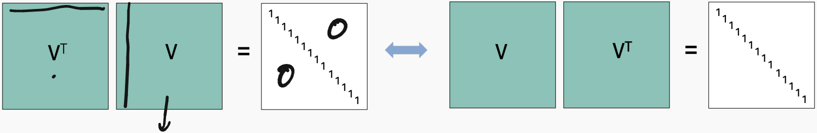
CS-GY 6763: Lecture 10
Singular value decomposition, low-rank approximation, Krylov subspace methods

NYU Tandon School of Engineering, Prof. Christopher Musco

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



v_1

$v_i^T v_j$ for $j \neq i$

this $= 0$

$$V^T V = I = VV^T$$

$$(V^T)^T V^T = I$$

$$\begin{bmatrix} -0.62 & 0.78 & -0.11 \\ -0.28 & -0.35 & -0.89 \\ -0.73 & -0.52 & 0.44 \end{bmatrix} \cdot \begin{bmatrix} -0.62 & -0.28 & -0.73 \\ 0.78 & -0.35 & -0.52 \\ -0.11 & -0.89 & 0.44 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

$$\begin{bmatrix} x_1, \ldots x_k \end{bmatrix}$$

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

$$\|M\|_F^2 = \sum_{ij} M_{ij}^2$$

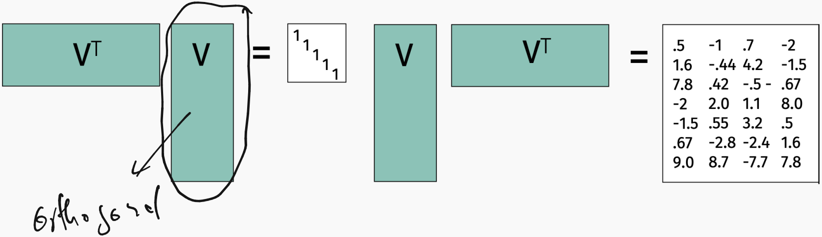
$$\|Vx\|_2^2 = \|x\|_2^2$$

$$= (Vx)^T Vx = x^T \underbrace{V^T V}_{=I} x = x^T x = \|x\|_2^2$$

$$VX = \begin{bmatrix} Vx_1, \ldots Vx_k \end{bmatrix}$$

$$\|VX\|_F^2 = \sum_{i=1}^{k} \|Vx_i\|_2^2 = \sum_{i=1}^{k} \|x_i\|_2^2 = \|X\|_F^2$$

The same is <u>not true</u> for rectangular matrices:



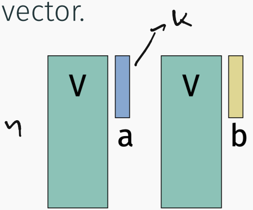
$$\mathbf{V}^T\mathbf{V} = \mathbf{I} \qquad \text{but} \qquad \mathbf{V}\mathbf{V}^T \neq \mathbf{I}$$

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.

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

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

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

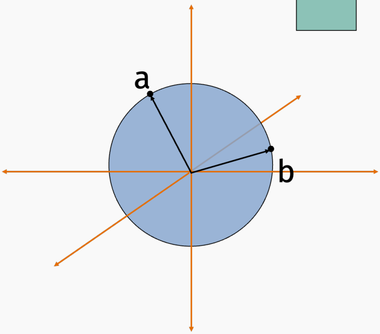


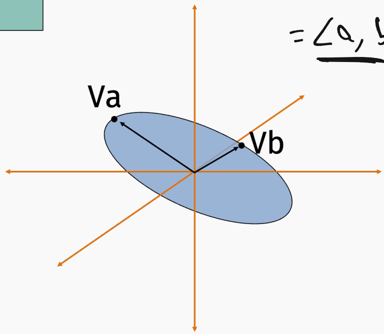
$$\underline{\|Va\|_2^2 = \|a\|_2^2}$$

$$\langle Va, Vb\rangle$$

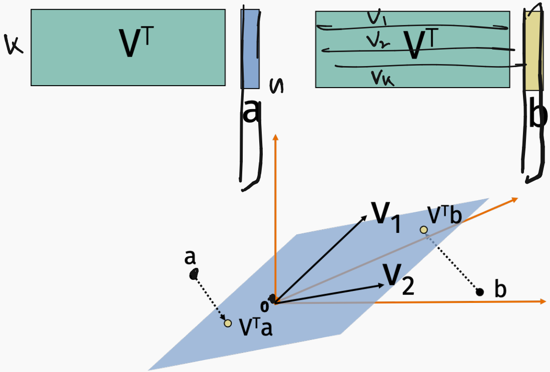
$$= a^T V^T V b = a^T b$$

$$= \underline{\langle a, b\rangle}$$





Multiplying a vector by a rectangular matrix $\mathbf{V}^T$ with orthonormal rows projects the vector (representing it as coordinates in the lower dimensional space).



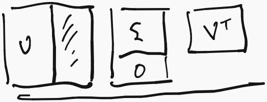
So we always have that $\|\mathbf{V}^T\mathbf{x}\|_2 \leq \|\mathbf{x}\|_2$.

Reduced SVD

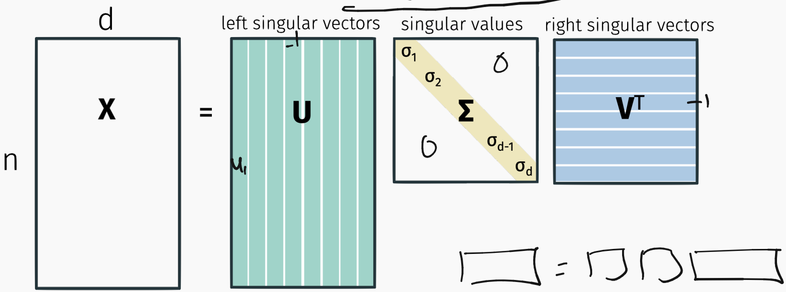
Economy SVD

One of the most fundamental results in linear algebra.

Any matrix **X** can be written:



$U^T u_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$



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

Singular values are unique. Factors are not. Would still get a valid SVD by multiplying both i^{th} column of **V** and **U** by -1.

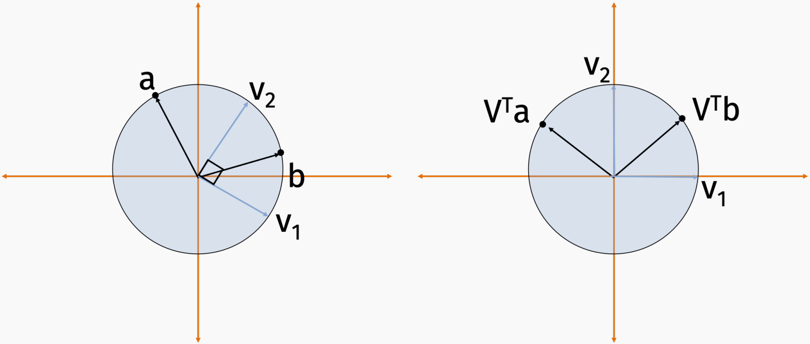
Important take away from singular value decomposition.

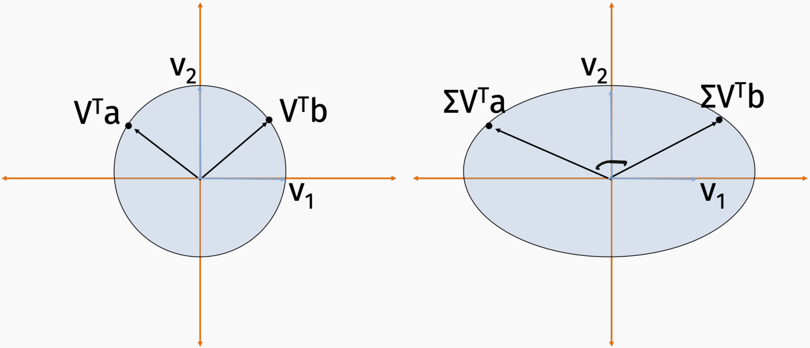
Multiplying any vector **a** by a matrix **X** to form **Xa** can be viewed as a composition of 3 operations:

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 **U**).

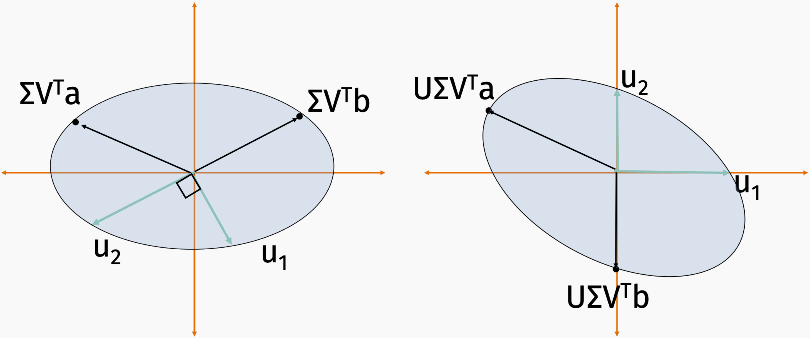
$$Xa = U\Sigma V^T a$$
$$= U(\Sigma(V^T a))$$

SINGULAR VALUE DECOMPOSITION: ROTATE/REFLECT

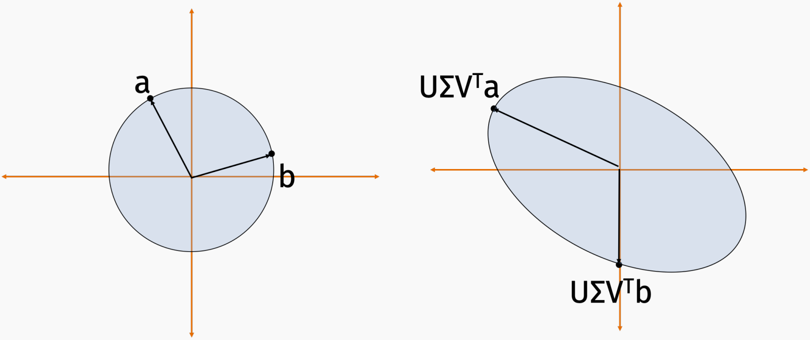



v_2
V^Ta
V^Tb
v_1
v_2
ΣV^Ta
ΣV^Tb
v_1

Singular Value Decomposition: Rotate/Reflect



SINGULAR VALUE DECOMPOSITION



COMPARISON TO EIGENDECOMPOSITION

scalar

Recall that an eigenvalue of a square matrix $\mathbf{X} \in \mathbb{R}^{d \times d}$ is any vector $\mathbf{v}$ such that $\mathbf{X}\mathbf{v} = \lambda \mathbf{v}$. A matrix has at most d linearly independent eigenvectors. If a matrix has a full set of d eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_d$ with eigenvalues $\lambda_1, \ldots, \lambda_n$ it is called "diagonalizable" and can be written as:

$X v_i = \lambda_i v_i$

$$\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}.$$

$V^{-1} \neq V^T$

$$\begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \qquad \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$$

Singluar value decomposition

- Exists for all matrices, square or rectangular.
- Singular values are always positive.
- Factors **U** and **V** are orthogonal.

Eigendecomposition

- Exists for some square matrices.
- Eigenvalues can be positive or negative.
- Factor **V** is orthogonal if and only if **X** is symmetric.

$V^{-1} = V^T$ $\quad V \Lambda V^T$

- U contains the orthogonal eigenvectors of XX^T.
- V contains the orthogonal eigenvectors of X^TX.
- $\sigma_i^2 = \lambda_i(XX^T) = \lambda_i(X^TX)$

$\sigma_1 \cdots \sigma_n$

$\Sigma \cdot \Sigma = \sigma_1^2 \cdots \sigma_n^2$

$X = U \Sigma V^T \qquad X^T = V \Sigma U^T \qquad XX^T = U \Sigma \underbrace{\cancel{V^T V}}_{I} \Sigma U^T$

$XX^T = U \Sigma^2 U^T$

$\begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2 \end{bmatrix}$

$XX^T u_1 = U \Sigma^2 \underbrace{U^T u_1}_{e_1}$ $\qquad e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

$= \lambda_1 u_1$

$= U \Sigma^2 e_1$

$= U \begin{bmatrix} \sigma_1^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \sigma_1^2 \cdot u_1$

$\min_a \|Xa - b\|_2^2$

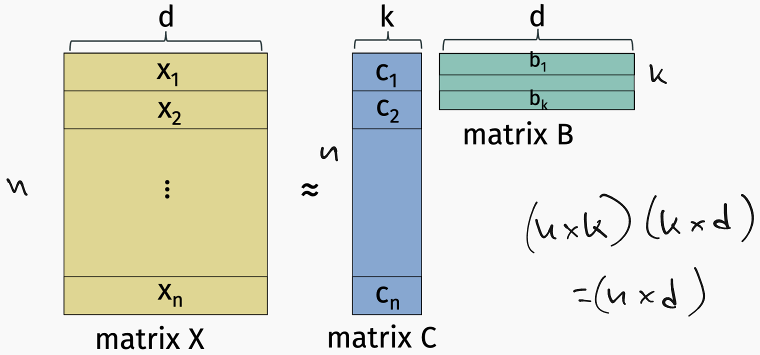
Lots of applications.

$\max \frac{\|Ax\|_2}{\|x\|_2}$

- Compute pseudoinverse $\mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{U}^T$.
- Read off condition number of $\mathbf{X}$, σ_1^2/σ_d^2.
- Compute matrix norms. E.g. $\|\mathbf{X}\|_2 = \sigma_1$, $\|\mathbf{X}\|_F = \sqrt{\sum_{i=1}^d \sigma_i^2}$.
- Compute matrix square root – i.e. find a matrix $\mathbf{B}$ such that $\mathbf{B}\mathbf{B}^T = \mathbf{X}$. Used e.g. in sampling from Gaussian with covariance $\mathbf{X}$.
- Principal component analysis.

Killer app: Read off optimal low-rank approximations for $\mathbf{X}$.

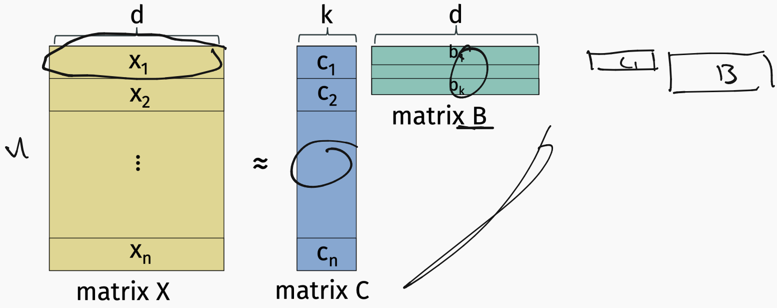
Approximate $\mathbf{X}$ as the product of two rank k matrices:



Typically choose $\mathbf{C}$ and $\mathbf{B}$ to minimize:

$$\min_{\mathbf{B},\mathbf{C}} \|\mathbf{X} - \mathbf{C}\mathbf{B}\|$$

for some matrix norm. Common choice is $\|\mathbf{X} - \mathbf{C}\mathbf{B}\|_F^2$.



nd

kd + kn

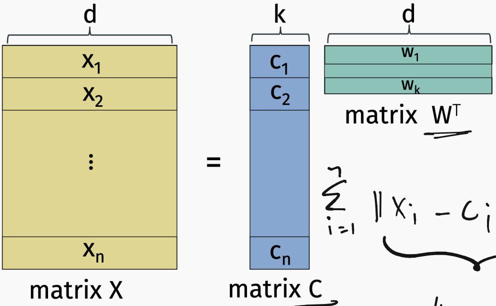
- CB takes $O(k(n+d))$ space to store instead of $O(nd)$.
- Regression problems involving CB can be solved in $O(nk^2)$ instead of $O(nd^2)$ time.
- Will see a bunch more in a minute.

Without loss of generality can assume that the right matrix is orthogonal. I.e. $\mathbf{W}^T$ with $\mathbf{W}^T\mathbf{W} = \mathbf{I}$

$x_1 \approx c_1 B$

$\downarrow$

$z_1 W^T = c_1 B$



$$\underbrace{\sum_{i=1}^{n} \|x_i - c_i W^T\|_2^2}$$

$$= \underbrace{\|x_i - W c_i\|_2^2}$$

Then we should choose $\mathbf{C}$ to minimize:

$$\min_{\mathbf{C},W} \|\mathbf{X} - \mathbf{C}\mathbf{W}^T\|_F^2$$

This is just n least squares regression problems!

$\min \|Az - b\|_2^2$ $\quad z^* = (A^T A)^{-1} A^T b$

$2A^T(Az - b) = 0$

$$\mathbf{c}_i = \arg\min_{\mathbf{c}} \|\mathbf{W}\mathbf{c} - \mathbf{x}_i\|_2^2$$

$(W^T W)^{-1} W^T x_i$

I

$$\mathbf{c}_i = \mathbf{W}^T \mathbf{x}_i$$

$$\mathbf{C} = \mathbf{X}\mathbf{W}$$

So our optimal low-rank approximation always has the form:

$$\mathbf{X} \approx \mathbf{X}\mathbf{W}\mathbf{W}^T$$

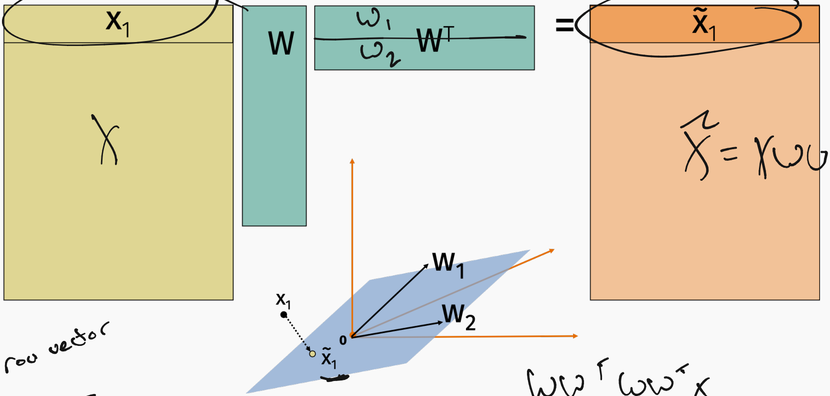
$\min_W \|X - XWW^T\|_F$

PROJECTION MATRICES

WW^T is a symmetric projection matrix.

$XWW^T \rightarrow$ projection

$P \cdot P = P$



row vector

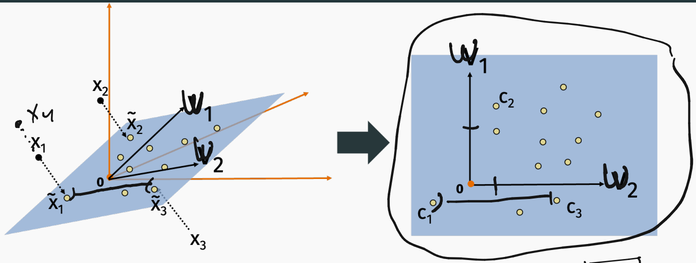
$x_1 WW^T$

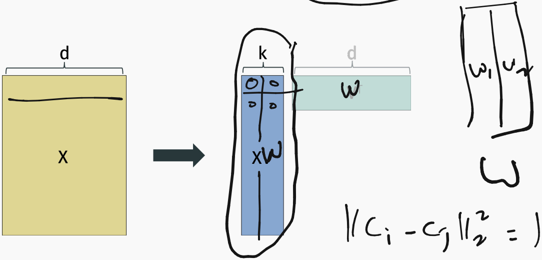
$= WW^T x_1 \rightarrow$ column vector

$\underbrace{W^T W}_{I}$ in $WW^T WW^T x$

$= WW^T x$

LOW-RANK APPROXIMATION





$\|c_i - c_j\|_2^2 = \|\tilde{x}_i - \tilde{x}_j\|_2^2$

$\mathbf{C} = \underline{\mathbf{XW}}$ can be used as a compressed version of data matrix $\mathbf{X}$.

Let $\mathbf{C} = \mathbf{X}\mathbf{W}$. We have that:

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2 \approx \|\overbrace{\mathbf{x}_i^T\mathbf{W}\mathbf{W}^T}^{\tilde{x}_i} - \overbrace{\mathbf{x}_j^T\mathbf{W}\mathbf{W}^T}^{\tilde{x}_j}\|_2 = \|\mathbf{c}_i - \mathbf{c}_j\|_2$$

Similarly, we expect that:

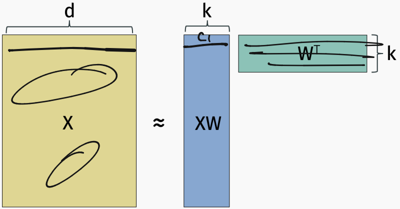
$$\|x_i - x_j\|_2 \approx \|c_i - c_j\|_2$$

- $\|\mathbf{x}_i\|_2 \approx \|\mathbf{c}_i\|_2$
- $\langle \mathbf{x}_i, \mathbf{x}_j \rangle \approx \langle \mathbf{c}_i, \mathbf{c}_j \rangle$
- etc.

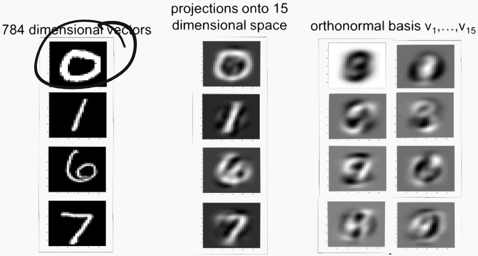
$c_i, c_j \in [\ ]\}k \qquad x_i, x_j \in [\ ]\}d$

How does this compare to Johnson-Lindenstrauss projection?

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.



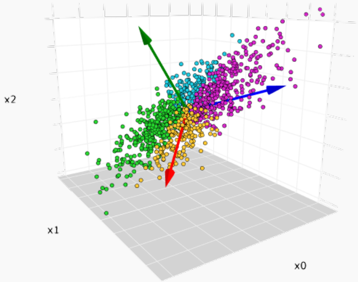
Colinearity/correlation of data features leads to a low-rank data matrix.

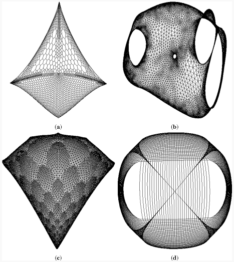
	bedrooms	bathrooms	sq.ft.	floors	list price	sale price
home 1	2	2	1800	2	200,000	195,000
home 2	4	2.5	2700	1	300,000	310,000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
home n	5	3.5	3600	3	450,000	450,000

APPLICATIONS OF LOW-RANK APPROXIMATION

Fact that $\|\mathbf{x}_i - \mathbf{x}_j\|_2 \approx \|\mathbf{x}_i^T\mathbf{W}\mathbf{W}^T - \mathbf{x}_j^T\mathbf{W}\mathbf{W}^T\|_2 = \|\mathbf{c}_i - \mathbf{c}_j\|_2$ leads to lots of applications.

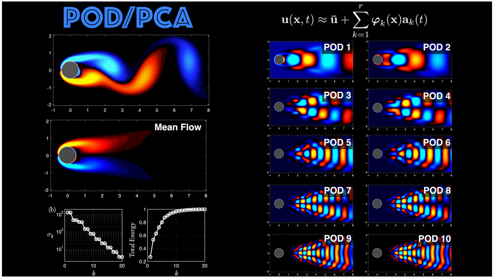
- Data compression. E.g. used in state-of-the-art data dependence methods for nearest neighbor search.
- Data visualization when $k = 2$ or 3.





- Entity embeddings (next lecture).

- Reduced order modeling for solving physical equations.

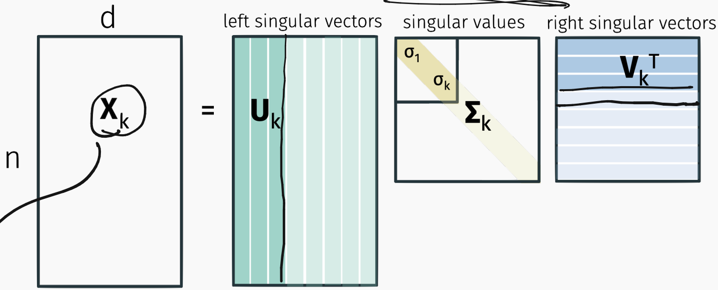


- Constructing preconditioners in optimization.
- Many more.

$X_k = U_k U_k^T X$ $\quad O(nd^2)$

Can find the best projection from the singular value decomposition.

$X_k = X V_k V_k^T$ $\quad O(ndk)$



rank

$$V_k = \underset{\text{orthogonal } W \in \mathbb{R}^{d \times k}}{\arg\min} \|X - XWW^T\|_F^2$$

Claim: $X_k = \underline{U_k \Sigma_k} V_k^T = \underline{X V_k} V_k^T$.

$$X V_k \underline{V_k}^T = U_k \Sigma_k \underline{V_k}^T$$

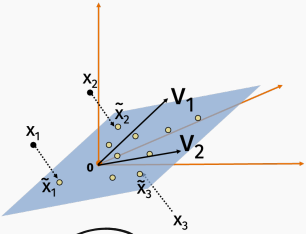
$$X V_k = U_k \Sigma_k$$

$$\underline{U} \Sigma \underbrace{V^T V_k} = \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{bmatrix} \to \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \\ & 0 & \end{bmatrix}$$

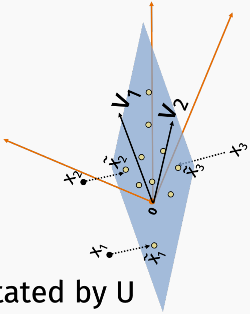
$U \, \Sigma_k \, U_k^T = U_k \Sigma_k V_k^T$

Claim 1:

$$\underset{\text{rank } k \ \mathbf{B}}{\arg\min} \|\mathbf{X} - \mathbf{B}\|_F^2 = \mathbf{U} \cdot \underset{\text{rank } k \ \mathbf{B}}{\arg\min} \|\mathbf{\Sigma} \mathbf{V}^T - \mathbf{B}\|_F^2$$



ΣV^T



Rotated by U on the left

$V \Sigma U^T$

$\underline{B^T = V_k \Sigma_k}$

Claim 2:

$B = \Sigma_k V_k^T$

$\begin{bmatrix} V_k & 0 \end{bmatrix}$

$$\underset{\text{rank } k \ \mathbf{B}}{\arg\min} \|\mathbf{\Sigma}\mathbf{V}^T - \mathbf{B}\|_F^2 = \underset{\text{rank } k \ \mathbf{B}}{\arg\min} \|\mathbf{V}\mathbf{\Sigma} - \mathbf{B}^T\|_F^2$$

Claim 3:

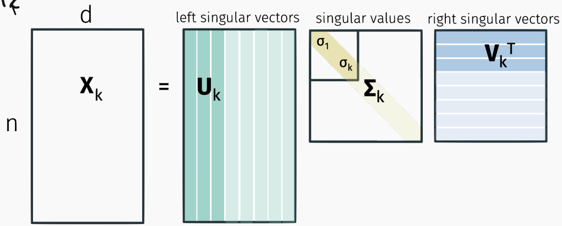
$\|V^T(V\Sigma - B^T)\|_F^2$

$$\underset{\text{rank } k \ \mathbf{B}}{\arg\min} \|\mathbf{V}\mathbf{\Sigma} - \mathbf{B}^T\|_F^2 = \underset{\text{rank } k \ \mathbf{B}}{\arg\min} \|\mathbf{\Sigma} - \mathbf{V}^T\mathbf{B}^T\|_F^2$$

Chose $\mathbf{B}^T$ so that $\mathbf{V}^T\mathbf{B}^T = \mathbf{\Sigma}_k$.

$$\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} - \begin{bmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_k & & \\ & & & 0 & \\ & & & & \ddots \ 0 \end{bmatrix} = \begin{bmatrix} 0 & & & & \\ & 0 & & & \\ & & \sigma_{k+1} & & \\ & & & \ddots & \\ & & & & \sigma_n \end{bmatrix}$$

$$\|X - XGG^T\|_F \geq \|X - XU_kU_k^T\|_F$$



Observation 1:

$$\arg\min_{W \in \mathbb{R}^{d\times k}} \|X - XWW^T\|_F^2 = \left(\arg\max_{\underline{W \in \mathbb{R}^{d\times k}}} \|XWW^T\|_F^2\right)$$

Follows from fact that for <u>all</u> orthogonal <u>W</u>:

$$\underline{\|X - XWW^T\|_F^2} = \underline{\|X\|_F^2 - \|XWW^T\|_F^2}$$

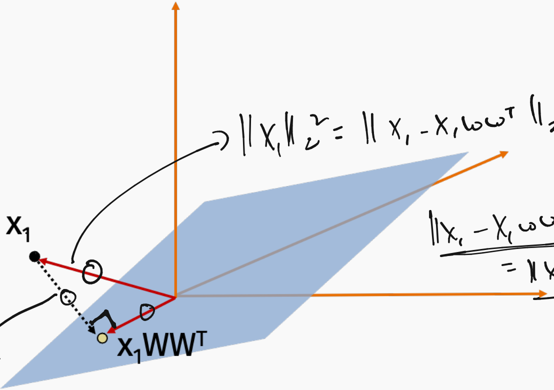
Pythagorean Theorem

Claim:

$$\sum_{i=1}^{n} \|x_i - x_i w w^T\|_2^2$$

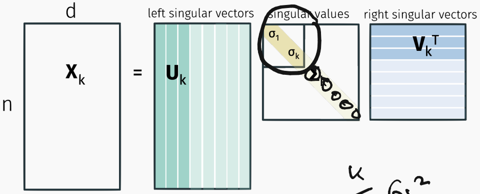
$$\|X - XWW^T\|_F^2 = \|X\|_F^2 - \|XWW^T\|_F^2$$

$\|x_1\|_2^2 = \|x_1 - x_1 w w^T\|_2^2 + \|x_1 w w^T\|_2^2$



$\|x_1 - x_1 w w^T\|_2^2 = \|x_1\|_2^2 - \|x_1 w w^T\|_2^2$

$x_1 - x_1 w w^T$



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

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:

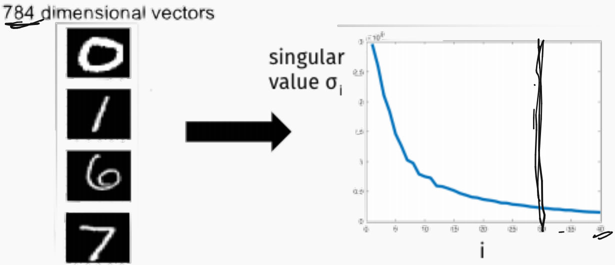
$$E_k = \sum_{i=k+1}^{d} \sigma_i^2.$$

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

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:

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

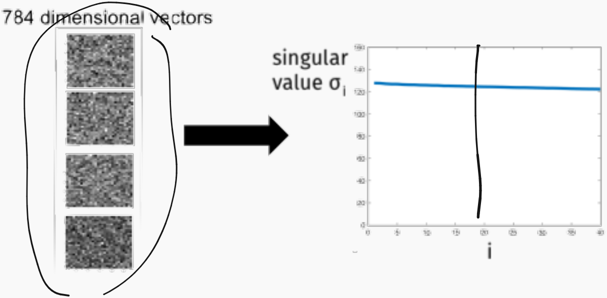
Can immediately get a sense of "how low-rank" a matrix is from it's spectrum:



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:

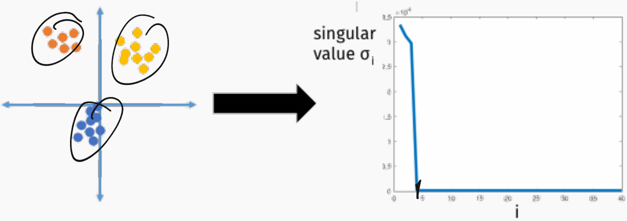


SPECTRAL PLOTS

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**:

$X = U\Sigma V^T$

$X = U\Sigma$

- Compute $\mathbf{X}^T\mathbf{X}$.
- Find eigendecomposition $\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T = \mathbf{X}^T\mathbf{X}$ using e.g. QR algorithm.
- Compute $\mathbf{L} = \mathbf{X}\mathbf{V}$. Set $\sigma_i = \|\mathbf{L}_i\|_2$ and $\mathbf{U}_i = \mathbf{L}_i / \|\mathbf{L}_i\|_2$.

$X \to n \times d$

Total runtime $\approx O(nd^2) + d^3 \log\log(1/\epsilon)$

- 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 $k = 1$?

$\|z - v_1\|_2^2 \le \epsilon$

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

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

$O(nd)$ $O(nd)$

$X^T(Xz^{(i-1)})$

Power method:

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

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

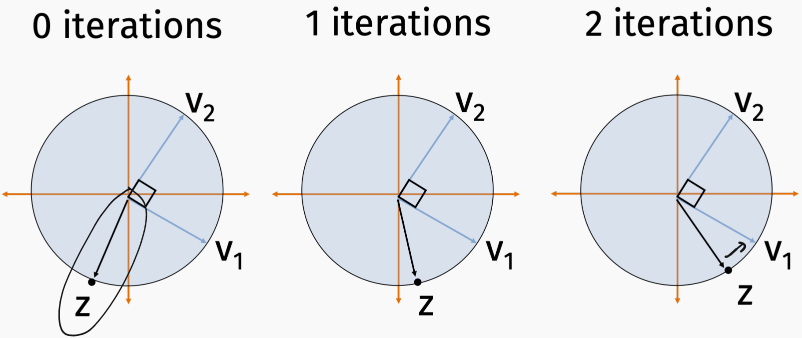
2nd of

z =

d

n

$O(ndT)$



$z = v_1$

$$\frac{X^T X z}{\|X^T X z\|_2} = \frac{\lambda_1(X^T X) \cdot v_1}{\|X^T X z\|_2}$$

$c \cdot v_1$

$\sigma_1 = \sigma_2 : \gamma = 0$

$\sigma_2 \ll \sigma_1 : \gamma$ larger

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

$< d$

$z^{(i)} = c \cdot X^T X z^{(i-1)}$

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

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

$V^T c^{(1)}$

$$\mathbf{z}^{(1)} = c_1^{(1)}\mathbf{v}_1 + c_2^{(1)}\mathbf{v}_2 + \ldots + c_d^{(1)}\mathbf{v}_d$$

$$\vdots$$

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

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

Also: $\|\mathbf{c}^{(i)}\|_2^2 = \sum_{j=1}^{d}\left(c_j^{(i)}\right)^2 = 1.$

$c_1^{(i-1)} \dots c_d^{(i-1)}$

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

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

$z^{(i-1)}$

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

$c^{(i)} = \Sigma^2 c^{(i-1)}$

$c^{(i-1)} = V^T z^{(i-1)}$

$c^{(i)} = V^T z^{(i)} = \underbrace{V^T X^T X z^{(i-1)}} \cdot \frac{1}{n_i}$

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

$\cancel{V^T V} \Sigma^2 \underbrace{V^T z^{(i-1)}}_{c^{(i-1)}}$

$\sigma_1^2 > \sigma_2^2 > \sigma_3^2 > \ldots$

Claim: After T updates:

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

≈ 0

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

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 $\left|\frac{\alpha_j}{\alpha_1}\right| \leq \sqrt{\frac{\epsilon}{d}}$ then:

$\|z^{(T)} - v_1\|_2^2 \leq 2\varepsilon$

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

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

$\leq \alpha_1^2 \cdot \frac{\varepsilon}{d} \leq \frac{\varepsilon}{d}$

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

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

$\geq (1-\varepsilon)$

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

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

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

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

Assumption: Starting coefficients are all roughly equal.

For all j $\qquad O(1/d^{1.5}) \leq \left|c_j^{(0)}\right| \leq 1.$

This is a very loose bound, but it's all that we will need. We will prove shortly that it holds with probability 99/100.

$$\frac{|\alpha_j|}{|\alpha_1|} = \frac{\sigma_j^{2T}}{\sigma_1^{2T}} \cdot \frac{|c_j^{(0)}|}{|c_1^{(0)}|} \leq \left(\frac{\sigma_j}{\sigma_1}\right)^{2T} \cdot d^{1.5} \leq \left(\frac{\sigma_2}{\sigma_1}\right)^{2T} \cdot d^{1.5}$$

≤ 1

$\geq 1/d^{1.5}$

$$\leq (1-\gamma)^{2T} \cdot d^{1.5} \leq \sqrt{\frac{\epsilon}{d}}$$

Need $T = \frac{\log(d/\epsilon)}{\gamma}$

$$\left((1-\gamma)^{1/\gamma}\right)^{\log(d/\epsilon)} \to \left(\frac{1}{e}\right)^{\log(d/\epsilon)} \leq \sqrt{\frac{\epsilon}{d}}$$

Need to prove: Starting coefficients are all roughly equal.

$$\text{For all } j \qquad O(1/d^{1.5}) \leq |c_j^{(0)}| \leq 1$$

$\frac{z^0}{\|z^0\|_2}$

with probability 99/100. **Prove using Gaussian (anti)-concentration.**

Right hand side is immediate from fact that $\sum_j (c_j^{(0)})^2 = 1$.

To show the left hand side we first use rotational invariance of Gaussian:

$$\mathbf{c}^{(0)} = \frac{\mathbf{V}^T\mathbf{z}^{(0)}}{\|\mathbf{z}^{(0)}\|_2} = \frac{\mathbf{V}^T\mathbf{z}^{(0)}}{\|\mathbf{V}^T\mathbf{z}^{(0)}\|_2} \sim \frac{\mathbf{g}}{\|\mathbf{g}\|_2},$$

where $\mathbf{g} \sim \mathcal{N}(0,1)^d$.

Need to show that with high probability, every entry of $\frac{\mathbf{g}}{\|\mathbf{g}\|_2} \geq c \cdot \frac{1}{d^{1.5}}$.

$\|g\|_2 \leq O(\sqrt{d})$

Part 1: With probablility 999/100,

$$\|\mathbf{g}\|_2^2 \leq \underline{2d}$$

$= \sum g_i^2$

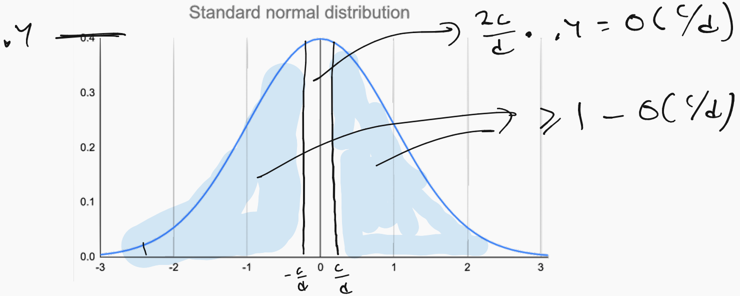
$\underline{\mathbb{E}[\|g\|_2^2]} = \sum \mathbb{E}[g_i^2] = \underline{d}$

Need to show that with high probability, the magnitude of every entry of $\underline{\mathbf{g}} \geq \underbrace{c \cdot \frac{1}{d}}$.

$1 - c$

Part 2: With probablility $1 - c/d$,

for any i, $\cancel{\min_i}$ $|g_i| \geq O\left(\frac{c}{d}\right)$.



Applying union bound completes the result.

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: $O(4d^2)$

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

The method truly won't converge if γ is very small. Consider extreme case when $\gamma = 0$.

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

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:*

$\cancel{\|z - v_1\|^2}$

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

Intuition: For a good low-rank approximation, we don't actually need to converge to $\mathbf{v}_1$ if σ_1 and σ_2 are the same or very close. Would suffice to return either $\mathbf{v}_1$ or $\mathbf{v}_2$, or some linear combination of the two.

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

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

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

$\frac{\log d/\varepsilon}{\varepsilon}$

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:

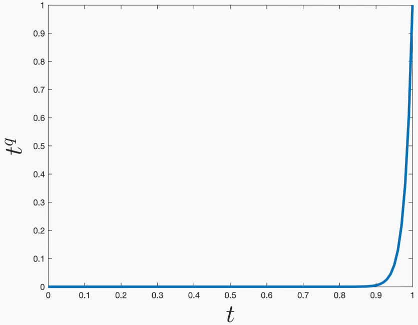
$$\|\mathbf{X} - \mathbf{X}\mathbf{Z}\mathbf{Z}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{X} - \mathbf{X}\mathbf{V_k}\mathbf{V_k}^T\|_F^2.$$

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

$\frac{\log(d/\varepsilon)}{\gamma}$ $\frac{\log(d/\varepsilon)}{\sqrt{\gamma}}$

$X^TX \, X^TX \ldots X^TX$

$$\underline{z^{(q)}} = c \cdot \underline{(X^TX)}^q \cdot g$$



$$z^{(q)} = c \cdot \left[c_1 \cdot \sigma_1^{2q} v_1 + c_2 \cdot \sigma_2^{2q} v_2 + \ldots + c_n \cdot \sigma_n^{2q} v_n\right]$$

$(X^T X)^{q-1} g$

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

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 $\|\mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T\|_F^2$.

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

Lanczos method:

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

2nd q

degree q

For a degree t polynomial p, let $\mathbf{v}_p = \frac{p(\mathbf{X}^T\mathbf{X})\mathbf{g}}{\|p(\mathbf{X}^T\mathbf{X})\mathbf{g}\|_2}$.

Power method returns:

$$(X^TX - 2(X^TX)^2 + 3(X^TX)^3 \ldots)g$$

$$\mathbf{v}_{X^q}.$$

Lanczos method returns $\mathbf{v}_{p^*}$ where:

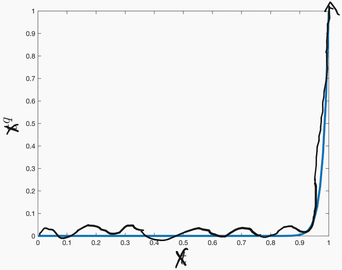
$$p^* = \underset{\text{degree } q\ p}{\arg\min} \|\mathbf{X} - \mathbf{X}\mathbf{v}_p\mathbf{v}_p^T\|_F^2.$$

$$c_0 g + c_1 X^TX g + \ldots + c_q (X^TX)^q g$$

$p(X^TX)\, g$ for deg q poly p.

$\Delta = \frac{\mathrm{poly}(\epsilon)}{\mathrm{poly}(d)}$

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



$q = \sqrt{\frac{\log(d/\epsilon)}{\gamma}}$

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

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

Again 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.

- Block Krylov methods

$\mathrm{orth}(p(X^TX)\cdot G)$ $\downarrow$ approx to $\underline{V_k}$.

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

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

$\underline{d(k+p)}$

$\sqrt{\gamma}$ $\underline{k+p \text{ vectors}}$

$\frac{\lambda_k - \lambda_{k+1}}{\lambda_k}$ $\frac{\lambda_k - \lambda_{k+p}}{\lambda_k}$