i^2$

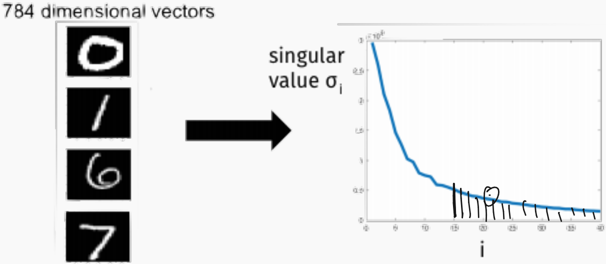
$$E_k = \sum_{i=k+1}^{d} \sigma_i^2.$$

$XV_kV_k^T = U\Sigma V^T V_k V_k^T$

Observation 2: The optimal low-rank approximation error $E_k = \|\mathbf{X} - \mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2$ can be written:

$$E_k = \sum_{i=k+1}^{d} \sigma_i^2.$$

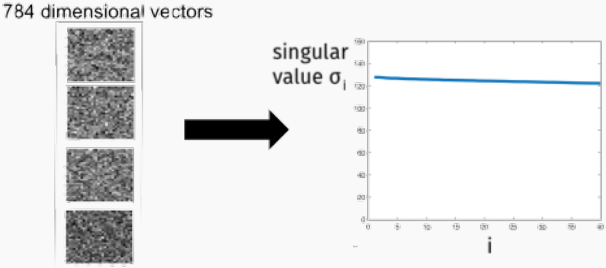
Can immediately get a sense of "how low-rank" a matrix is from it's spectrum:



Observation 2: The optimal low-rank approximation error $E_k = \|\mathbf{X} - \mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2$ can be written:

$$E_k = \sum_{i=k+1}^{d} \sigma_i^2.$$

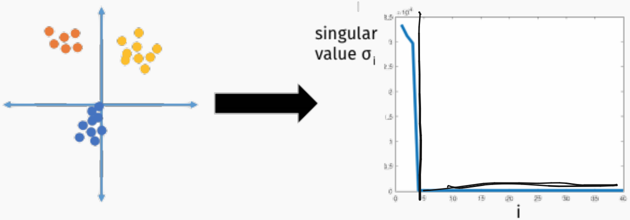
Can immediately get a sense of "how low-rank" a matrix is from it's spectrum:



Observation 2: The optimal low-rank approximation error $E_k = \|\mathbf{X} - \mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2$ can be written:

$$E_k = \sum_{i=k+1}^{d} \sigma_i^2.$$

Can immediately get a sense of "how low-rank" a matrix is from it's spectrum:



Suffices to compute right singular vectors **V**: $\rightarrow O(d^3)$

- Compute $\mathbf{X}^T\mathbf{X}$. $d \times d$ $\quad d \times n \quad n \times d \rightarrow O(nd^2)$
- Find eigendecomposition $\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T = \mathbf{X}^T\mathbf{X}$.
- Compute $\mathbf{L} = \mathbf{X}\mathbf{V}$. Set $\sigma_i = \|\mathbf{L}_i\|_2$ and $\mathbf{U}_i = \mathbf{L}_i / \|\mathbf{L}_i\|_2$.

Total runtime $\approx \underline{O(nd^2)} + O(d^3)$

$X = U \Sigma V^T$

$\underline{XV} = U\Sigma$

- Compute approximate solution.
- Only compute top k singular vectors/values. Runtime will depend on k. When $k = d$ we can't do any better than classical algorithms based on eigendecomposition.
- Iterative algorithms achieve runtime $\approx O(ndk)$ vs. $O(nd^2)$ time.
  - **Krylov subspace methods** like the Lanczos method are most commonly used in practice.
  - **Power method** is the simplest Krylov subspace method, and still works very well.

What we won't discuss today: sketching methods and stochastic methods (which are faster in some settings).

Today: What about when $k = 1$?

Goal: Find some $\mathbf{z} \approx \mathbf{v}_1$.

Input: $\mathbf{X} \in \mathbb{R}^{n \times d}$ with SVD $\mathbf{U\Sigma V}^T$.

Power method:

- Choose $\mathbf{z}^{(0)}$ randomly. E.g. $\mathbf{z}_0 \sim \mathcal{N}(0, 1)$.
- $\mathbf{z}^{(0)} = \mathbf{z}^{(0)} / \|\mathbf{z}^{(0)}\|_2$
- For $i = 1, \ldots, T$
  - $\mathbf{z}^{(i)} = \mathbf{X}^T \cdot (\mathbf{X}\mathbf{z}^{(i-1)})$
  - $n_i = \|\mathbf{z}^{(i)}\|_2$
  - $\mathbf{z}^{(i)} = \mathbf{z}^{(i)} / n_i$

  Return $\mathbf{z}^{(T)}$

$X^T X = V \Sigma U^T U \Sigma V^T = V \Sigma^2 V^T$

$V \underbrace{\Sigma^2 V^T z}$

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

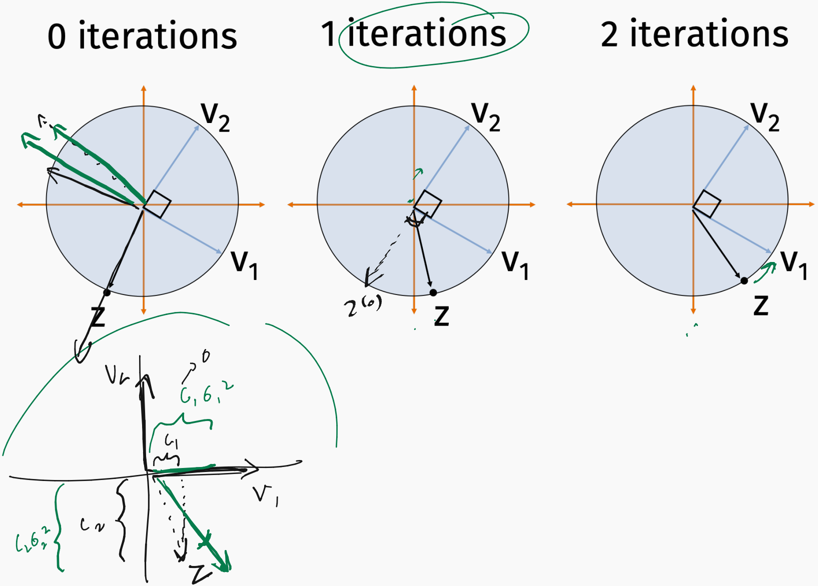
$z^{(i)} = c_1 v_1 + c_2 v_2 + \ldots c_d v_d$

$\Sigma^2 V^T z = \boxed{\begin{bmatrix} \sigma_1^2 c_1 \\ \sigma_2^2 c_1 \\ \vdots \\ \sigma_d^2 c_d \end{bmatrix}}$

$z^{(i+1)} = c_1 \sigma_1^2 v_1 + \ldots + c_d \sigma_d^2 v_d$

$z^{(i)} = \frac{1}{\prod_{j=1}^{i} n_j} \underbrace{(X^T X)(X^T X) \ldots (X^T X)}_{(X^T X)^i} z^{0}$

$(X^T X)^i z^{(0)}$


0 iterations
1 iterations
2 iterations
V2
V1
Z
V2
V1
Z
V2
V1
Z

Theorem (Basic Power Method Convergence)

Let $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$ be parameter capturing the "gap" between the first and second largest singular values. If Power Method is initialized with a random Gaussian vector then, with high probability, after $T = O\left(\frac{\log d/\epsilon}{\gamma}\right)$ steps, we have either:

$$\|\mathbf{v}_1 - \mathbf{z}^{(T)}\|_2 \leq \epsilon \qquad \text{or} \qquad \|\mathbf{v}_1 - (-\mathbf{z}^{(T)})\|_2 \leq \epsilon.$$

Total runtime: $O\left(nd \cdot \frac{\log d/\epsilon}{\gamma}\right) \cdot k$)

$O(nd^2)$

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

$$\begin{aligned}
\mathbf{z}^{(0)} &= c_1^{(0)}\mathbf{v}_1 + c_2^{(0)}\mathbf{v}_2 + \ldots + c_d^{(0)}\mathbf{v}_d \\
\mathbf{z}^{(1)} &= \underline{c_1^{(1)}\mathbf{v}_1 + c_2^{(1)}\mathbf{v}_2 + \ldots + c_d^{(1)}\mathbf{v}_d} \\
&\vdots \\
\mathbf{z}^{(i)} &= c_1^{(i)}\mathbf{v}_1 + c_2^{(i)}\mathbf{v}_2 + \ldots + c_d^{(i)}\mathbf{v}_d
\end{aligned}$$

Note: $[c_1^{(i)}, \ldots, c_d^{(i)}] = \mathbf{c}^{(i)} = \underline{\mathbf{V}^T\mathbf{z}^{(i)}}$.

Also: $\underline{\sum_{j=1}^{d} \left(c_j^{(i)}\right)^2} = 1.$

Claim: After update $\mathbf{z}^{(i)} = \frac{1}{n}\mathbf{X}^T\mathbf{X}\mathbf{z}^{(i-1)}$,

$$c_j^{(i)} = \frac{1}{n}\sigma_j^2 c_j^{(i-1)}$$

$$\mathbf{z}^{(i)} = \frac{1}{n}\left[c_1^{(i-1)}\sigma_1^2 \cdot \mathbf{v}_1 + c_2^{(i-1)}\sigma_2^2 \cdot \mathbf{v}_2 + \ldots + c_d^{(i-1)}\sigma_d^2 \cdot \mathbf{v}_d\right]$$

Claim: After T updates:

near 1 near 0 near 0

$$\mathbf{z}^{(T)} = \frac{1}{\prod_{i=1}^{T} n_i}\left[c_1^{(0)}\sigma_1^{2T}\cdot\mathbf{v}_1 + c_2^{(0)}\sigma_2^{2T}\cdot\mathbf{v}_2 + \ldots + c_d^{(0)}\sigma_d^{2T}\cdot\mathbf{v}_d\right]$$

$\approx v_1$

$[1 \cdot v_1 + 0\, v_2 + \ldots + \ldots 0 \cdot v_d$

Let $\alpha_j = \frac{1}{\prod_{i=1}^{T} n_i} c_j^{(0)}\sigma_j^{2T}$. **Goal:** Show that $\alpha_j \ll \alpha_1$ for all $j \neq 1$.

$\alpha_1 \approx 1$

POWER METHOD FORMAL CONVERGENCE

Since $\mathbf{z}^{(T)}$ is a unit vector, $\sum_{i=1}^{d} \alpha_i^2 = 1$. So $\alpha_1 \leq 1$.

If we can prove that $\frac{\alpha_j}{\alpha_1} \leq \sqrt{\frac{\epsilon}{d}}$ then:

$$\alpha_j^2 \leq \alpha_1^2 \cdot \frac{\epsilon}{d}$$

$$1 = \alpha_1^2 + \sum_{j=2}^{d} \alpha_j^2 \leq \alpha_1^2 + \epsilon$$

$$\alpha_1^2 \geq 1 - \epsilon$$

$$|\alpha_1| \geq 1 - \epsilon$$

$$\|\mathbf{v}_1 - \mathbf{z}^{(T)}\|_2 = 2 - 2\langle \mathbf{v}_1, \mathbf{z}^{(T)} \rangle \leq 2\epsilon$$

Lets proves that $\frac{\alpha_j}{\alpha_1} \leq \sqrt{\frac{\epsilon}{d}}$ where $\alpha_j = \frac{1}{\prod_{i=1}^{T} n_i} c_j^{(0)} \sigma_j^{2T}$

First observation: Starting coefficients are all roughly equal.

$$\text{For all } j \qquad O(1/d^3) \leq c_j^{(0)} \leq 1$$

with probability $1 - \frac{1}{d}$. This is a very loose bound, but it's all suff. that we will need. **Prove using Gaussian concentration.**

$$\frac{\alpha_j}{\alpha_1} = \boxed{\frac{\sigma_j^{2T}}{\sigma_1^{2T}}} \cdot \boxed{\frac{c_j^{(0)}}{c_1^{(0)}}} \leq \frac{\sigma_j^{2T}}{\sigma_1^{2T}} \cdot \frac{1}{O(1/d^3)} = \frac{\sigma_j^{2T}}{\sigma_1^{2T}} \cdot O(d^3)$$

$$= \left(\frac{\sigma_j}{\sigma_1}\right)^{2T} \cdot O(d^3)$$

$$\leq (1-\gamma)^{2T} \cdot O(d^3)$$

$$\leq \frac{\epsilon^c}{d^c} \quad \text{if } T = O\left(\frac{\log d/\epsilon}{\gamma}\right)$$

$$\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$$

$$\gamma = 1 - \frac{\sigma_2}{\sigma_1} \qquad \frac{\sigma_2}{\sigma_1} = 1 - \gamma$$

Need $T =$

Theorem (Gapless Power Method Convergence)

If Power Method is initialized with a random Gaussian vector then, with high probability, after $T = O\left(\frac{\log d/\epsilon}{\epsilon}\right)$ steps, we obtain a $\mathbf{z}$ satisfying:

$$\|\mathbf{X} - \mathbf{X}\mathbf{z}\mathbf{z}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2$$

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

Power method:

- Choose $\mathbf{G} \in \mathbb{R}^{d \times k}$ be a random Gaussian matrix.
- $\mathbf{Z}_0 = \text{orth}(\mathbf{G})$.
- For $i = 1, \ldots, T$
  - $\mathbf{Z}^{(i)} = \mathbf{X}^T \cdot (\mathbf{X}\mathbf{Z}^{(i-1)})$
  - $\mathbf{Z}^{(i)} = \text{orth}(\mathbf{Z}^{(i)})$

  Return $\mathbf{Z}^{(T)}$

$$\underbrace{X^T \underbrace{X Z^{(i-1)}}_{O(ndk)}}_{O(ndk)}$$

Runtime: $O\left(\frac{\log d/\epsilon}{\epsilon}\right)$ iterations to obtain a nearly optimal low-rank approximation:

$$\|\mathbf{X} - \mathbf{X}\mathbf{Z}\mathbf{Z}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{X} - \mathbf{X}\mathbf{V_k}\mathbf{V_k}^T\|_F^2.$$

Possible to "accelerate" these methods.

Convergence Guarantee: $T = O\left(\frac{\log d/\epsilon}{\sqrt{\epsilon}}\right)$ iterations to obtain a nearly optimal low-rank approximation:

$$\|\mathbf{A} - \mathbf{A}\mathbf{Z}\mathbf{Z}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}\mathbf{V_k}\mathbf{V_k}^T\|_F^2.$$

Runtime: $O(\text{nnz}(\mathbf{X}) \cdot k \cdot T) \leq O(ndk \cdot T)$.

$$p(X^TX) \cdot g$$

$$\mathbf{z}^{(q)} = c \cdot (\mathbf{X}^T\mathbf{X})^q \cdot \mathbf{g}$$

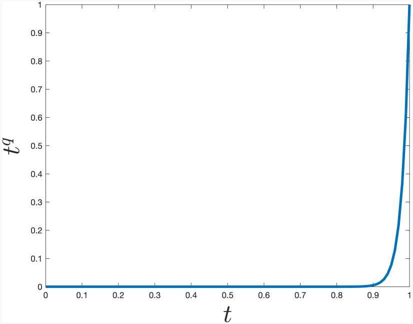
$(c_0 g + c_1 X^T X g$

$+ c_2 (X^T X)^2 g$

$+$

$\ddots$

$c_T (X^T X)^T g)$



$$\mathbf{z}^{(q)} = c \cdot \left[c_1 \cdot \sigma_1^{2q}\mathbf{v}_1 + c_2 \cdot \sigma_2^{2q}\mathbf{v}_2 + \ldots + c_n \cdot \sigma_n^{2q}\mathbf{v}_n\right]$$

$$\mathbf{z}^{(q)} = c \cdot \left(\mathbf{X}^T\mathbf{X}\right)^q \cdot \mathbf{g}$$

Along the way we computed:

$$\mathcal{K}_q = \left[\mathbf{g}, \left(\mathbf{X}^T\mathbf{X}\right) \cdot \mathbf{g}, \left(\mathbf{X}^T\mathbf{X}\right)^2 \cdot \mathbf{g}, \ldots, \left(\mathbf{X}^T\mathbf{X}\right)^q \cdot \mathbf{g}\right]$$

$\mathcal{K}$ is called the Krylov subspace of degree q.

Idea behind Krlyov methods: Don't throw away everything before $\left(\mathbf{X}^T\mathbf{X}\right)^q \cdot \mathbf{g}$. What you're using when you run `svds` or `eigs` in MATLAB or Python.

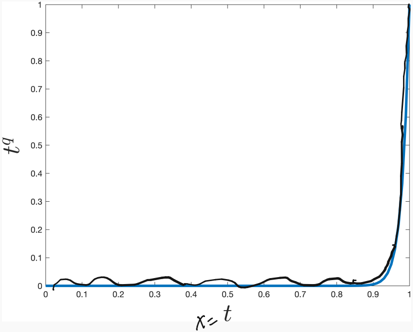
Want to find $\mathbf{v}$, which minimizes $\|\mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T\|_F^2$.

Lanczos method:

- Let $\mathbf{Q} \in \mathbb{R}^{d \times k}$ be an orthonormal span for the vectors in $\mathcal{K}$.
- Solve $\min_{\mathbf{v}=\mathbf{Q}\mathbf{w}} \|\mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T\|_F^2$.
    - Find <u>best</u> vector in the Krylov subspace, instead of just using last vector.
    - Can be done in $O\left(\mathrm{nnz}(\mathbf{X}) \cdot k + dk^2\right)$ time.

Claim: There is an $O\left(\sqrt{q\log\frac{1}{\epsilon}}\right)$ $\to O(\sqrt{q})$ degree polynomial $\hat{p}$ approximating $\mathbf{x}^q$ up to error $\epsilon\sigma_1^2$ on $[0, \sigma_1^2]$.

$\sigma_1^2 = 1$



$$\|\mathbf{X} - \mathbf{X}\mathbf{v}_{p*}\mathbf{v}_{p*}^T\|_F^2 \leq \|\mathbf{X} - \mathbf{X}\mathbf{v}_{\hat{p}}\mathbf{v}_{\hat{p}}^T\|_F^2 \approx \|\mathbf{X} - \mathbf{X}\mathbf{v}_{x^q}\mathbf{v}_{x^q}^T\|_F^2 \approx \|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2$$

Runtime: $O\left(\frac{\log(d/\epsilon)}{\sqrt{\gamma}} \cdot \mathrm{nnz}(\mathbf{X})\right)$ vs. $O\left(\frac{\log(d/\epsilon)}{\gamma} \cdot \mathrm{nnz}(\mathbf{X})\right)$

POWER METHOD – NO GAP DEPENDENCE

Convergence is slow when $\gamma = \frac{\sigma_1 - \sigma_2}{\sigma_1}$ is small. $\mathbf{z}^{(q)}$ has large components of <u>both</u> $\mathbf{v}_1$ and $\mathbf{v}_2$. But in this case:

$$\|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2 = \sum_{i \neq 1} \sigma_i^2 \approx \sum_{i \neq 2} = \sigma_i^2 \|\mathbf{X} - \mathbf{X}\mathbf{v}_2\mathbf{v}_2^T\|_F^2.$$

So we don't care! Either $\mathbf{v}_1$ or $\mathbf{v}_2$ give good rank-1 approximations.

Claim: To achieve

$$\|\mathbf{X} - \mathbf{X}\mathbf{z}\mathbf{z}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{X} - \mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2$$

we need $O\left(\frac{\log(d/\epsilon)}{\epsilon}\right)$ power method iterations or $O\left(\frac{\log(d/\epsilon)}{\sqrt{\epsilon}}\right)$ Lanczos iterations.

GENERALIZATIONS TO LARGER k

- Block Krylov methods
- Let $\mathbf{G} \in \mathbb{R}^{d \times k}$ be a random Gaussian matrix.
- $\mathcal{K}_q = \left[\mathbf{G}, (\mathbf{X}^T\mathbf{X}) \cdot \mathbf{G}, (\mathbf{X}^T\mathbf{X})^2 \cdot \mathbf{G}, \ldots, (\mathbf{X}^T\mathbf{X})^q \cdot \mathbf{G}\right]$

Runtime: $O\left(\text{nnz}(\mathbf{X}) \cdot k \cdot \frac{\log d/\epsilon}{\sqrt{\epsilon}}\right)$ to obtain a nearly optimal low-rank approximation.