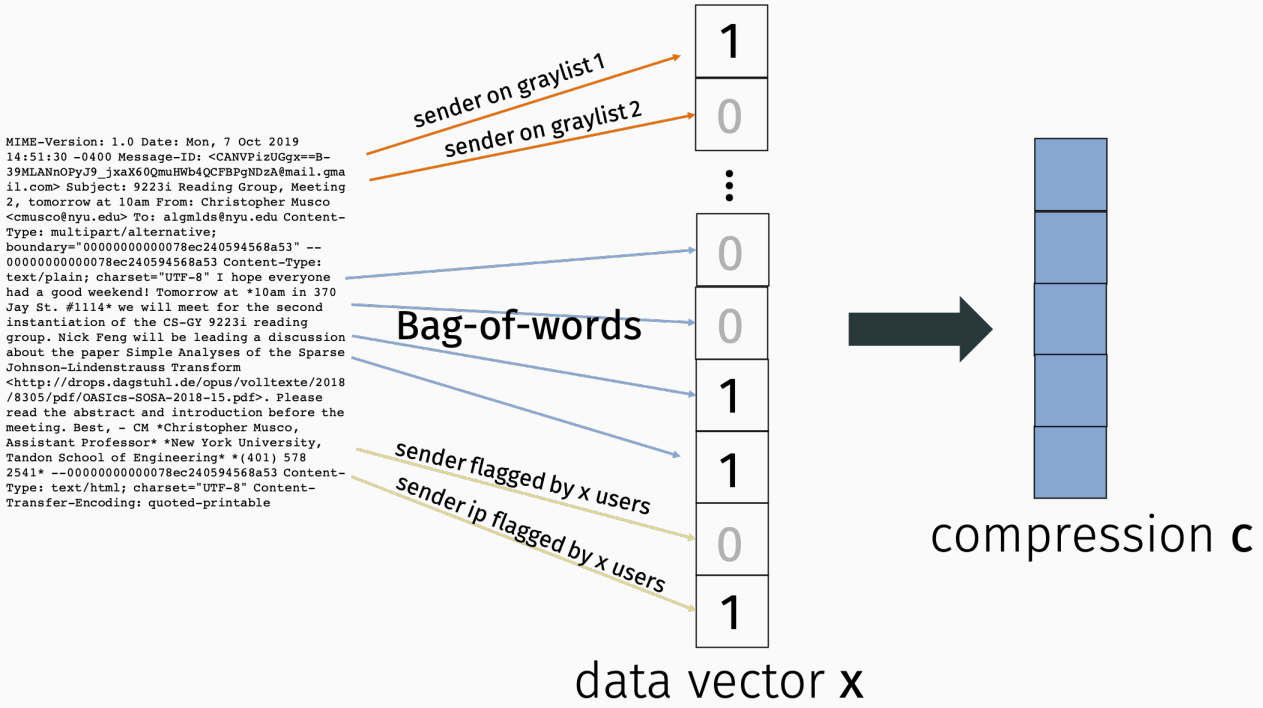
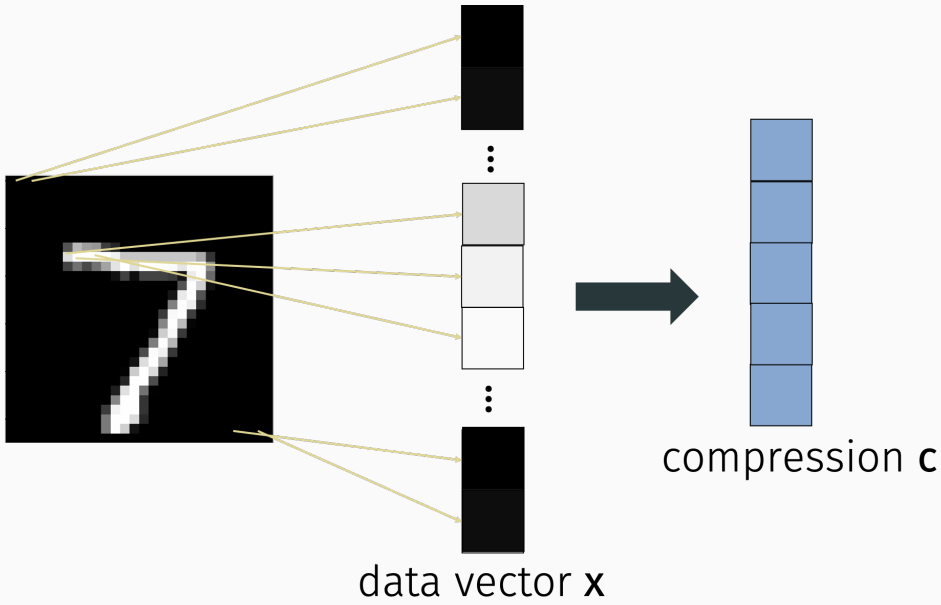
CS-GY 9223 I: Lecture 9
Low-rank approximation and singular value decomposition

NYU Tandon School of Engineering, Prof. Christopher Musco

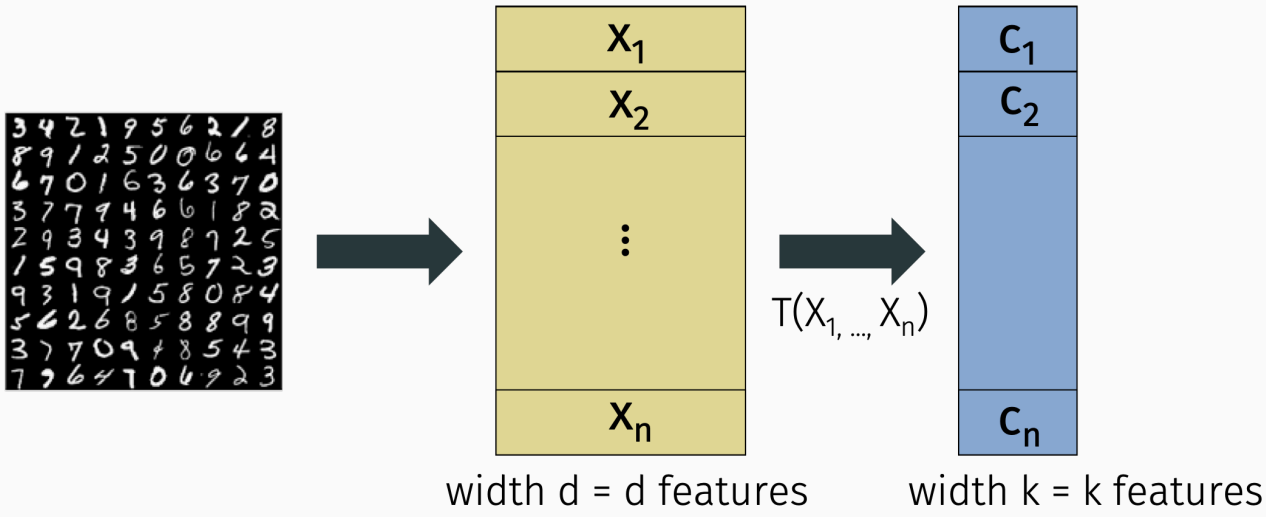
Return to data compression:



Return to data compression:



Main difference from randomized methods:



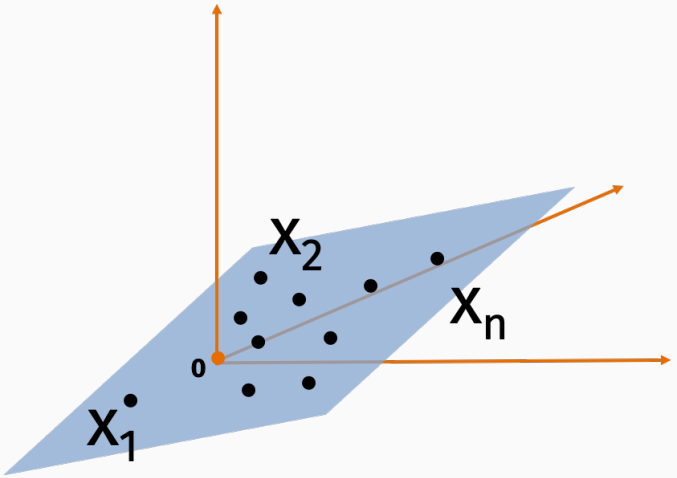
In this section, we will discuss data dependent data transformations. Johnson-Lindenstrauss, MinHash, SimHash were all data oblivious.

Advantages of data independent methods:

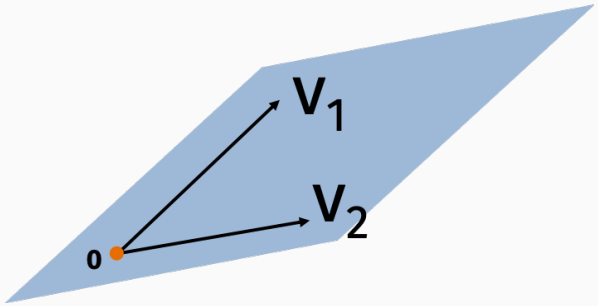
Advantages of data dependent methods:

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 $k < d$.



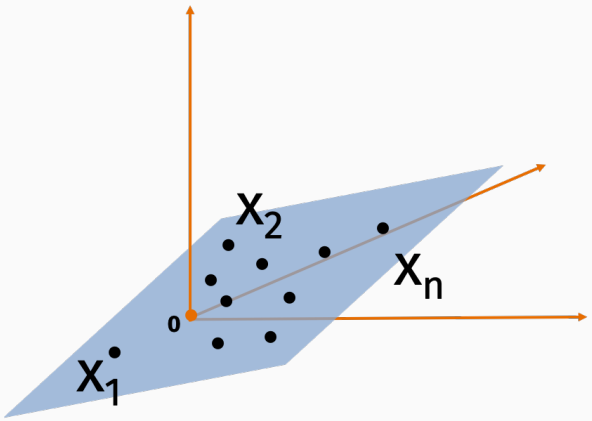
Let $\mathbf{v}_1, \ldots, \mathbf{v}_k$ be orthogonal unit vectors spanning S.

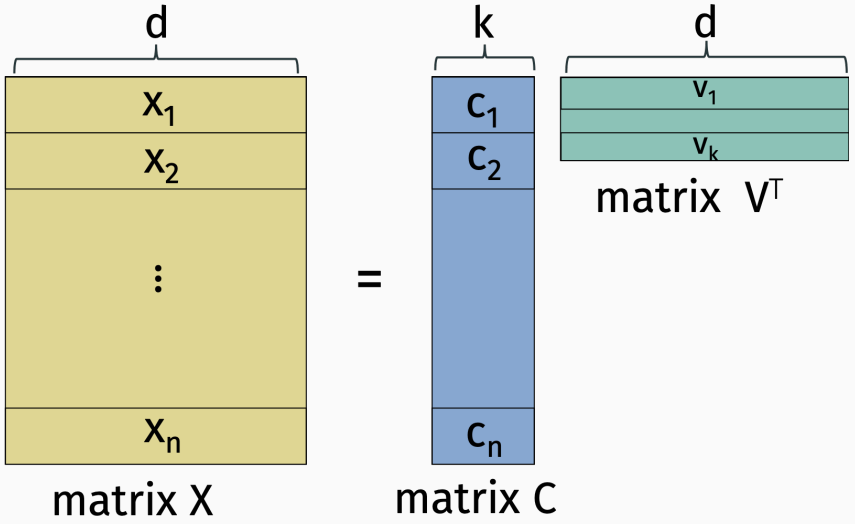


For all i, we can write:

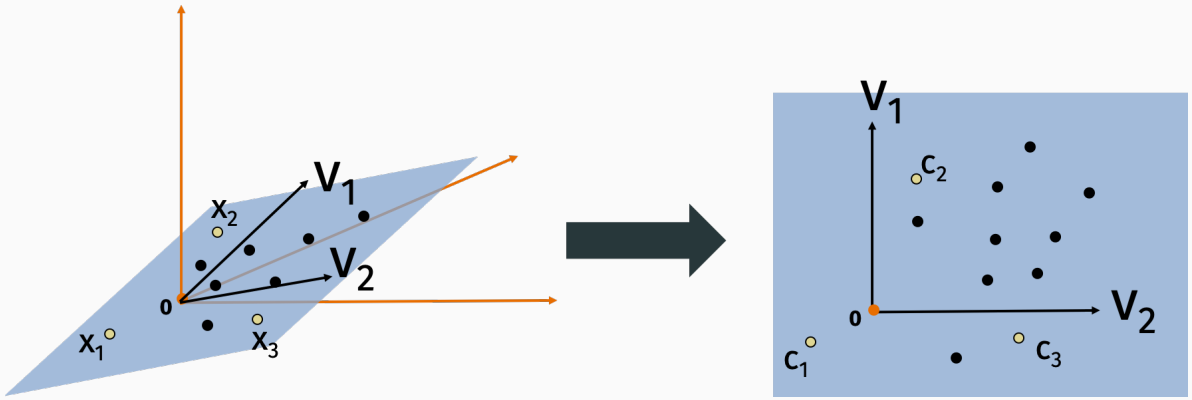
$$\mathbf{x}_i = c_{i,1}\mathbf{v}_1 + \ldots + c_{i,k}\mathbf{v}_k.$$

LOW-RANK DATA

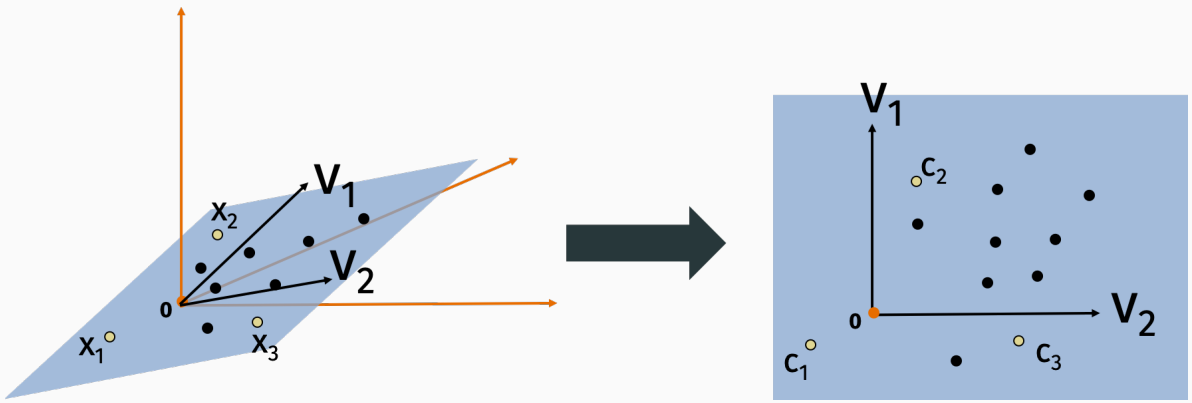




What are $c_1, \ldots, c_n$?



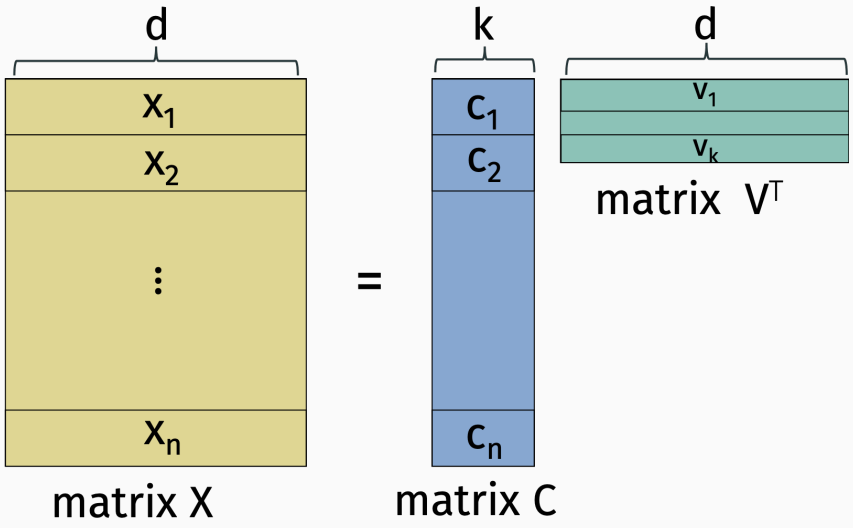
LOW-RANK DATA



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, $C = XV^T$:

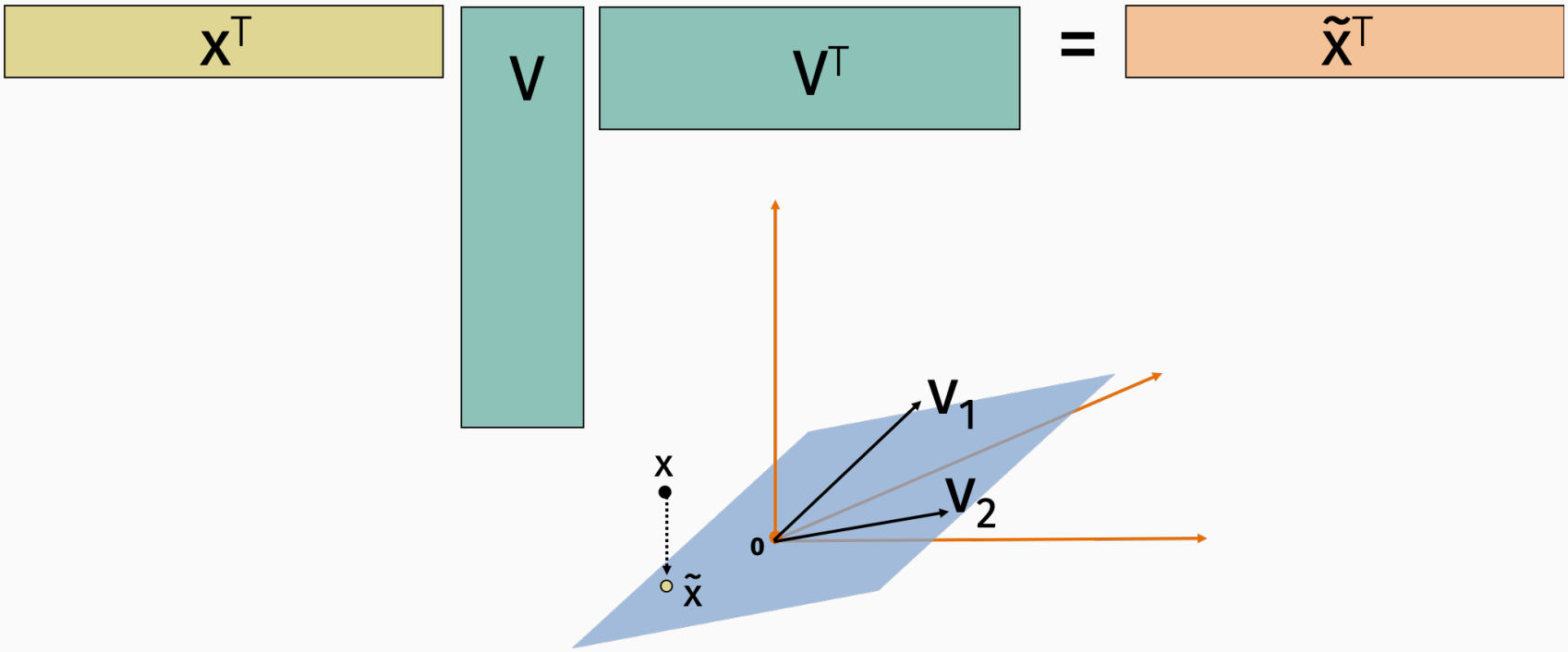
$$X = CV^T \Rightarrow XV = CV^TV$$

Since V's columns are an orthonormal basis, $V^TV = I$.

So $X = XVV^T$.

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

LOW-RANK APPROXIMATION

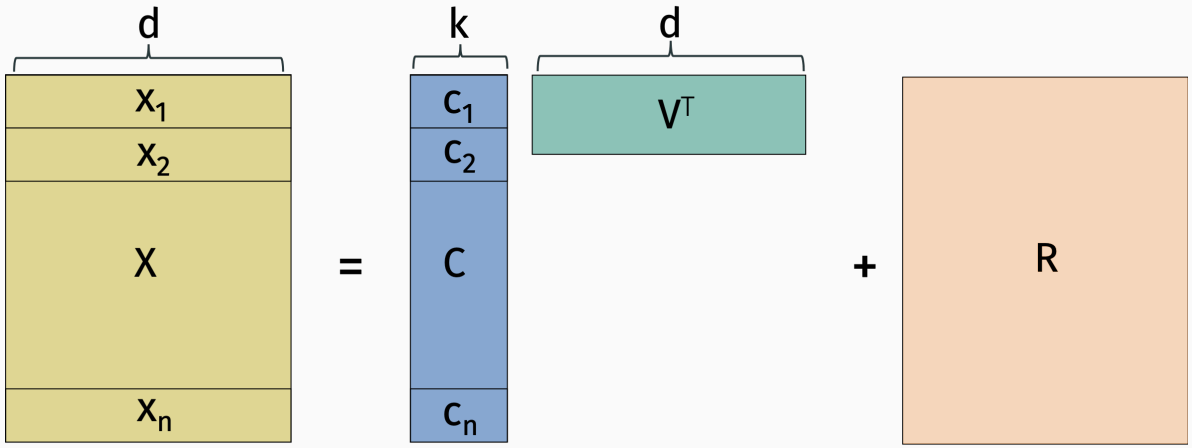
When $\mathbf{X}$'s rows lie close to a k dimensional subspace, we can still approximate

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

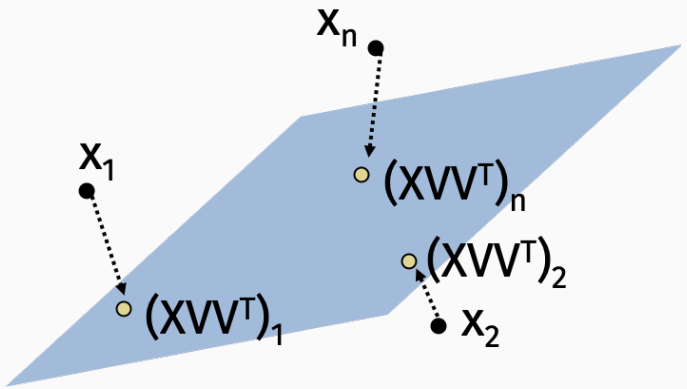
$\mathbf{X}\mathbf{V}\mathbf{V}^T$ is a low-rank approximation for $\mathbf{X}$.

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

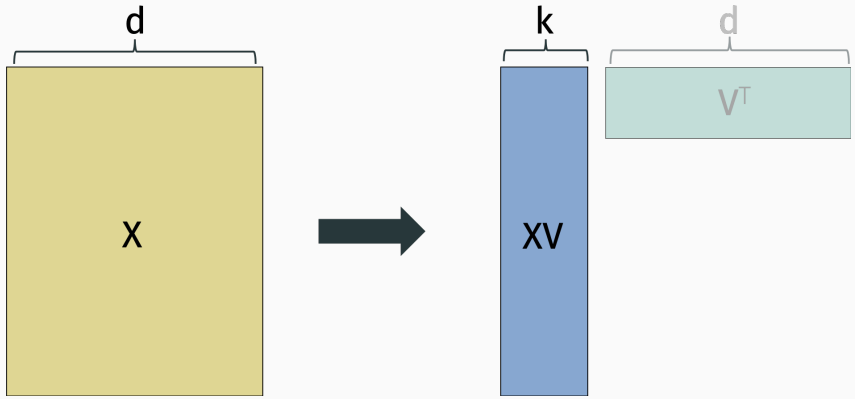
$$\mathbf{X}\mathbf{V}\mathbf{V}^T = \arg\min_{\mathbf{C}} \|\mathbf{X} - \mathbf{C}\mathbf{V}^T\|_F^2 = \sum_{i,j} (\mathbf{X}_{i,j} - (\mathbf{C}\mathbf{V}^T)_{i,j})^2.$$



LOW-RANK APPROXIMATION



$$\|x_i - x_j\|_2 \approx \|(XVV^T)_i - (XVV^T)_j\|_2 = \|(XV^T)_i - (XV^T)_j\|_2$$

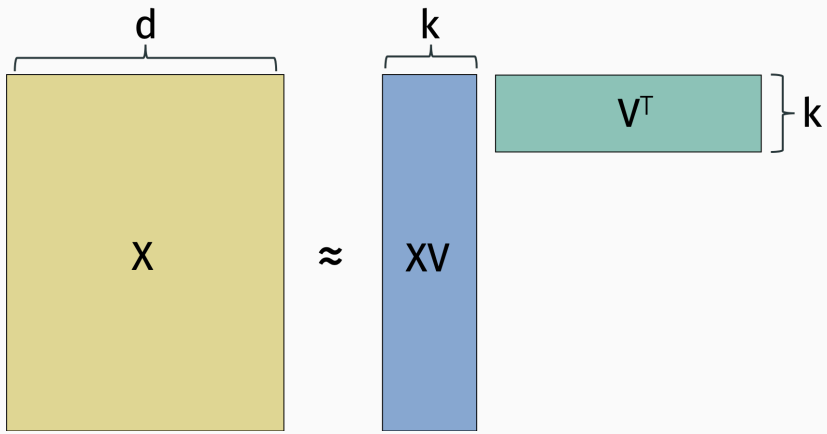


XV can be used as a compressed version of data matrix **X**.

Why is data approximately low-rank?

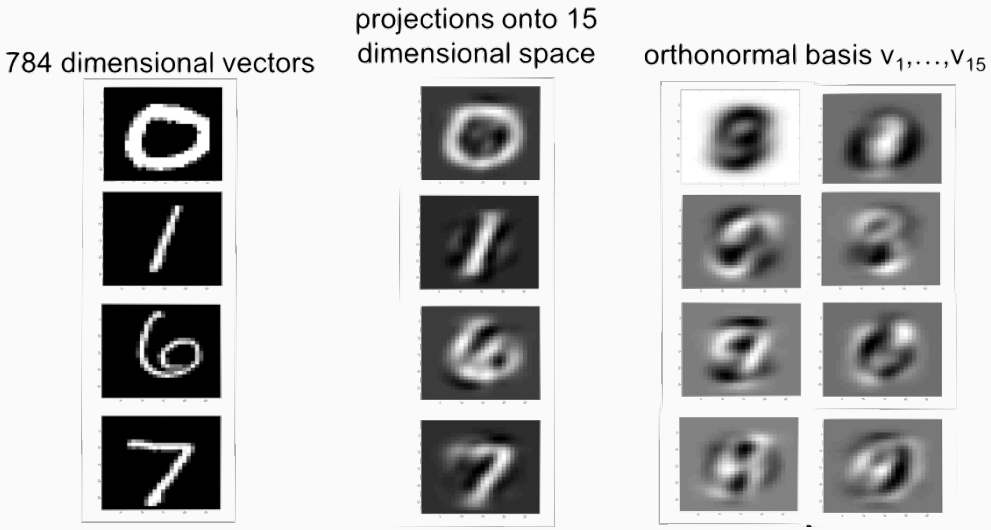
DUAL VIEW

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.



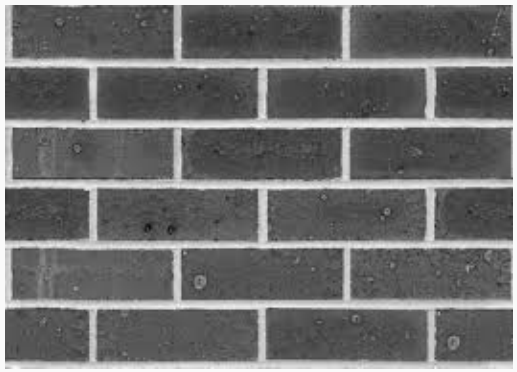
COLUMN REDUNDANCY

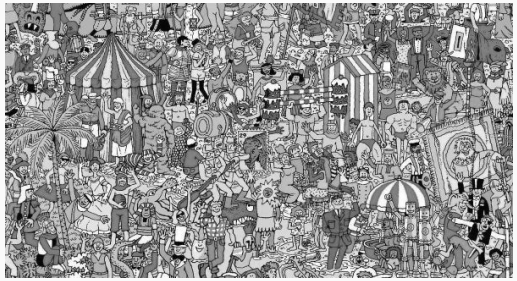
Colinearity/correlation of data features leads to a low-rank data matrix.

	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

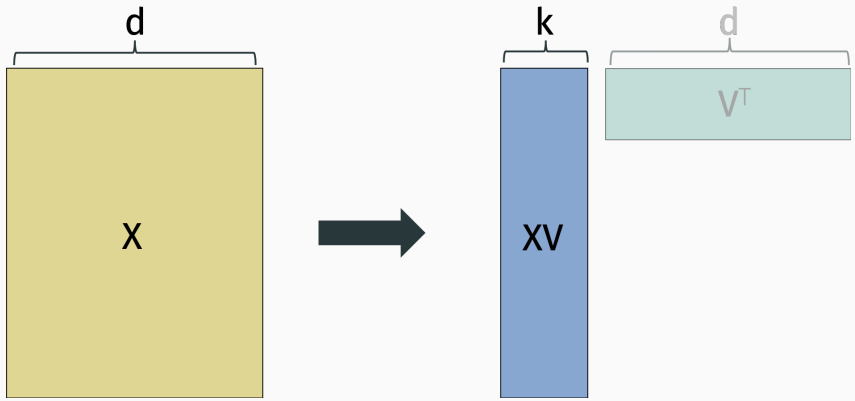
OTHER REASONS FOR LOW-RANK STRUCTURE

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



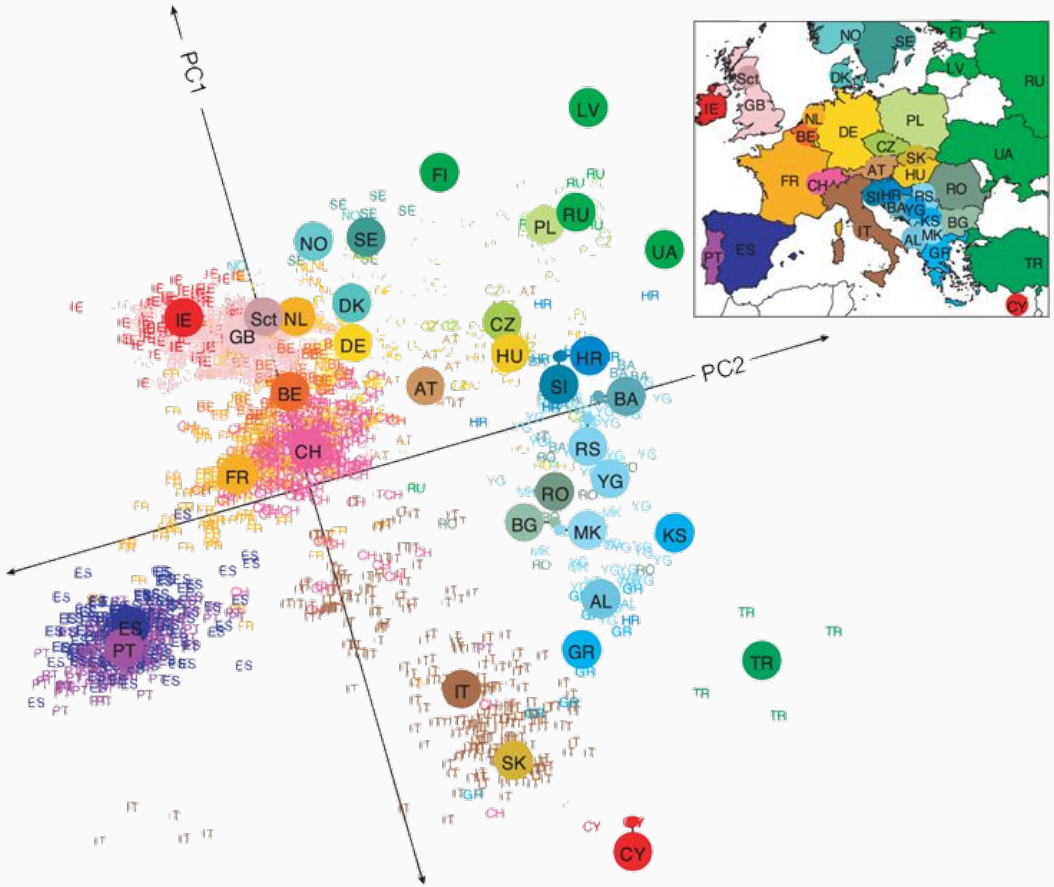


APPLICATIONS OF LOW-RANK APPROXIMATION



- $\mathbf{XV} \cdot \mathbf{V}^T$ 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 $k = 2, 3$.
- We will discuss many more next class.

"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 $\mathbf{V}$,

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

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:

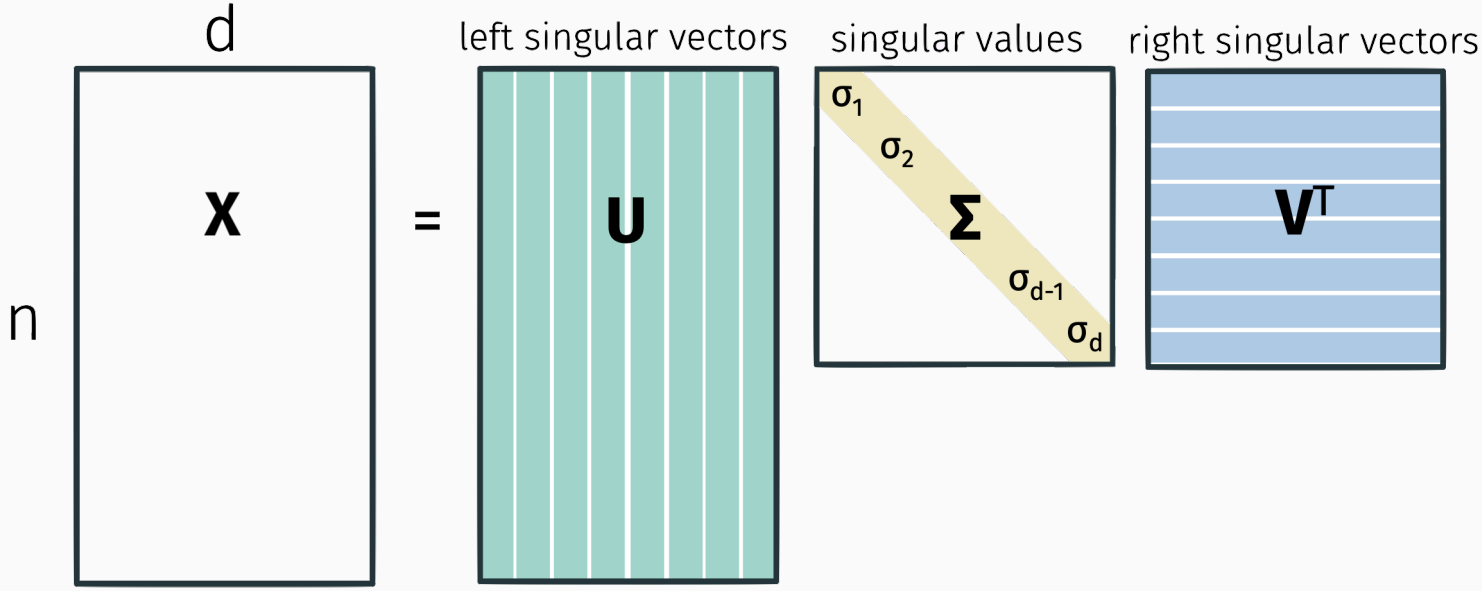
$$\|\mathbf{X}\mathbf{v}_1\mathbf{v}_1^T\|_F^2 = \|\mathbf{X}\mathbf{v}_1\|_F^2 = \|\mathbf{X}\mathbf{v}_1\|_2^2 = \mathbf{v}_1^T\mathbf{X}^T\mathbf{X}\mathbf{v}_1.$$

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

What about higher k?

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

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



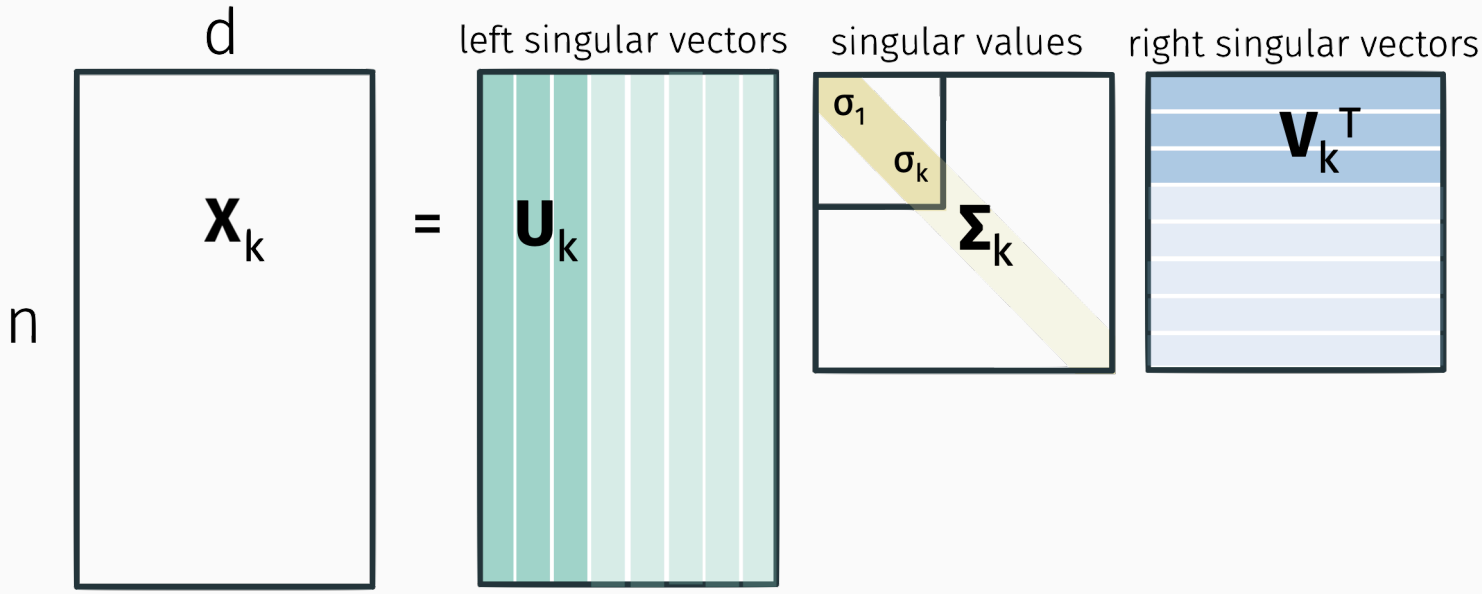
Where $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, and $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_d \geq 0$.

CONNECTION TO EIGENDECOMPOSITION

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

This can be checked directly:

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



$$X_k = U_k U_k^T X = X V_k V_k^T.$$

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

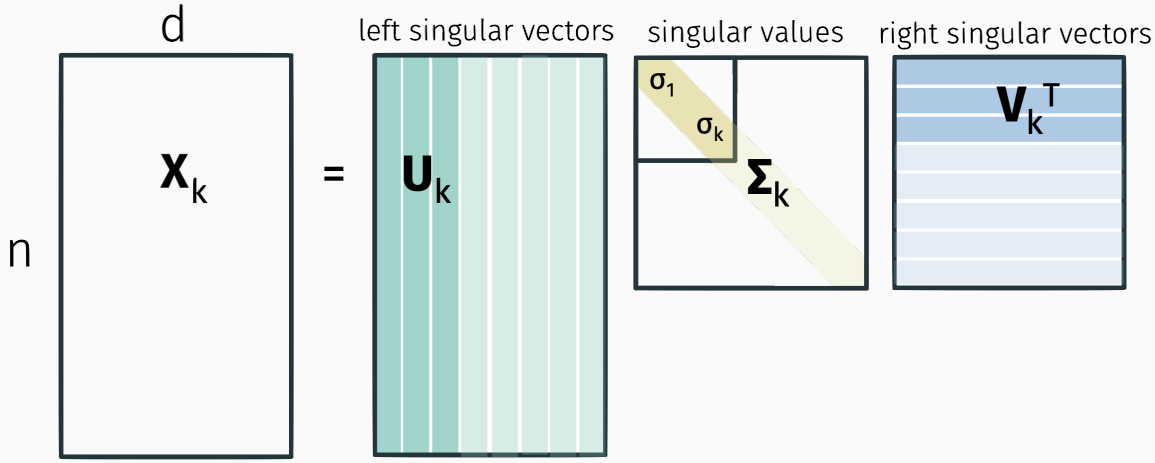
SINGULAR VALUE DECOMPOSITION

- $\mathbf{V}_k$'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).

Connection to Principal Component Analysis:

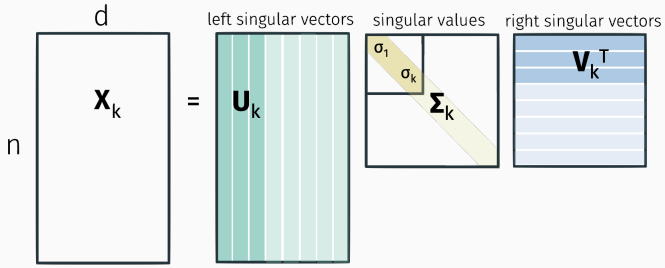
- Let $\bar{\mathbf{X}} = \mathbf{X} - \mathbf{1}\boldsymbol{\mu}^T$ where $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$. I.e. $\bar{\mathbf{X}}$ is obtained by mean centering $\mathbf{X}$'s rows.
- Let $\bar{\mathbf{U}}\bar{\mathbf{\Sigma}}\bar{\mathbf{V}}^T$ be the SVD of $\bar{\mathbf{X}}$. $\bar{\mathbf{U}}$'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 $\mathbf{XV}_k$ has orthogonal columns.

USEFUL OBSERVATIONS



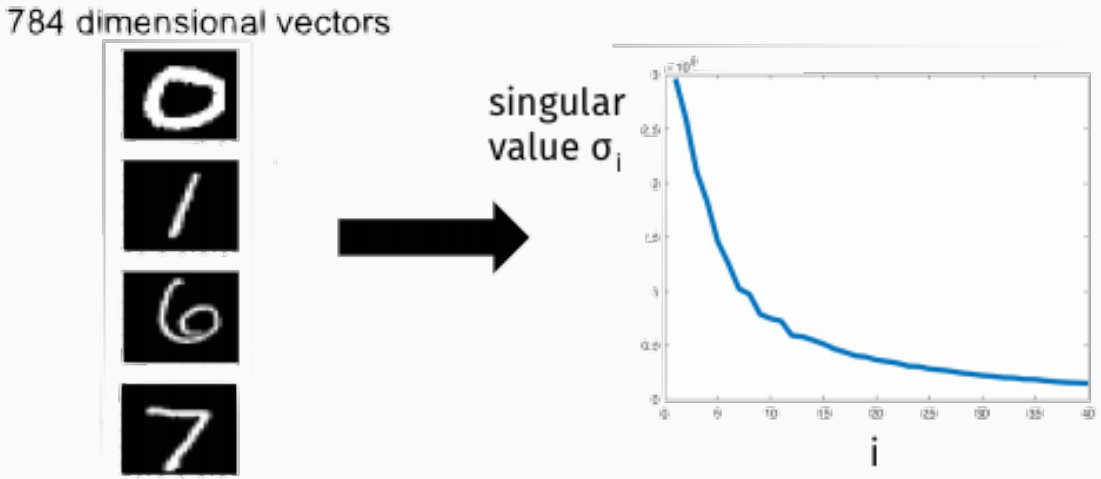
Observation 2: The optimal low-rank approximation error $E_k = \|\mathbf{X} - \mathbf{U}_k\mathbf{U}_k^T\mathbf{X}\|_F^2 = \|\mathbf{X} - \mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2$ can be written:

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

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

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

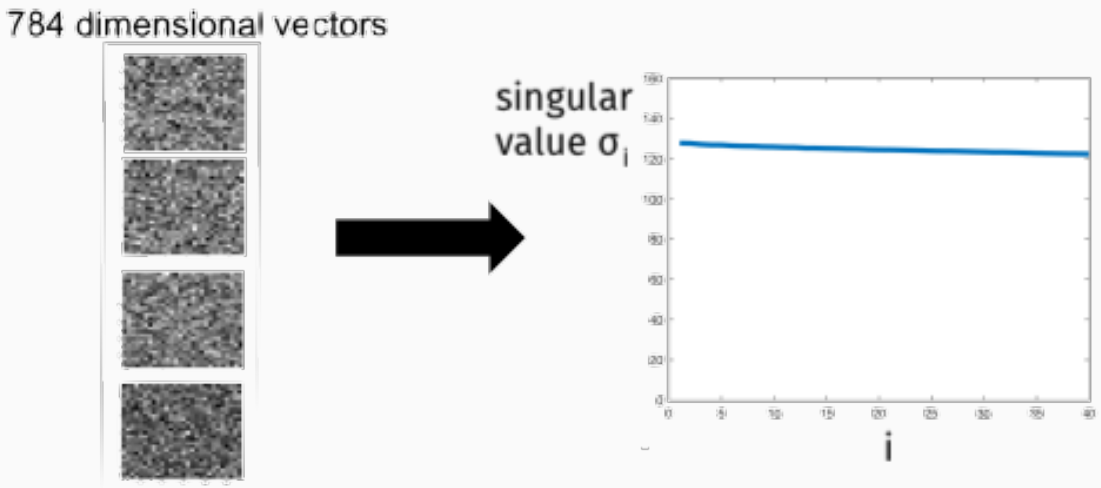
Can immediately get a sense of "how low-rank" a matrix is from it's spectrum:



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

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

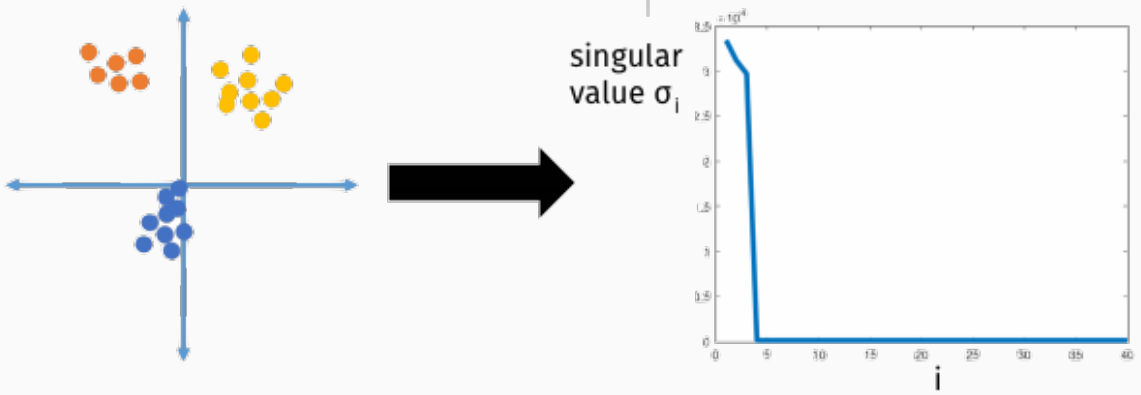
Can immediately get a sense of "how low-rank" a matrix is from it's spectrum:



Observation 2: The optimal low-rank approximation error $E_k = \|\mathbf{X} - \mathbf{U}_k\mathbf{U}_k^T\mathbf{X}\|_F^2 = \|\mathbf{X} - \mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2$ 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 $\mathbf{V}$. Then $\mathbf{U\Sigma} = \mathbf{XV}$.

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

Total runtime $\approx$

COMPUTING THE SVD (FASTER)

- Use an iterative algorithm.
- 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.

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 $\mathbf{U}\mathbf{\Sigma}\mathbf{V}$.

Power method:

- Choose $\mathbf{z}^{(0)}$ randomly. E.g. $\mathbf{z}_0 \sim \mathcal{N}(0, 1)$.
- 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 $\mathbf{z}_T$

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

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

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

Claim:

$$\mathbf{z}^{(1)} = \frac{1}{n_1}\left[c_1 \cdot \sigma_1^2\mathbf{v}_1 + \mathbf{c}_2 \cdot \sigma_2^2\mathbf{v}_2 + \ldots + c_d \cdot \sigma_d^2\mathbf{v}_d\right]$$

Claim:

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

Theorem (Basis 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:

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

Total runtime:

First observation: For all i

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

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

Corollary:

$$\max_j \frac{c_j}{c_1} \leq O(d^3).$$

POWER METHOD FORMAL CONVERGENCE

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

POWER METHOD FORMAL CONVERGENCE

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

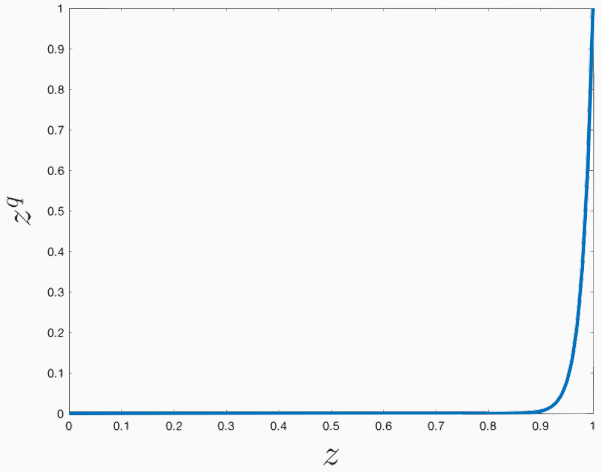
- $\alpha_1 \leq 1$.
- $\alpha_j^2 \leq (\epsilon/2d)^2$ for $j \geq 2$.
- $\alpha_1^2 \geq 1 - d \cdot (\epsilon/2d)^2 \implies \alpha_1 \geq 1 - \epsilon/2$.

$\|\mathbf{v}_1 - \mathbf{z}^{(T)}\|_2 \leq$

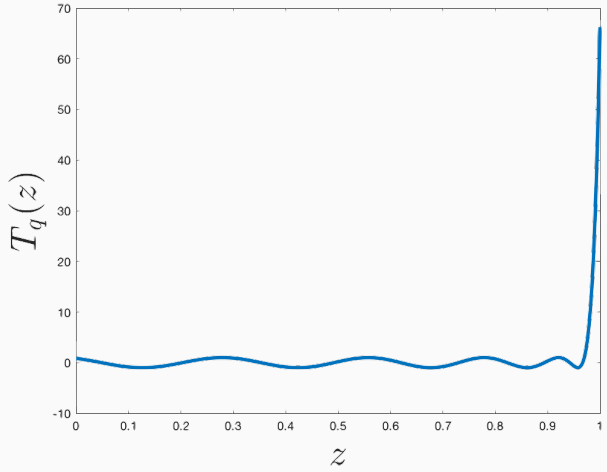
Theorem (Basis Power Method Convergence)

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:*

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



VS.



Lanczos method, Arnoldi method, etc. require $T = O\left(\frac{\log d/\epsilon}{\sqrt{\epsilon}}\right)$ steps for the same guarantee.

GENERALIZATIONS TO LARGE k

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

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

to obtain a nearly optimal low-rank approximation.