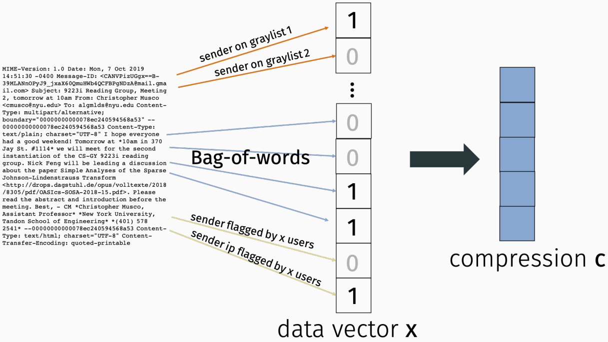
CS-GY 6763: Lecture 9
Low-rank approximation and singular value decomposition

NYU Tandon School of Engineering, Prof. Rajesh Jayaram

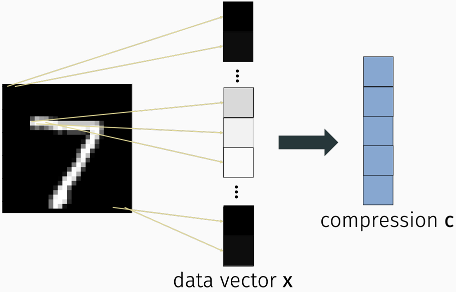
ADMINISTRATIVE

- Third reading group this Thursday at 4:30pm. Dennis and Jesse will present the paper: "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization"
- Hw 3 is due next Monday!
- Next two lectures: Spectral methods and Randomized Numerical Linear Algebra.
- Afterwards, Teal will teach a lecture (4/18) on Compressed Sensing.

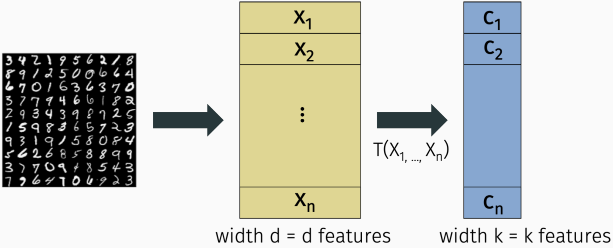
Return to data compression:



Return to data compression:



Main difference from randomized methods:



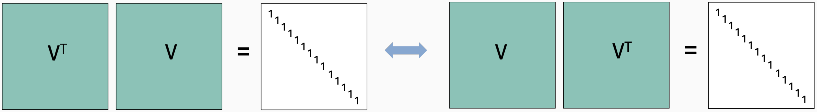
In this section, we will discuss data dependent transformations. Johnson-Lindenstrauss, MinHash, SimHash were all data oblivious.

Advantages of data independent methods:

Advantages of data dependent methods:

LINEAR ALGEBRA REMINDER

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

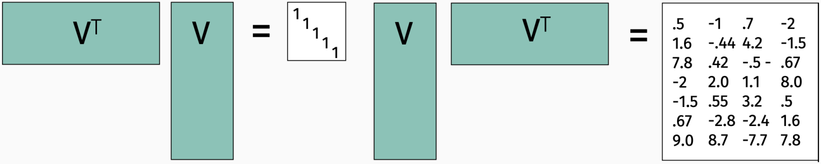


$$\mathbf{V}^T\mathbf{V} = \mathbf{I} = \mathbf{V}\mathbf{V}^T$$

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

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

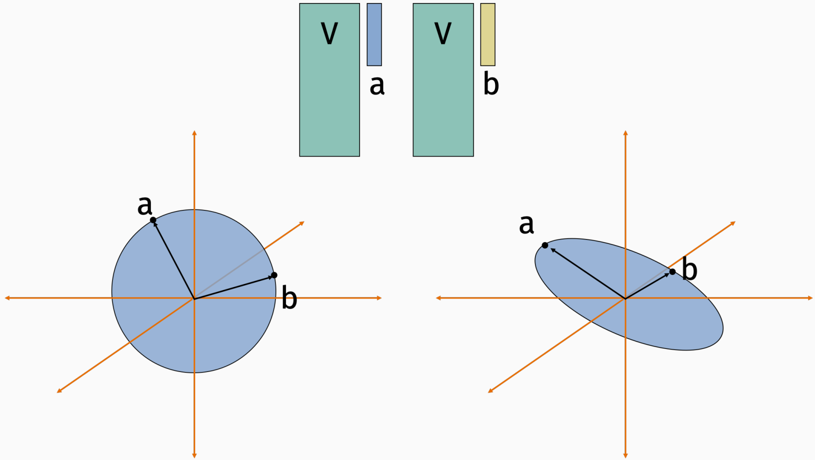
The same is not true for rectangular matrices:



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

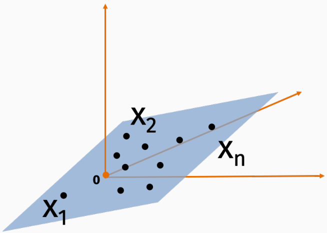
For any $\mathbf{x}$, $\|\mathbf{V}\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2$ but $\|\mathbf{V}^T\mathbf{x}\|_2^2 \neq \|\mathbf{x}\|_2^2$ in general.

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

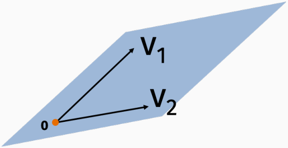


LOW-RANK DATA

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 $k < d$.



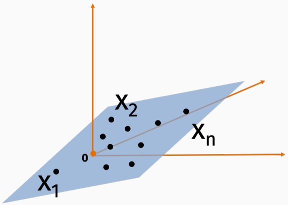
Let $\mathbf{v}_1, \ldots, \mathbf{v}_k$ be orthogonal unit vectors spanning S.

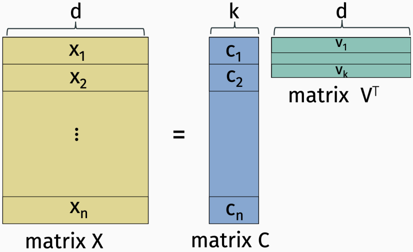


For all i, we can write:

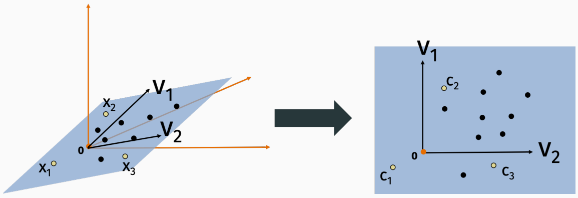
$$\mathbf{x}_i = c_{i,1}\mathbf{v}_1 + \ldots + c_{i,k}\mathbf{v}_k.$$

LOW-RANK DATA

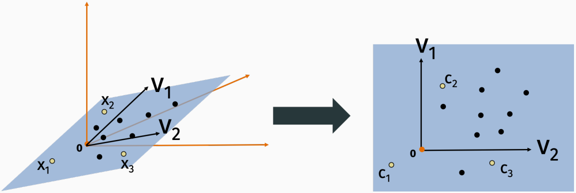




What are $\mathbf{c}_1, \ldots, \mathbf{c}_n$?



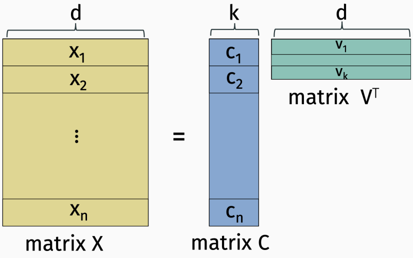
LOW-RANK DATA



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.

LOW-RANK DATA



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

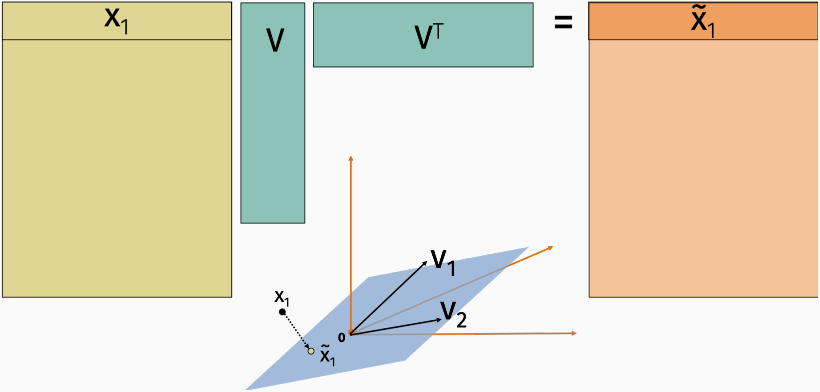
$$\mathbf{X} = \mathbf{C}\mathbf{V}^T \Rightarrow \mathbf{X}\mathbf{V} = \mathbf{C}\mathbf{V}^T\mathbf{V}$$

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

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

PROJECTION MATRICES

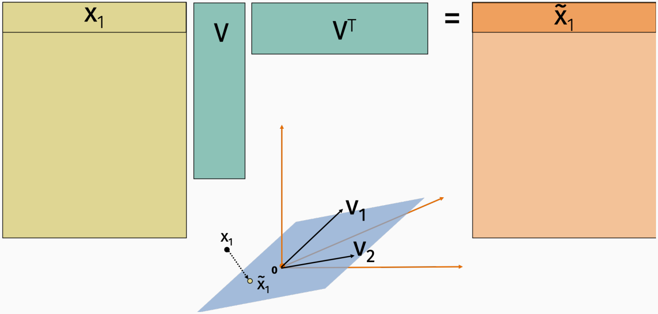
$\mathbf{V}\mathbf{V}^T$ is a symmetric projection matrix.



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

PROJECTION MATRICES

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



$\mathbf{x}_1^T\mathbf{V}\mathbf{V}^T$ is the projection of $\mathbf{x}_1^T$ onto the subspace.

By pythagorean theorem, $\|\mathbf{x}_1^T - \mathbf{x}_1^T\mathbf{V}\mathbf{V}^T\|_2^2 = \|\mathbf{x}_1^T\|_2^2 - \|\mathbf{x}_1^2\mathbf{V}\mathbf{V}^T\|_2^2$ and by apply to all rows, $\|\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2$.

LOW-RANK APPROXIMATION

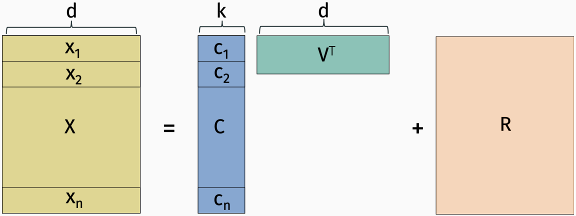
When $\mathbf{X}$'s rows lie close to a k dimensional subspace, we can still approximate

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

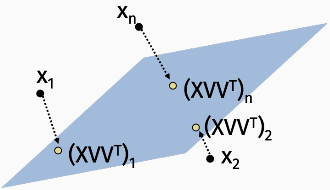
$\mathbf{X}\mathbf{V}\mathbf{V}^T$ is a low-rank approximation for $\mathbf{X}$.

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

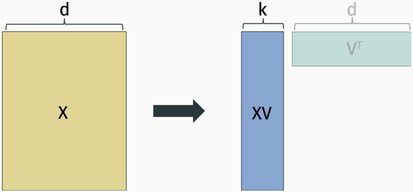
$$\mathbf{X}\mathbf{V}\mathbf{V}^T = \arg\min_{\mathbf{C}} \|\mathbf{X} - \mathbf{C}\mathbf{V}^T\|_F^2 = \sum_{i,j} (\mathbf{X}_{i,j} - (\mathbf{C}\mathbf{V}^T)_{i,j})^2.$$



LOW-RANK APPROXIMATION



$$\|\mathbf{x}_i - \mathbf{x}_j\|_2 \approx \|\mathbf{x}_i^T \mathbf{V}\mathbf{V}^T - \mathbf{x}_j^T \mathbf{V}\mathbf{V}^T\|_2 = \|\mathbf{x}_i^T \mathbf{V} - \mathbf{x}_j^T \mathbf{V}\|_2$$

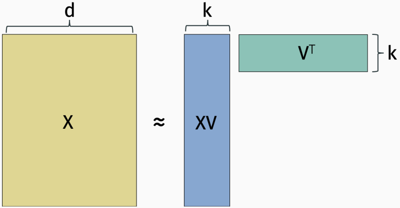


XV can be used as a compressed version of data matrix **X**.

WHY IS DATA APPROXIMATELY LOW-RANK?

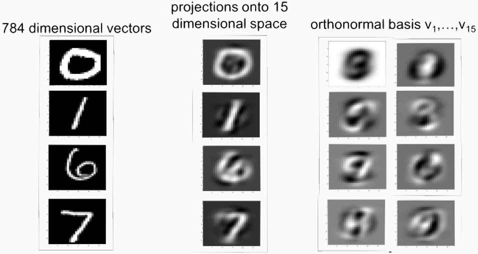
DUAL VIEW

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



ROW REDUNDANCY

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.



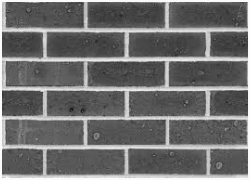
COLUMN REDUNDANCY

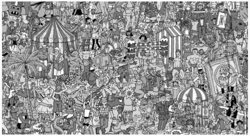
Colinearity/correlation of data features leads to a low-rank data matrix.

	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

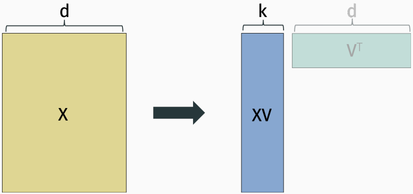
OTHER REASONS FOR LOW-RANK STRUCTURE

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





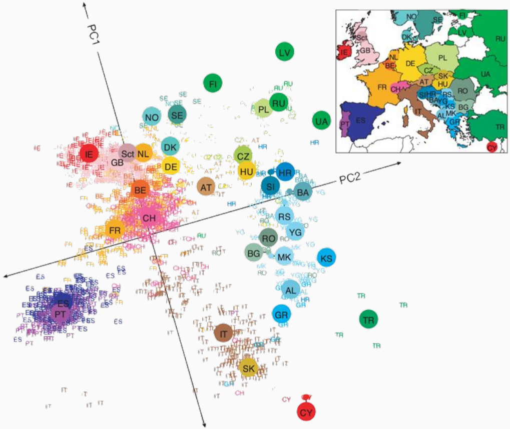
APPLICATIONS OF LOW-RANK APPROXIMATION



- $\mathbf{XV} \cdot \mathbf{V}^T$ 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 $k = 2, 3$.

APPLICATIONS OF LOW-RANK APPROXIMATION

“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.

COMPUTATIONAL QUESTION

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

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

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

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

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:

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

RANK 1 CASE

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

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

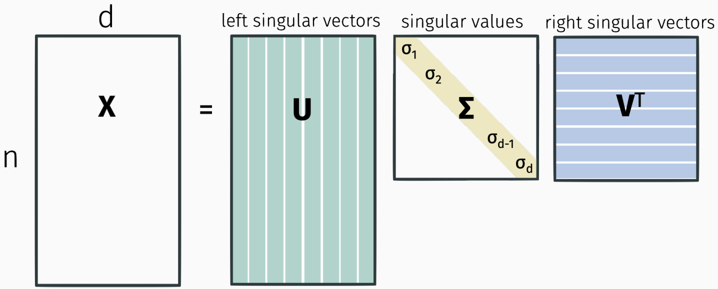
Choose $\mathbf{v}_1$ to be the top eigenvector of $\mathbf{X}^T\mathbf{X}$.

What about higher k?

SINGULAR VALUE DECOMPOSITION

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

Any matrix $\mathbf{X}$ can be written:



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

CONNECTION TO EIGENDECOMPOSITION

- $\mathbf{V}_k$'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 $\mathbf{X}$"
- $\sigma_1, \ldots, \sigma_k$ are the "top singular values". $\sigma_1, \ldots, \sigma_d$ 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.

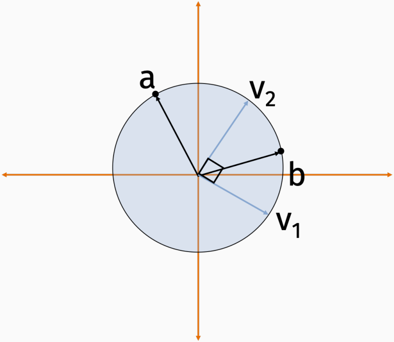
SINGULAR VALUE DECOMPOSITION

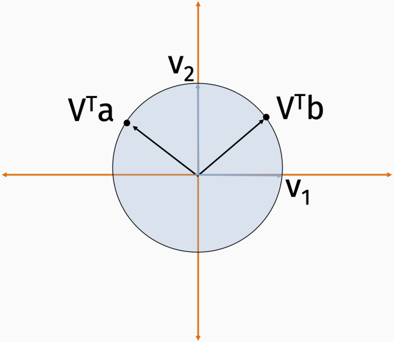
Important take away from singular value decomposition.

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:

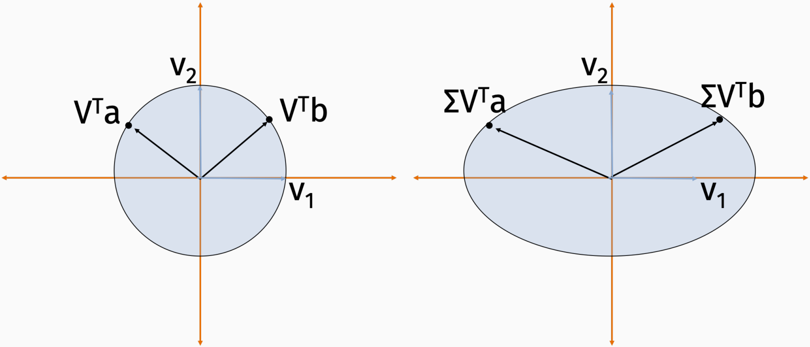
1. Rotate/reflect the vector (multiplication by to $\mathbf{V}^T$).
2. Scale the coordinates (multiplication by $\mathbf{\Sigma}$.
3. Rotate/reflect the vector again (multiplication by $\mathbf{U}$).

SINGULAR VALUE DECOMPOSITION: ROTATE/REFLECT

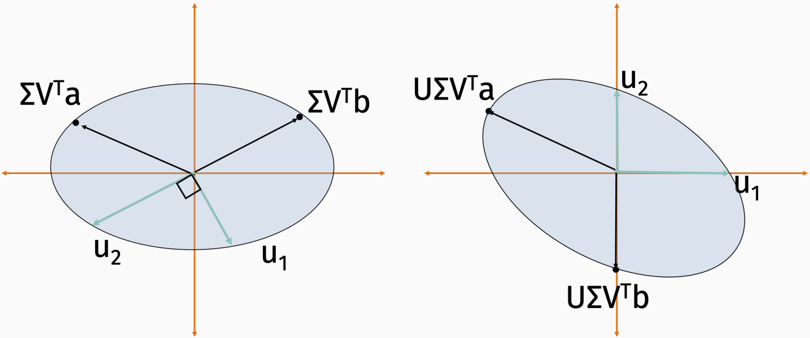




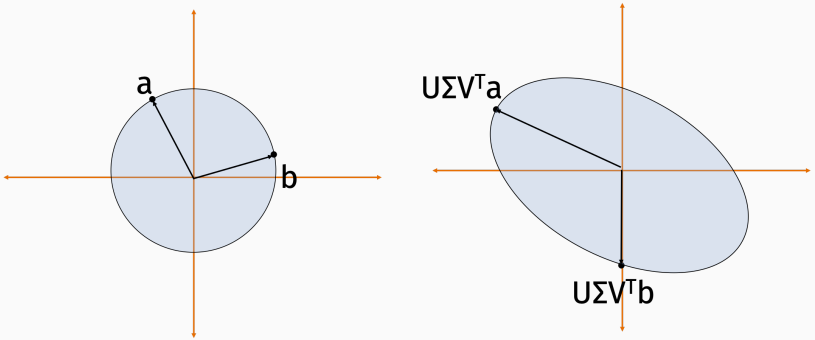
SINGULAR VALUE DECOMPOSITION: STRETCH



SINGULAR VALUE DECOMPOSITION: ROTATE/REFLECT

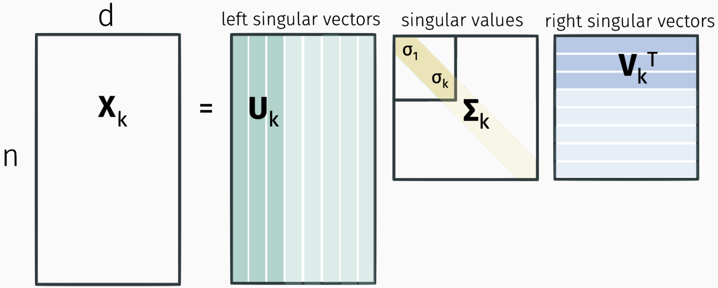


SINGULAR VALUE DECOMPOSITION



SINGULAR VALUE DECOMPOSITION

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



$$\mathbf{X}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T = \mathbf{U}_k \mathbf{U}_k^T \mathbf{X} = \mathbf{X} \mathbf{V}_k \mathbf{V}_k^T.$$

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

Theorem (Eckart–Young–Mirsky theorem)

Let $\mathbf{X} \in \mathbb{R}^{n \times k}$ *be any matrix, and let* $\mathbf{X}_k = \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^T$ *be the k-truncated SVD of* $\mathbf{A}$*. Then the best rank-k approximation to* $\mathbf{X}$ *is* $\mathbf{X}_k$*. Namely:*

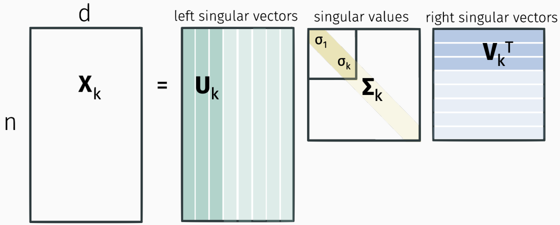
$$\begin{aligned}\min_{\text{rank-}k\ B} \|\mathbf{X} - \mathbf{B}\|_F^2 &= \|\mathbf{X} - \mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2 \\ &= \|\mathbf{X} - \mathbf{X}_k\|_F^2\end{aligned}$$

SINGULAR VALUE DECOMPOSITION

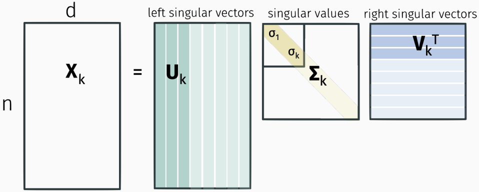
Connection to **Principal Component Analysis**:

- 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 $\mathbf{X}$. $\mathbf{V}$'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.



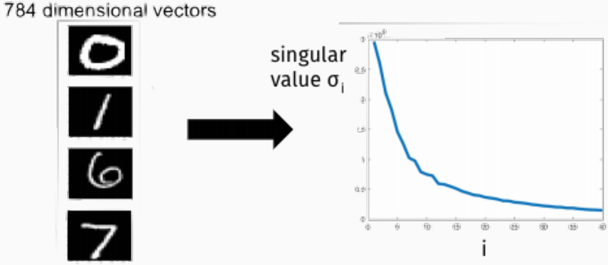
Observation 2: The optimal low-rank approximation error $E_k = \|\mathbf{X} - \mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2$ can be written:

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

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:

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

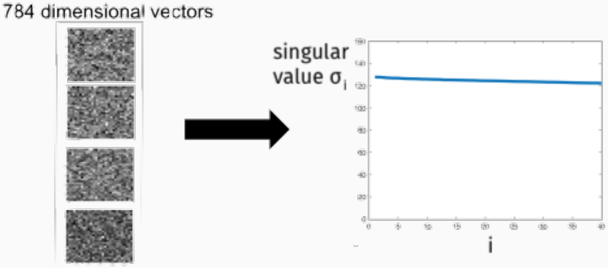
Can immediately get a sense of "how low-rank" a matrix is from it's spectrum:



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:

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

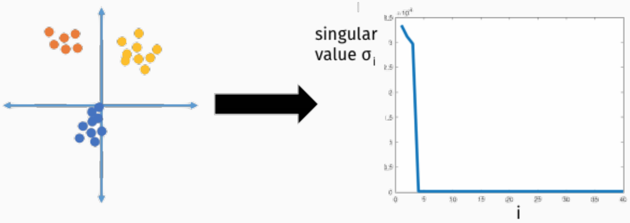
Can immediately get a sense of "how low-rank" a matrix is from it's spectrum:



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:



COMPUTING THE SVD

Suffices to compute right singular vectors $\mathbf{V}$:

- Compute $\mathbf{X}^T\mathbf{X}$.
- 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 ≈

COMPUTING THE SVD (FASTER)

- 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).

POWER METHOD

Today: What about when $k = 1$?

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

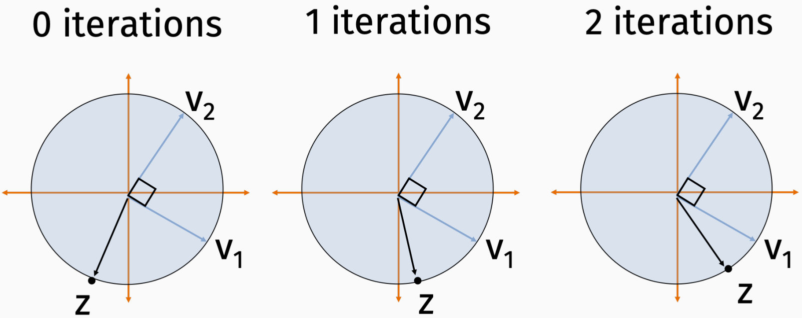
Input: $\mathbf{X} \in \mathbb{R}^{n \times d}$ with SVD $\mathbf{U\Sigma V}^T$.

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

POWER METHOD INTUITION



POWER METHOD FORMAL CONVERGENCE

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 of a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$. 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: $O\left(nd \cdot \frac{\log d/\epsilon}{\gamma}\right)$

Refined runtime: $O\left(\mathrm{nnz}(\mathbf{X}) \cdot \frac{\log d/\epsilon}{\gamma}\right)$, where $\mathrm{nnz}(\mathbf{X})$ is the number of non-zero entries in $\mathbf{X}$.

ONE STEP ANALYSIS OF POWER METHOD

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

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

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

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

ONE STEP ANALYSIS OF POWER METHOD

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

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

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

MULTI-STEP ANALYSIS OF POWER METHOD

Claim: After T updates:

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

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 $j \neq 1$.

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:

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

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

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

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

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

POWER METHOD FORMAL CONVERGENCE

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 that we will need. **Prove using Gaussian concentration.**

$$\frac{\alpha_j}{\alpha_1} = \frac{\sigma_j^{2T}}{\sigma_1^{2T}} \cdot \frac{c_j^{(0)}}{c_1^{(0)}} \leq$$

Need $T =$

POWER METHOD – NO GAP DEPENDENCE

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:

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

GENERALIZATIONS TO LARGER K

- 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:

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