

CS-GY 6763: Lecture 9

Low-rank approximation and singular value decomposition

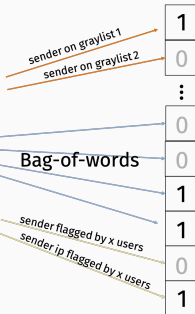
NYU Tandon School of Engineering, Prof. Christopher Musco

- 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 many scenarios.
- Midterms will be returned at the end of class.
- Problem Set 2 is being graded.

Return to data compression:

```

MIME-Version: 1.0 Date: Mon, 7 Oct 2019
14:51:30 -0400 Message-ID: <CANV120Gp=9-
39MLA8n0Py29_jwax6Qm9uWbiGCF8FgHDSa8@mail.gm
il.com> Subject: 92231 Reading Group, Meeting
2, tomorrow at 10am From: Christopher Musco
<cmusco@nyu.edu> To: aljmds@nyu.edu Content-
Type: multipart/alternative;
boundary="0000000000078ec24059456ba53" --
0000000000078ec24059456ba53 Content-Type:
text/plain; charset="UTF-8" I hope everyone
had a good weekend! Tomorrow at 10am in 370
Jay St. #1114* we will meet for the second
instantiation of the CS-09 92231 reading
group. Nick Peng will be leading a discussion
about the paper Simple Analysis of the Sparse
Johnson-Lindenstrauss Transform
<http://drops.dagstuhl.de/opus/volltexte/2018
/8305/pdf/Dagstuhl-OSEA-2018-15.pdf>. Please
read the abstract and introduction before the
meeting. Best, - CM *Christopher Musco,
Assistant Professor* *New York University,
Tandon School of Engineering* *401 578
2541* --0000000000078ec24059456ba53 Content-
Type: text/html; charset="UTF-8" Content-
Transfer-Encoding: quoted-printable
    
```

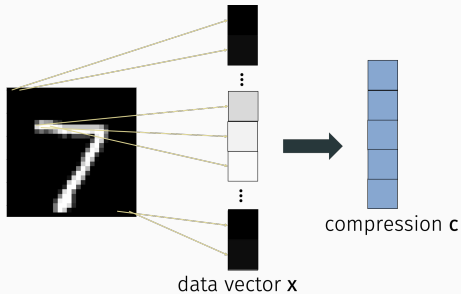


data vector x

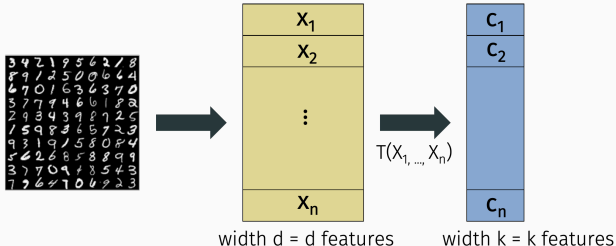


compression c

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:

The diagram illustrates the property of orthonormal matrices. It shows two equations connected by a double-headed blue arrow. On the left, a teal square labeled V^T is multiplied by a teal square labeled V , resulting in a white square with a diagonal line of ones from the top-left to the bottom-right, representing the identity matrix I . On the right, a teal square labeled V is multiplied by a teal square labeled V^T , also resulting in a white square with a diagonal line of ones, representing the identity matrix I .

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

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 $\|\mathbf{V}^T \mathbf{X}\|_F^2 = \|\mathbf{X}\|_F^2$.

LINEAR ALGEBRA REMINDER

The same is not true for rectangular matrices:

$$\begin{array}{|c|} \hline \mathbf{V}^T \\ \hline \end{array} \begin{array}{|c|} \hline \mathbf{V} \\ \hline \end{array} = \begin{array}{|c|} \hline 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \\ \hline \end{array} \quad \begin{array}{|c|} \hline \mathbf{V} \\ \hline \end{array} \begin{array}{|c|} \hline \mathbf{V}^T \\ \hline \end{array} = \begin{array}{|c|} \hline .5 & -1 & .7 & -2 \\ 1.6 & -.44 & 4.2 & -1.5 \\ 7.8 & .42 & -.5 & .67 \\ -2 & 2.0 & 1.1 & 8.0 \\ -1.5 & .55 & 3.2 & .5 \\ .67 & -2.8 & -2.4 & 1.6 \\ 9.0 & 8.7 & -7.7 & 7.8 \\ \hline \end{array}$$

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

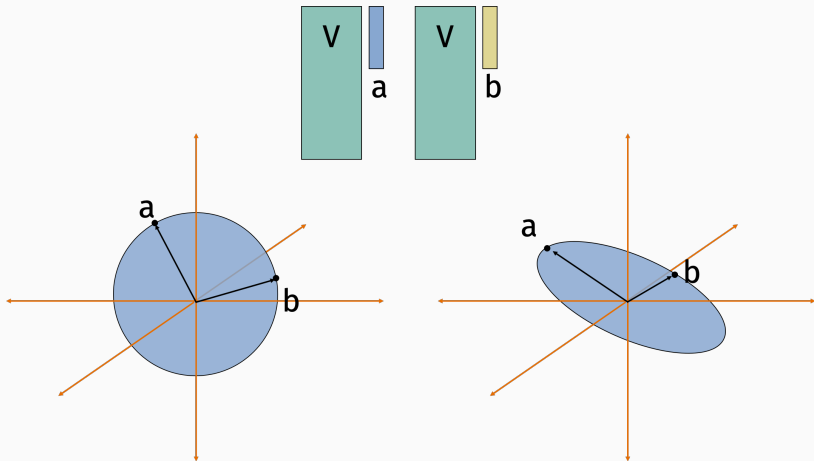
but

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

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.

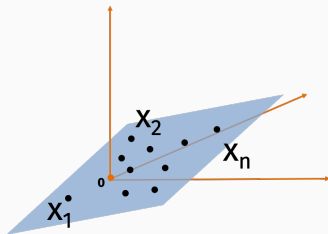
LINEAR ALGEBRA REMINDER

Multiplying a vector by V with orthonormal columns rotates and/or reflects the vector.

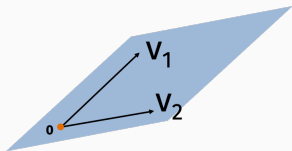


LOW-RANK DATA

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



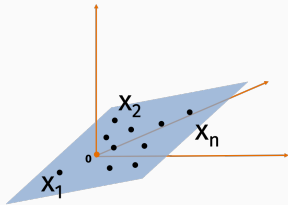
Let $\mathbf{v}_1, \dots, \mathbf{v}_k$ be orthogonal unit vectors spanning S .



For all i , we can write:

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

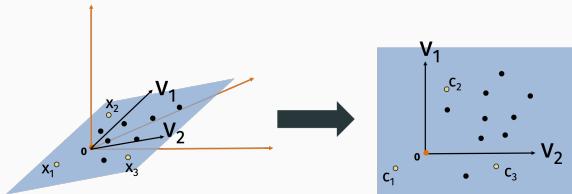
LOW-RANK DATA



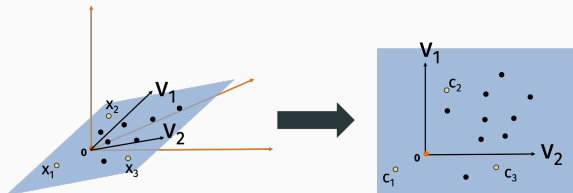
$$\begin{array}{|c|} \hline d \\ \hline x_1 \\ x_2 \\ \vdots \\ x_n \\ \hline \end{array} = \begin{array}{|c|} \hline k \\ \hline c_1 \\ c_2 \\ \vdots \\ c_n \\ \hline \end{array} \begin{array}{|c|} \hline d \\ \hline v_1 \\ v_k \\ \hline \end{array} \text{matrix } V^T$$

matrix X matrix C

What are c_1, \dots, c_n ?



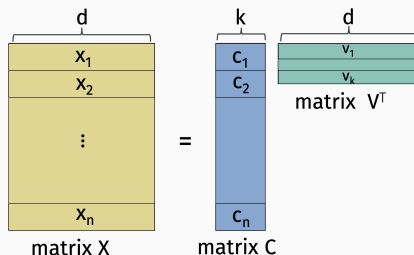
LOW-RANK DATA



Lots of information preserved:

- $\|x_i - x_j\|_2 = \|c_i - c_j\|_2$ for all i, j .
- $x_i^T x_j = c_i^T c_j$ for all i, j .
- Norms preserved, linear separability preserved, $\min \|Xy - b\| = \min \|Cz - b\|$, etc., etc.

LOW-RANK DATA



Formally, $C = XV^T$:

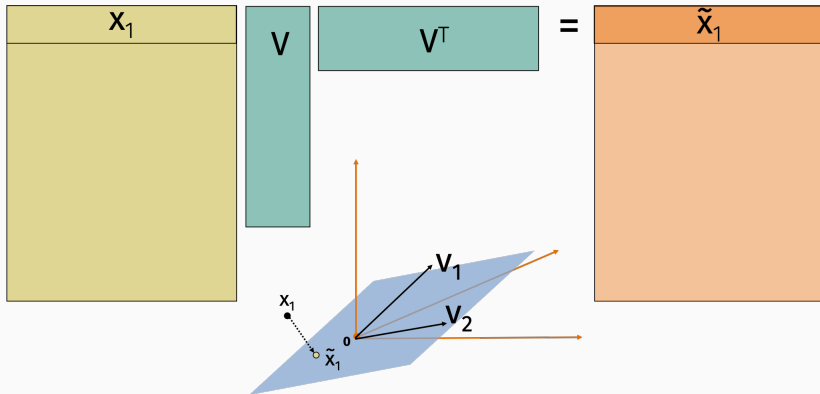
$$X = CV^T \Rightarrow XV = CV^TV$$

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

$$\text{So } X = XVV^T.$$

PROJECTION MATRICES

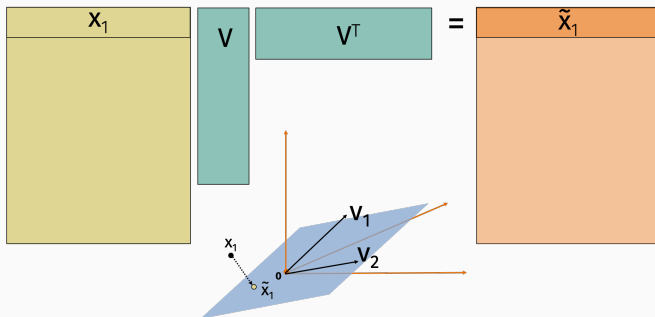
VV^T is a symmetric projection matrix.



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

PROJECTION MATRICES

VV^T is a symmetric projection matrix.



$x_1^T VV^T$ 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

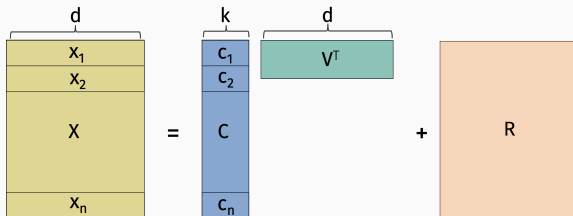
When \mathbf{X} 's rows lie close to a k dimensional subspace, we can still approximate

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

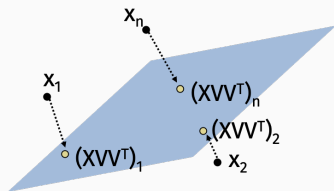
\mathbf{XV}^T is a low-rank approximation for \mathbf{X} .

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

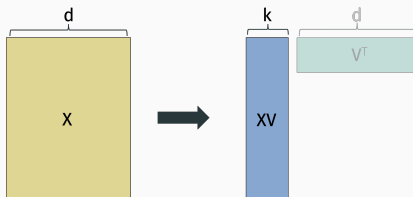
$$\mathbf{XV}^T = \arg \min_{\mathbf{C}} \|\mathbf{X} - \mathbf{CV}^T\|_F^2 = \sum_{i,j} (\mathbf{X}_{i,j} - (\mathbf{CV}^T)_{i,j})^2.$$



LOW-RANK APPROXIMATION



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

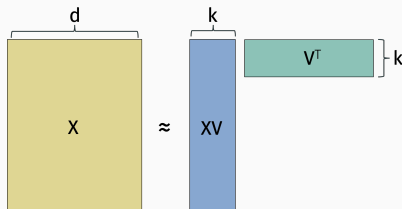


XV can be used as a compressed version of data matrix X .

WHY IS DATA APPROXIMATELY LOW-RANK?

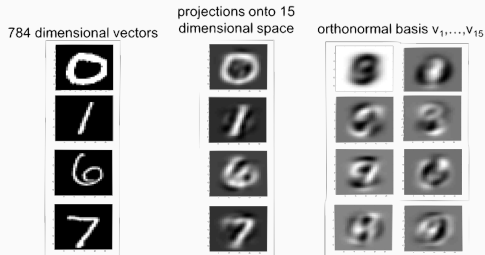
DUAL VIEW

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 .



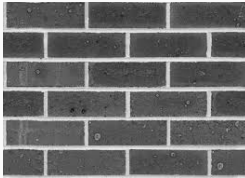
COLUMN REDUNDANCY

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

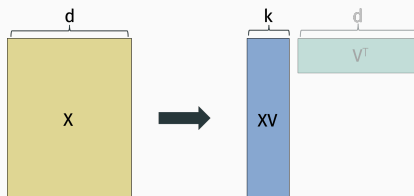
	bedrooms	bathrooms	sq.ft.	floors	list price	sale price
home 1	2	2	1800	2	200,000	195,000
home 2	4	2.5	2700	1	300,000	310,000
.
.
.
home n	5	3.5	3600	3	450,000	450,000

OTHER REASONS FOR LOW-RANK STRUCTURE

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



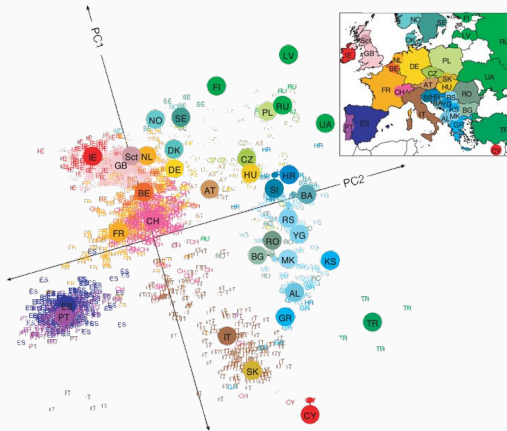
APPLICATIONS OF LOW-RANK APPROXIMATION



- $XV \cdot V^T$ takes $O(k(n + d))$ space to store instead of $O(nd)$.
- Regression problems involving $XV \cdot V^T$ can be solved in $O(nk^2)$ instead of $O(nd^2)$ time.
- 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 $\mathbf{V} \in \mathbb{R}^{d \times k}$:

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

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:

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

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

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

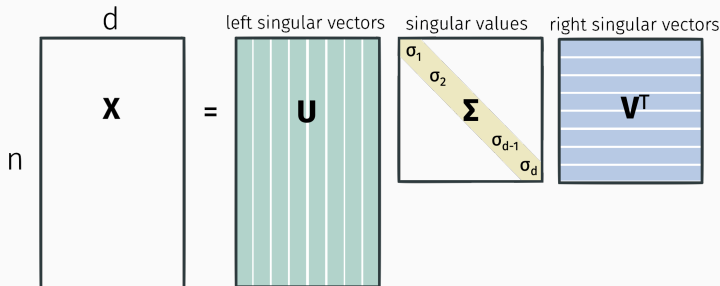
Choose \mathbf{v}_1 to be the top eigenvector of $\mathbf{X}^T \mathbf{X}$.

What about higher k ?

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 $\sigma_1 \geq \sigma_2 \geq \dots \sigma_d \geq 0$.

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} ”
- $\sigma_1, \dots, \sigma_k$ 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 $\mathbf{X}^T\mathbf{X}$.
- $\sigma_i^2 = \lambda_i(\mathbf{X}\mathbf{X}^T) = \lambda_i(\mathbf{X}^T\mathbf{X})$

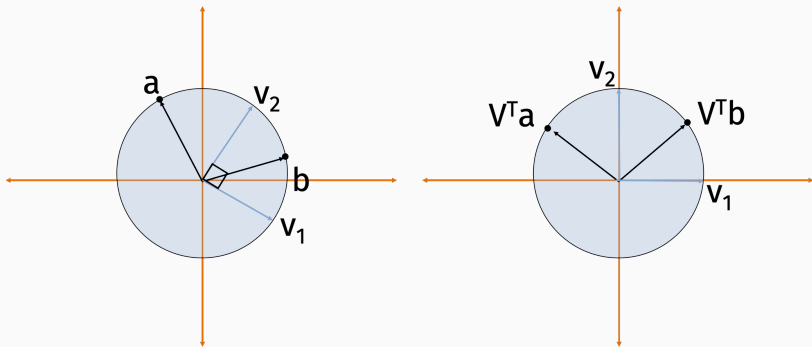
Exercise: Check this can be checked directly.

Important take away from singular value decomposition.

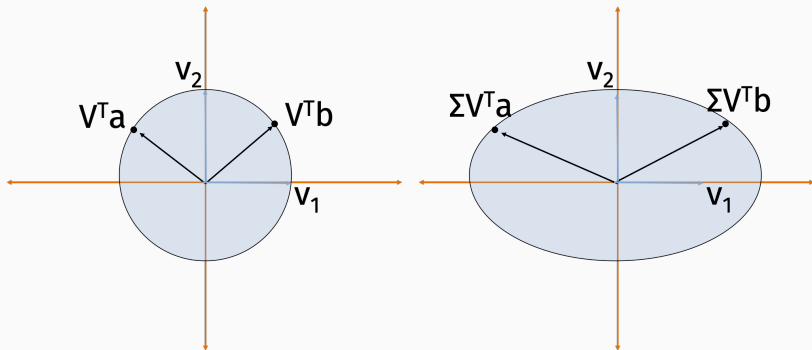
Multiplying any vector \mathbf{a} by a matrix \mathbf{X} to form \mathbf{Xa} can be viewed as a composition of 3 operations:

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

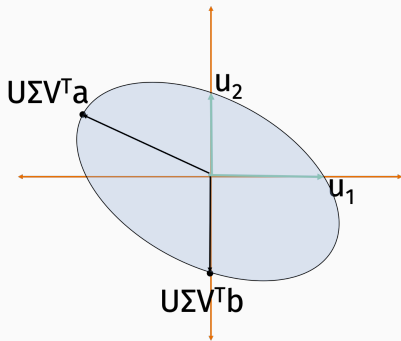
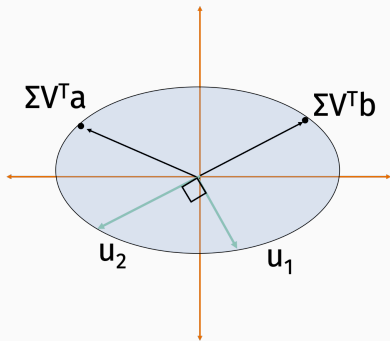
SINGULAR VALUE DECOMPOSITION: ROTATE/REFLECT



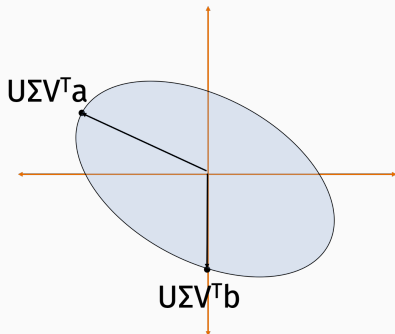
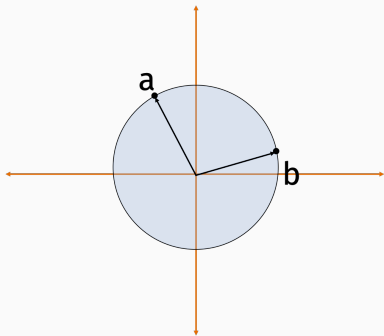
SINGULAR VALUE DECOMPOSITION: STRETCH



SINGULAR VALUE DECOMPOSITION: ROTATE/REFLECT

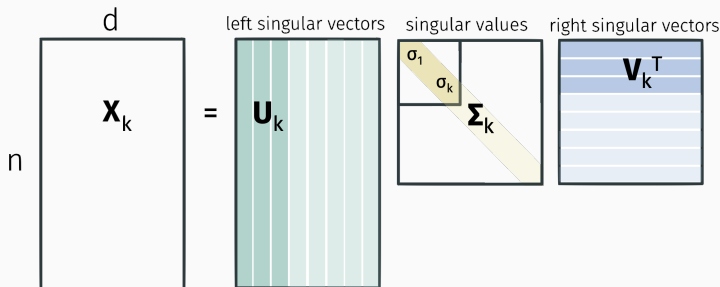


SINGULAR VALUE DECOMPOSITION



SINGULAR VALUE DECOMPOSITION

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



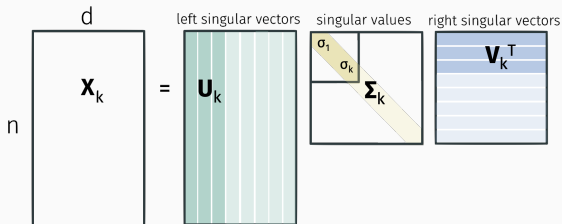
$$\mathbf{X}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T = \mathbf{U}_k \mathbf{U}_k^T \mathbf{X} = \mathbf{X} \mathbf{V}_k \mathbf{V}_k^T.$$

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

Connection to Principal Component Analysis:

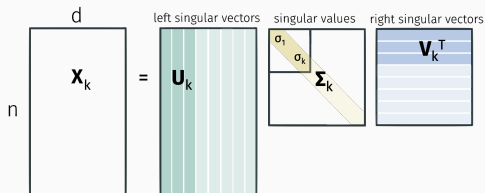
- Let $\bar{\mathbf{X}} = \mathbf{X} - \mathbf{1}\boldsymbol{\mu}^T$ where $\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$. I.e. $\bar{\mathbf{X}}$ is obtained by mean centering \mathbf{X} 's rows.
- Let $\bar{\mathbf{U}}\bar{\boldsymbol{\Sigma}}\bar{\mathbf{V}}^T$ be the SVD of $\bar{\mathbf{X}}$.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

USEFUL OBSERVATIONS



Observation 2: The optimal low-rank approximation error

$E_k = \|\mathbf{X} - \mathbf{X}\mathbf{V}_k\mathbf{V}_k^T\|_F^2 = \|\mathbf{X}\|_F^2 - \|\mathbf{X}\mathbf{V}_k\|_F^2$ can be written:

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

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:

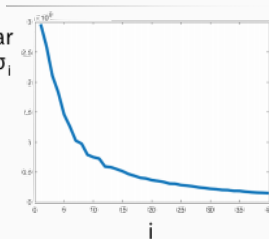
$$E_k = \sum_{i=k+1}^d \sigma_i^2.$$

Can immediately get a sense of “how low-rank” a matrix is from it’s spectrum:

784 dimensional vectors



singular
value σ_i



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:

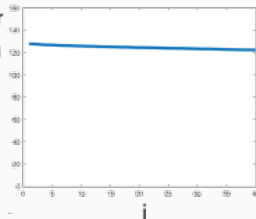
$$E_k = \sum_{i=k+1}^d \sigma_i^2.$$

Can immediately get a sense of “how low-rank” a matrix is from it’s spectrum:

784 dimensional vectors



singular
value σ_i



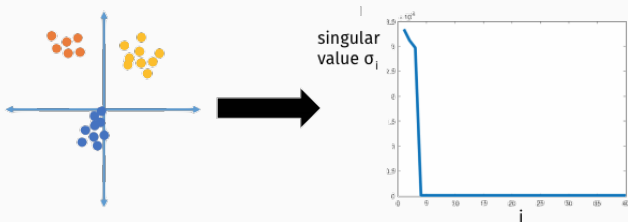
SPECTRAL PLOTS

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 \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 \approx

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).

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

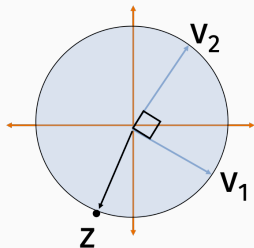
Power method:

- 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 $i = 1, \dots, T$
 - $\mathbf{z}^{(i)} = \mathbf{X}^T \cdot (\mathbf{X}\mathbf{z}^{(i-1)})$
 - $n_i = \|\mathbf{z}^{(i)}\|_2$
 - $\mathbf{z}^{(i)} = \mathbf{z}^{(i)} / n_i$

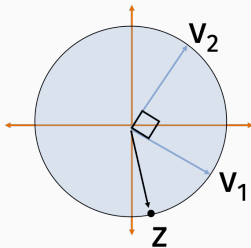
Return $\mathbf{z}^{(T)}$

POWER METHOD INTUITION

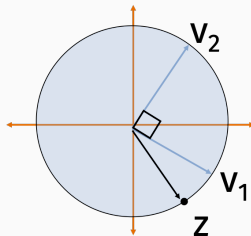
0 iterations



1 iterations



2 iterations



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:

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

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

ONE STEP ANALYSIS OF POWER METHOD

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

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

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

$$\vdots$$

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

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

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

ONE STEP ANALYSIS OF POWER METHOD

Claim: After update $\mathbf{z}^{(i)} = \frac{1}{n_1} \mathbf{X}^T \mathbf{X} \mathbf{z}^{(i-1)}$,

$$c_j^{(i)} = \frac{1}{n_1} \sigma_j^2 c_j^{(i-1)}$$

$$\mathbf{z}^{(i)} = \frac{1}{n_1} \left[c_1^{(i-1)} \sigma_1^2 \cdot \mathbf{v}_1 + c_2^{(i-1)} \sigma_2^2 \cdot \mathbf{v}_2 + \dots + c_d^{(i-1)} \sigma_d^2 \cdot \mathbf{v}_d \right]$$

MULTI-STEP ANALYSIS OF POWER METHOD

Claim: After T updates:

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

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:

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

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

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

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

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

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.

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

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 $T =$

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 $i = 1, \dots, T$
 - $\mathbf{Z}^{(i)} = \mathbf{X}^T \cdot (\mathbf{X} \mathbf{Z}^{(i-1)})$
 - $\mathbf{Z}^{(i)} = \text{orth}(\mathbf{Z}^{(i)})$

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

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

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

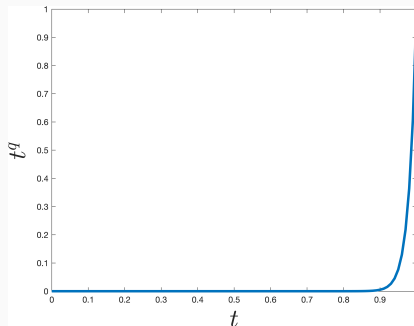
Possible to “accelerate” these methods.

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

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



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

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

Along the way we computed:

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

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

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

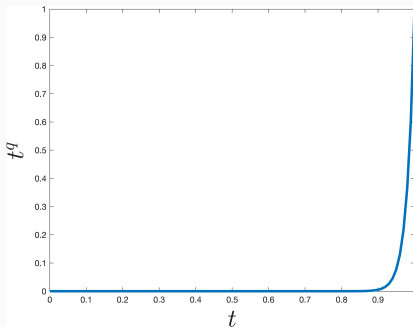
Want to find \mathbf{v} , which minimizes $\|\mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T\|_F^2$.

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 best vector in the Krylov subspace, instead of just using last vector.
 - Can be done in $O(\text{nnz}(\mathbf{X}) \cdot k + dk^2)$ time.

LANCZOS METHOD ANALYSIS

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



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

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

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 both \mathbf{v}_1 and \mathbf{v}_2 . But in this case:

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

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

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.