CS 58500 – Theoretical Computer Science Toolkit

Lecture 13 (03/03)

Spectral Methods

https://ruizhezhang.com/course_spring_2026.html

Today's Lecture

- Introduction to Spectral Method
- SVD and Best-Fit Subspace
- k-Means Clustering
- Complexity of SVD

Introduction to Spectral Methods: Motivation

Datasets are the foundation of progress in AI

For text:

- GPT-1 (2018): 3 B Tokens
- GPT-2 (2019): 30 B Tokens
- GPT-3 (2020): 300 B Tokens
- GPT-4 (2023): 13k B Tokens
- GPT-5 (2025): 70k B Tokens (?)

20000x growth in 7 years

For images:

- ImageNet (2009): 1 Million images
- LAION-5B (2022): 5 Billion Images

5000x growth in 5 years

Slides from Alex Dimakis's talk

Introduction to Spectral Methods: Motivation

Challenges in modern data science

- Enormous data
- Curse of dimensionality
- Imperfect data (noisy, messy, missing features, …)

"Spectral method refers to a collection of algorithms built upon the eigenvectors (resp. singular vectors) and eigenvalues (resp. singular values) of some properly designed matrices generated from data"

(*Spectral methods for data science: A statistical perspective,* Chen et al. '21)

Introduction to Spectral Methods: Applications

Clustering

- Community detection in networks
- Joint image alignment
- Ranking

Learning Hidden Structures

- Synchronization in cryo-EM
- Tensor estimation

Numerical Linear Algebra

- Dimensionality reduction
- low-rank matrix estimation

Econometric and financial modeling

Spectral graph theory

- Random walk
- Electrical flows and effective resistance
- Expander graph
- Graph sparsification
- Graph partitioning

Sum-of-squares (SoS) / “Finding a needle in a haystack”

- Planted sparse vector
- Tensor PCA
- Refuting random CSP

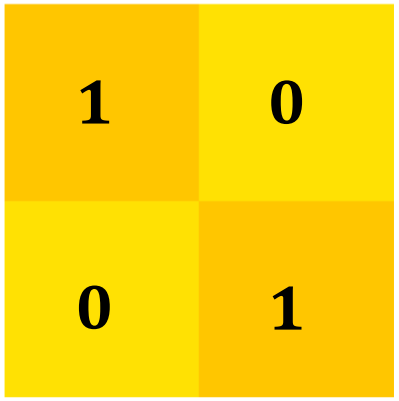
Introduction to Spectral Methods: Examples

Clustering: grouping the elements based on their mutual similarities

- Suppose n people are divided into two groups, with the first $n/2$ people in group A, and the remaining $n/2$ people in group B
- We observe pairwise similarity measurements based on their group memberships:

$$A_{ij} = \begin{cases} 1 & \text{if } i,j \text{ are in the same group} \\ 0 & \text{otherwise} \end{cases}$$

$$\tilde{A}_{ij} \sim \text{Bernoulli}\left(\frac{1 \pm \delta}{2}\right)$$

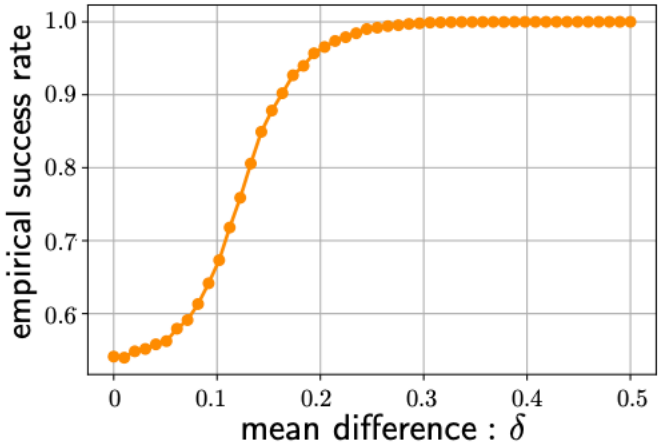


Ideal $\boldsymbol{A}$

$$\boldsymbol{v}_1 = \mathbf{1}_n$$

$$\boldsymbol{v}_2 = \begin{bmatrix} +\mathbf{1}_{\frac{n}{2}} \\ -\mathbf{1}_{\frac{n}{2}} \end{bmatrix}$$

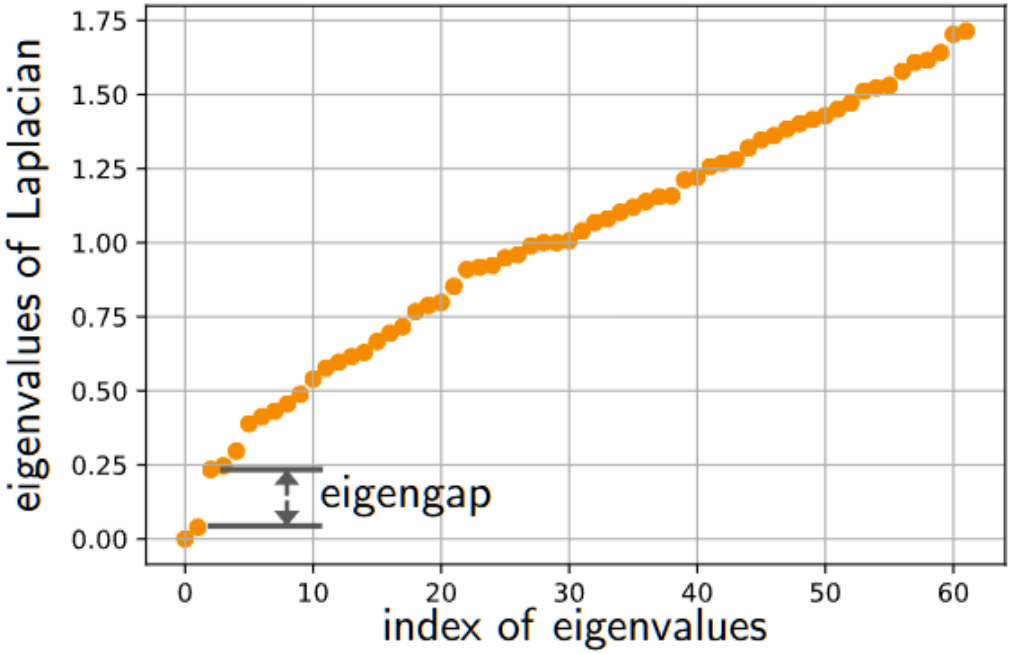
Noisy $\tilde{\boldsymbol{A}}$

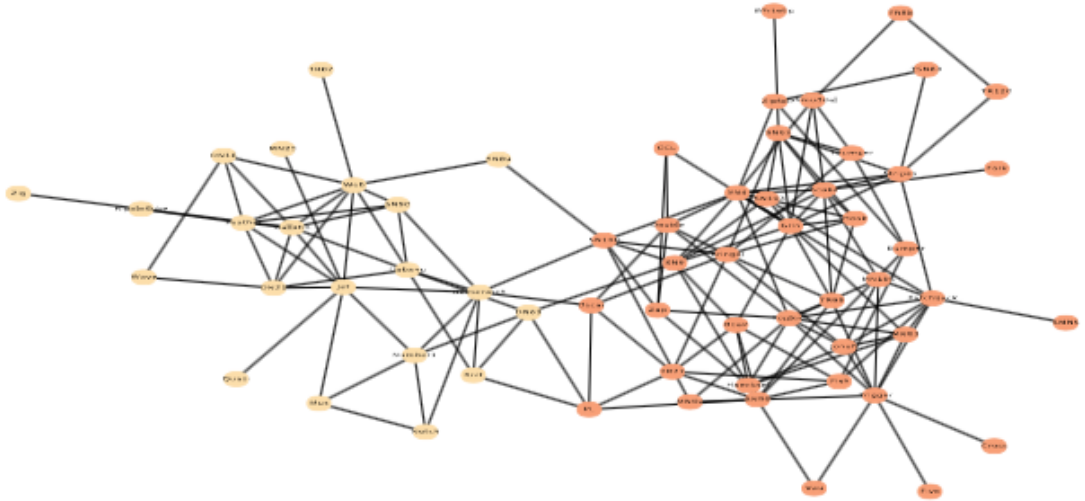


Introduction to Spectral Methods: Examples

Clustering: grouping the elements based on their mutual similarities

- Real-world data: two communities of dolphins in Doubtful Sound, New Zealand





Introduction to Spectral Methods: Examples

Principal component analysis (PCA): identifying a rank-r subspace to fit the data samples $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^d$

- Consider the sample covariance matrix:

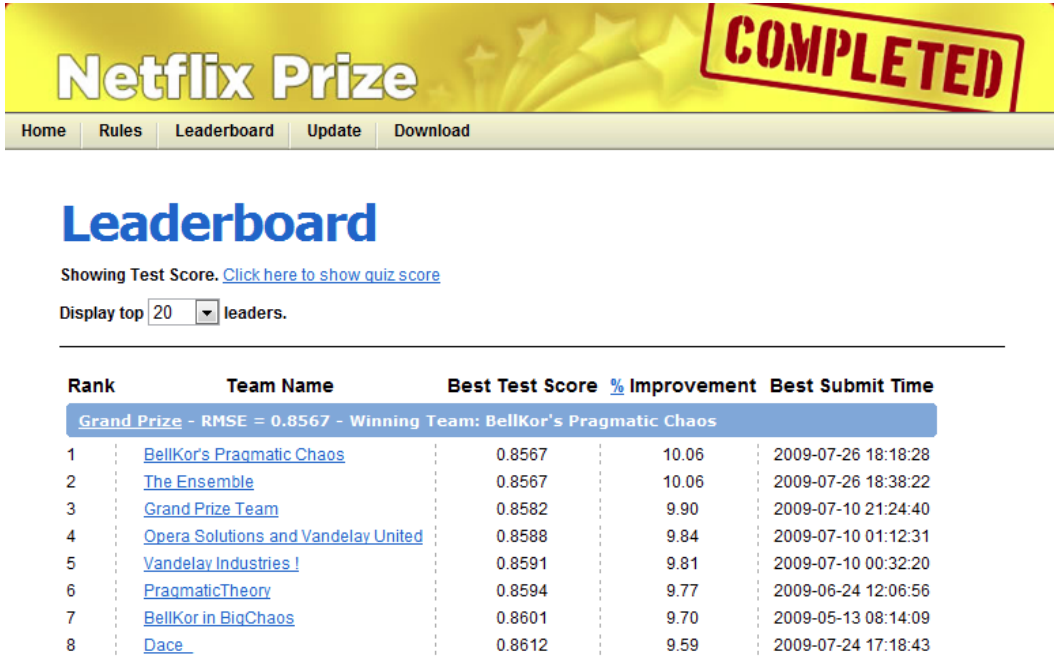
$$\boldsymbol{M} \coloneqq \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top \in \mathbb{R}^{d\times d}$$

- If $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ are approximately lie in a rank-r subspace $\boldsymbol{U}_\star$
- Then we can take the rank-r leading eigenspace of $\boldsymbol{M}$, if the signal-to-noise ratio (SNR) is sufficiently high
- Real-world example: Eigenface

Introduction to Spectral Methods: Examples

Matrix completion: recovering a low-rank matrix from sub-sampled or incomplete observations

- Netflix prize



	movie 1	movie 2	movie 3	movie 4	movie 5	movie 6	movie 7	movie 8	movie 9	movie 10	...	movie 17770
user 1			1		2							3
user 2		2		3	3			4				
user 3							5	3		4		
user 4	2				3			2				2
user 5		4				5			3			4
user 6			2									
user 7			2					4	2	3		
user 8	3	4				4						
user 9									3			
user 10			1		2							2
...												
user 480189		4			3			3				

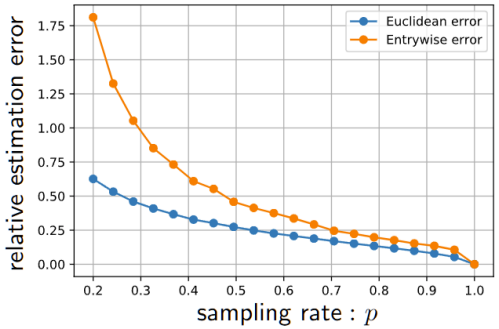
Introduction to Spectral Methods: Examples

Matrix completion: recovering a low-rank matrix from sub-sampled or incomplete observations

- Let $\boldsymbol{M}^\star \in \mathbb{R}^{n\times n}$ be the ground-truth low-rank matrix
- Suppose the entries of $\boldsymbol{M}^\star$ are randomly sampled with probability p independently
- An unbiased estimator for $\boldsymbol{M}^\star$ is:

$$\hat{M}_{ij} := \begin{cases} \frac{1}{p} M^\star_{ij} & \text{if } (i,j) \text{ is observed} \\ 0 & \text{otherwise} \end{cases}$$

- Then, we compute the best rank-r approximation of $\widehat{\boldsymbol{M}}$ by the rank-r singular value decomposition



Introduction to Spectral Methods: Examples

Ranking based on pairwise comparison

- People struggle to rank many items at once, but find it much easier to make pairwise comparisons
- Bradley-Terry model: suppose there are latent scores $\{w_i^\star\}_{i\in[n]}$ for each item
- The outcome of the comparison is generated as:

$$\Pr[i \succ j] = \frac{w_i^\star}{w_i^\star + w_j^\star} \qquad \forall i, j \in [n]$$

- Let $u_i^\star := \log w_i^\star$ for $i \in [n]$. We also have

$$\Pr[i \succ j] = \frac{\exp(u_i^\star)}{\exp(u_i^\star) + \exp(u_j^\star)} = \sigma(u_i^\star - u_j^\star) \qquad \forall i, j \in [n]$$

  where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the logistic function

- Very popular recently due to applications in aligning LLMs with human preference (RLHF, DPO…)

Introduction to Spectral Methods: Examples

Ranking based on pairwise comparison

- Bradley-Terry model: suppose there are latent scores $\{w_i^\star\}_{i\in[n]}$ for each item
- The outcome of the comparison is generated as:

$$\Pr[i \succ j] = \frac{w_i^\star}{w_i^\star + w_j^\star} \qquad \forall i, j \in [n]$$

- We can recover the global ranking by considering a probability transition matrix $\boldsymbol{P}^\star$ of a Markov chain:

$$P_{ij}^\star = \begin{cases} \frac{1}{n}\frac{w_j^\star}{w_i^\star + w_j^\star} & \text{if } i \neq j \\ 1 - \sum_{k\neq i} P_{ik}^\star & \text{if } i = j \end{cases}$$

- Every entry of $\boldsymbol{P}^\star$ is non-negative, and every row sums to one, i.e., $\boldsymbol{P}^\star$ is a stochastic matrix

Introduction to Spectral Methods: Examples

Ranking based on pairwise comparison

- Bradley-Terry model: suppose there are latent scores $\{w_i^\star\}_{i\in[n]}$ for each item
- The outcome of the comparison is generated as:

$$\Pr[i \succ j] = \frac{w_i^\star}{w_i^\star + w_j^\star} \qquad \forall i, j \in [n]$$

- We can recover the global ranking by considering a probability transition matrix $\boldsymbol{P}^\star$ of a Markov chain:

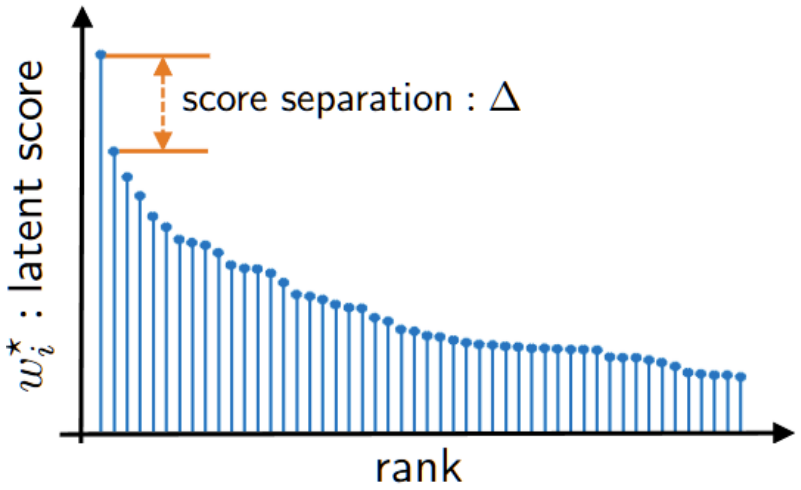
$$P_{ij}^\star = \begin{cases} \frac{1}{n}\frac{w_j^\star}{w_i^\star + w_j^\star} & \text{if } i \neq j \\ 1 - \sum_{k\neq i} P_{ik}^\star & \text{if } i = j \end{cases}$$

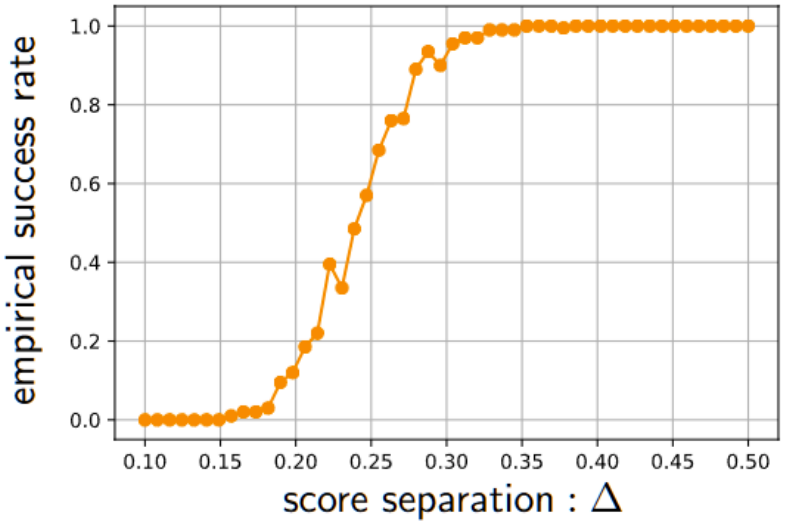
- $\boldsymbol{w}^{\star\top}\boldsymbol{P}^\star = \boldsymbol{w}^{\star\top}$, i.e., the latent score vector is a left eigenvector with eigenvalue one

Introduction to Spectral Methods: Examples

Ranking based on pairwise comparison

- Spectral ranking algorithm:
  - Use pairwise comparison to form an unbiased estimator of $\boldsymbol{P}^\star$
  - Compute its leading left eigenvector and report the ranking according to this eigenvector





- Google's PageRank algorithm uses a similar idea

Introduction to Spectral Method: Unified Framework

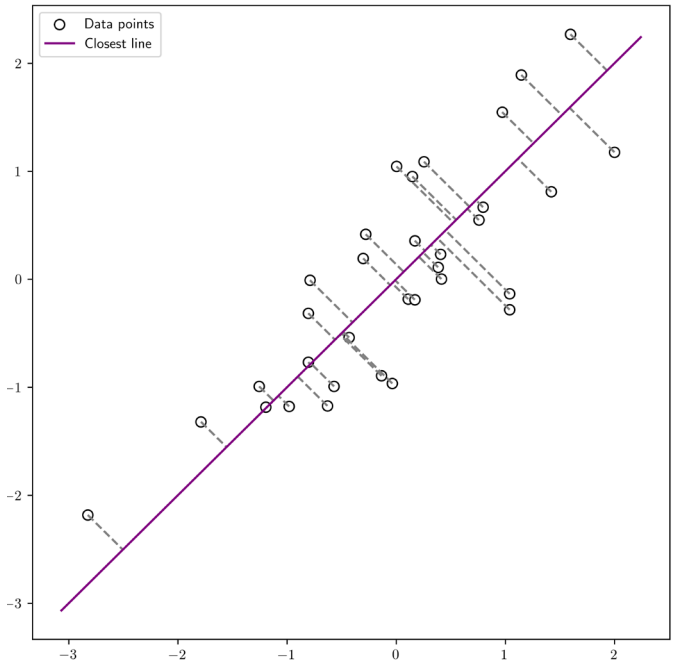
1. Identify a key matrix $\boldsymbol{M}^{\star}$ whose eigenvectors or singular vectors contains the target information (but $\boldsymbol{M}^{\star}$ is usually inaccessible)
2. Construct a surrogate matrix $\boldsymbol{M}$ of $\boldsymbol{M}^{\star}$ using the data samples in hand, and compute the corresponding eigenvectors or singular vectors of this surrogate matrix

- Sample complexity?
- Time complexity?
- Noise stability?
- Beyond worst-case (e.g., average-case, smoothed analysis)?

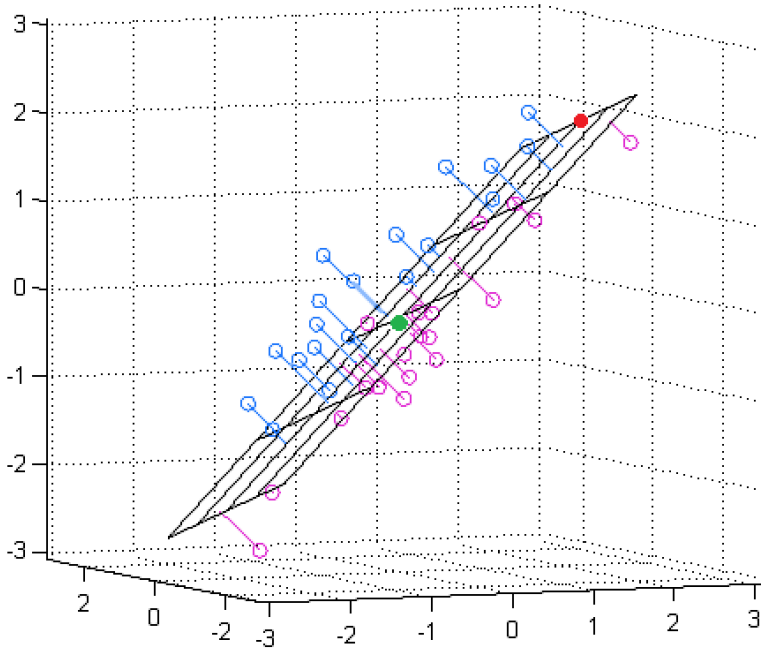
Today's Lecture

- Introduction to Spectral Method
- SVD and Best-Fit Subspace
- k-Means Clustering
- Complexity of SVD

SVD and Best-Fit Subspace



Linear regression



Best-fit subspace

Linear Algebra Self-Test

- Invertible matrix
- Diagonalizable matrix
- Normal matrix
- Determinant
- Similarity transformation
- Transpose, Hermitian transpose
- Pseudoinverse
- Orthogonal matrix, Unitary matrix
- Permutation matrix
- Vandermonde matrix
- Hankle matrix, Toeplitz matrix

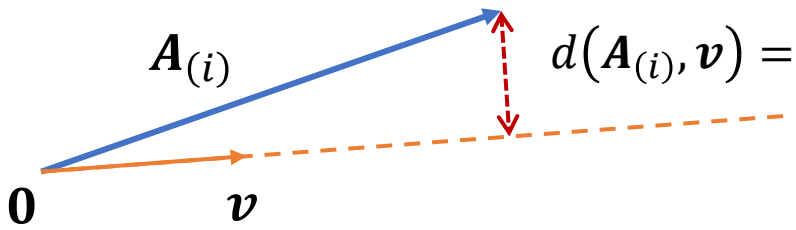
- Eigendecomposition
- Singular value decomposition
- QR decomposition/Gram-Schmidt
- Jordan decomposition
- Polar decomposition
- Operator norm
- Frobenius norm
- Schatten p-norms
- Rank
- Kernel, Range

Singular Value Decomposition

- For an $n \times n$ matrix $\boldsymbol{A}$, an eigenvalue λ with corresponding eigenvector $\boldsymbol{v}$ satisfies
$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}$$
- If $\det(\boldsymbol{A}) \neq 0$, then $\boldsymbol{A}$ has n nonzero eigenvalues and n corresponding eigenvectors
- For an $m \times n$ matrix $\boldsymbol{A}$, a singular value σ and corresponding singular vectors $\boldsymbol{u} \in \mathbb{R}^m, \boldsymbol{v} \in \mathbb{R}^n$ satisfies
$$\boldsymbol{A}\boldsymbol{v} = \sigma \boldsymbol{u}, \qquad \boldsymbol{u}^\top \boldsymbol{A} = \sigma \boldsymbol{v}^\top$$
- You can observe that u and v are equal length:
$$\boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{v} = \sigma \boldsymbol{u}^\top \boldsymbol{u} = \sigma \|\boldsymbol{u}\|_2^2 = \sigma \boldsymbol{v}^\top \boldsymbol{v} = \sigma \|\boldsymbol{v}\|_2^2$$
- We may always assume that $\|\boldsymbol{u}\|_2 = \|\boldsymbol{v}\|_2 = 1$

Proposition. The vector $\boldsymbol{v}_1 = \arg\max_{\|\boldsymbol{v}\|_2=1} \|\boldsymbol{A}\boldsymbol{v}\|_2^2$ is a singular vector, and $\|\boldsymbol{A}\boldsymbol{v}_1\|_2$ is the largest singular value

Best-Fit Line



$$d(\boldsymbol{A}_{(i)}, \boldsymbol{v}) = \left\| \boldsymbol{A}_{(i)} - (\boldsymbol{A}_{(i)}\boldsymbol{v})\boldsymbol{v}^{\mathsf{T}} \right\|_2$$

$$\max_{\|\boldsymbol{v}\|_2=1} \|\boldsymbol{A}\boldsymbol{v}\|_2^2 = \|\boldsymbol{A}\|_F^2 - \min_{\|\boldsymbol{v}\|_2=1} \sum_{i\in[m]} d(\boldsymbol{A}_{(i)}, \boldsymbol{v})^2$$

- If we take the data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^n$ to be the rows of A
- Then the top singular vector $\boldsymbol{v}_1 = \arg\max_{\|\boldsymbol{v}\|_2=1} \|\boldsymbol{A}\boldsymbol{v}\|_2^2$ defines the direction of the line through the origin that best fits the data points in the mean squared error (MSE) sense

Singular Value Decomposition

Lemma. The (right) singular vectors of $\boldsymbol{A}$ are the eigenvectors of $\boldsymbol{A}^\top \boldsymbol{A}$

Proof.

- Let $\boldsymbol{v}$ be a singular vector of $\boldsymbol{A}$ with singular value σ. Then, we have

$$\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{v} = \boldsymbol{A}^\top \sigma \boldsymbol{u} = \sigma(\boldsymbol{u}^\top \boldsymbol{A})^\top = \sigma^{\mathbf{2}} \boldsymbol{v}$$

- Conversely, let $\boldsymbol{v}$ be an eigenvector of $\boldsymbol{A}^\top \boldsymbol{A}$ with eigenvalue λ. Then, we have

$$\boldsymbol{v}^\top \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{v} = \lambda \|\boldsymbol{v}\|_2^2 \implies \boldsymbol{u} \coloneqq \frac{\boldsymbol{A}\boldsymbol{v}}{\sqrt{\lambda}} \text{ and } \sigma = \sqrt{\lambda}$$

■

- The proof also implies that the singular values $\sigma \geq 0$

Singular Value Decomposition

Proposition. The vector $\boldsymbol{v}_1 = \arg\max_{\|\boldsymbol{v}\|_2=1} \|\boldsymbol{A}\boldsymbol{v}\|_2^2$ is a singular vector, and $\|\boldsymbol{A}\boldsymbol{v}_1\|_2$ is the largest singular value

Proof.

- Define the Lagrangian $\mathcal{L}(\boldsymbol{v}, \lambda) \coloneqq \|\boldsymbol{A}\boldsymbol{v}\|_2^2 - \lambda(\|\boldsymbol{v}\|_2^2 - 1) = \boldsymbol{v}^\top \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{v} - \lambda(\boldsymbol{v}^\top \boldsymbol{v} - 1)$
- Then, we have

$$\nabla_{\boldsymbol{v}} \mathcal{L}(\boldsymbol{v}, \lambda) = 2(\boldsymbol{A}^\top \boldsymbol{A})\boldsymbol{v} - 2\lambda \boldsymbol{v} = 0 \quad \Longrightarrow \quad (\boldsymbol{A}^\top \boldsymbol{A})\boldsymbol{v} = \lambda \boldsymbol{v}$$

- That is, every eigenvector of $\boldsymbol{A}^\top \boldsymbol{A}$ is a stationary point for this constrained optimization problem
- Thus, $\boldsymbol{v}_1$ is the top eigenvector of $\boldsymbol{A}^\top \boldsymbol{A}$, which is also the top singular vector of $\boldsymbol{A}$ by the previous lemma

■

Singular Value Decomposition

Theorem. Define the k-dimensional subspace V_k as the span of the following k vectors:

$$\boldsymbol{v}_1 \coloneqq \arg\max_{\|\boldsymbol{v}\|_2=1} \|\boldsymbol{A}\boldsymbol{v}\|_2^2$$

$$\boldsymbol{v}_2 \coloneqq \arg\max_{\|\boldsymbol{v}\|_2=1,\, \boldsymbol{v}\perp\boldsymbol{v}_1} \|\boldsymbol{A}\boldsymbol{v}\|_2^2$$

$$\vdots$$

$$\boldsymbol{v}_k \coloneqq \arg\max_{\|\boldsymbol{v}\|_2=1,\, \boldsymbol{v}\perp\boldsymbol{v}_1,\ldots,\boldsymbol{v}_{k-1}} \|\boldsymbol{A}\boldsymbol{v}\|_2^2$$

Then, V_k is optimal in the sense that

$$V_k = \arg\min_{\dim(V)=k} \sum_{i\in[m]} d\left(\boldsymbol{A}_{(i)}, V\right)^2$$

Moreover, $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ are all singular vectors with corresponding singular values

$$\sigma_1 = \|\boldsymbol{A}\boldsymbol{v}_1\|_2 \geq \sigma_2 = \|\boldsymbol{A}\boldsymbol{v}_2\|_2 \geq \cdots \geq \sigma_n = \|\boldsymbol{A}\boldsymbol{v}_n\|_2$$

And $\boldsymbol{A} = \sum_{i=1}^{n} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$ with $\boldsymbol{u}_i \coloneqq \sigma_i^{-1}\boldsymbol{A}\boldsymbol{v}_i$ is a singular value decomposition (SVD) of $\boldsymbol{A}$

Singular Value Decomposition

Proof.

- We prove the optimality of V_k by induction. $k = 1$ is trivial. Suppose that V_{k-1} is optimal
- Suppose V_k' is the optimal k-dimension subspace.
- Let $\boldsymbol{w}_1, \dots, \boldsymbol{w}_k \in \mathbb{R}^n$ be a set of orthonormal basis for V_k' such that $\boldsymbol{w}_k \in V_{k-1}^{\perp}$
- Then, by the optimality of V_{k-1}, we have

$$\|\boldsymbol{A}\boldsymbol{w}_1\|_2^2 + \cdots + \|\boldsymbol{A}\boldsymbol{w}_{k-1}\|_2^2 + \|\boldsymbol{A}\boldsymbol{w}_k\|_2^2 \leq \|\boldsymbol{A}\boldsymbol{v}_1\|_2^2 + \cdots + \|\boldsymbol{A}\boldsymbol{v}_{k-1}\|_2^2 + \|\boldsymbol{A}\boldsymbol{w}_k\|_2^2$$

- Thus, $V_k' = \mathrm{span}\{V_{k-1} \cup \{\boldsymbol{w}_k\}\}$
- By the optimality of V_k', $\boldsymbol{w}_k = \arg \max_{\|\boldsymbol{v}\|_2=1,\, \boldsymbol{v} \perp \boldsymbol{v}_1, \dots, \boldsymbol{v}_{k-1}} \|\boldsymbol{A}\boldsymbol{v}\|_2^2$
- Hence, V_k is also optimal

Singular Value Decomposition

Proof of 'Moreover'.

- We can generalize the previous proposition to $\boldsymbol{v}_2, \dots, \boldsymbol{v}_n$.
- Consider $v_2 = \arg\max_{\|v\|_2=1,\, \boldsymbol{v}\perp\boldsymbol{v}_1} \|\boldsymbol{A}\boldsymbol{v}\|_2^2$. Define the Lagrangian

$$\mathcal{L}(\boldsymbol{v}, \lambda, \mu) \coloneqq \|\boldsymbol{A}\boldsymbol{v}\|_2^2 - \lambda(\|\boldsymbol{v}\|_2^2 - 1) - \mu\boldsymbol{v}_1^\top\boldsymbol{v}$$

- $\nabla_{\boldsymbol{v}}\mathcal{L}(\boldsymbol{v}, \lambda, \mu) = 2(\boldsymbol{A}^\top\boldsymbol{A})\boldsymbol{v} - 2\lambda\boldsymbol{v} - \mu\boldsymbol{v}_1 = 0$
- Multiplying $\boldsymbol{v}_1^\top$ on the left:

$$2\boldsymbol{v}_1^\top(\boldsymbol{A}^\top\boldsymbol{A})\boldsymbol{v} - 2\lambda\boldsymbol{v}_1^\top\boldsymbol{v} - \mu\boldsymbol{v}_1^\top\boldsymbol{v}_1 = 2\sigma_1^2\boldsymbol{v}_1^\top\boldsymbol{v} - 2\lambda\boldsymbol{v}_1^\top\boldsymbol{v} - \mu = -\mu = 0$$

- Thus, $(\boldsymbol{A}^\top\boldsymbol{A})\boldsymbol{v} = \lambda\boldsymbol{v}$, i.e., v_2 is an eigenvector of $\boldsymbol{A}^\top\boldsymbol{A}$, which is also a singular vector of $\boldsymbol{A}$
- For the decomposition $\boldsymbol{A} = \sum_{i=1}^n \sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^\top$, notice that $\{\boldsymbol{v}_1, \dots, \boldsymbol{v}_n\}$ is an orthonormal basis of $\mathbb{R}^n$, and

$$\left(\sum_{i=1}^n \sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^\top\right)\boldsymbol{v}_k = \sigma_k\boldsymbol{u}_k = \|\boldsymbol{A}\boldsymbol{v}_k\|_2\frac{\boldsymbol{A}\boldsymbol{v}_k}{\|\boldsymbol{A}\boldsymbol{v}_k\|_2} = \boldsymbol{A}\boldsymbol{v}_k \qquad \forall k \in [n]$$

■

Singular Value Decomposition

In general, for $\boldsymbol{A} = \sum_{i=1}^{n} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathsf{T}}$ with

- $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$
- $\{\boldsymbol{u}_i\}$ and $\{\boldsymbol{v}_i\}$ are orthonormal

Then we say it is a singular value decomposition of $\boldsymbol{A}$

- $V_k = \mathrm{span}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ gives the best-fit subspace for $\boldsymbol{A}_{(1)}, \ldots, \boldsymbol{A}_{(m)} \in \mathbb{R}^n$

Theorem (Best rank-k approximation). Among all rank k matrices $\widetilde{\boldsymbol{A}}$, the matrix $\boldsymbol{A}_k = \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathsf{T}}$ is the one which minimizes $\left\|\boldsymbol{A} - \widetilde{\boldsymbol{A}}\right\|_F^2$. Moreover, $\|\boldsymbol{A} - \boldsymbol{A}_k\|_F^2 = \sum_{i=k+1}^{n} \sigma_i^2$

Singular Value Decomposition

Theorem (Best rank-k approximation). Among all rank k matrices $\widetilde{\boldsymbol{A}}$, the matrix $\boldsymbol{A} = \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\top}$ is the one which minimizes $\left\|\boldsymbol{A} - \widetilde{\boldsymbol{A}}\right\|_F^2$. Moreover, $\|\boldsymbol{A} - \boldsymbol{A}_k\|_F^2 = \sum_{i=k+1}^{n} \sigma_i^2$

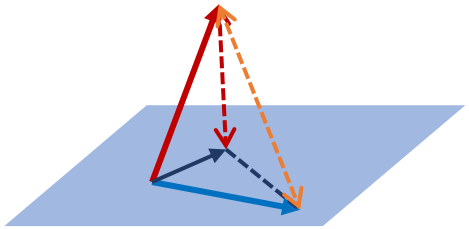
Proof.

- Note that

$$\left\|\boldsymbol{A} - \widetilde{\boldsymbol{A}}\right\|_F^2 = \sum_{i=1}^{m} \left\|\boldsymbol{A}_{(i)} - \widetilde{\boldsymbol{A}}_{(i)}\right\|_2^2$$

- Since $\operatorname{rank}\left(\widetilde{\boldsymbol{A}}\right) = k$, let V be the k-dimensional subspace spanned by the rows of $\widetilde{\boldsymbol{A}}$
- $\left\|\boldsymbol{A}_{(i)} - \widetilde{\boldsymbol{A}}_{(i)}\right\|_2^2 \geq \left\|\boldsymbol{A}_{(i)} - \mathcal{P}_V\left(\boldsymbol{A}_{(i)}\right)\right\|_2^2 = d\left(\boldsymbol{A}_{(i)}, V\right)^2$
- Then, we have

$$\left\|\boldsymbol{A} - \widetilde{\boldsymbol{A}}\right\|_F^2 = \sum_{i=1}^{m} d\left(\boldsymbol{A}_{(i)}, V\right)^2 \geq \|\boldsymbol{A} - \boldsymbol{A}_k\|_F^2$$



■

Singular Value Decomposition

Matrix norm and singular values

- Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ with singular values $\sigma_1 \geq \cdots \geq \sigma_n$
- The spectral norm $\|\boldsymbol{A}\| = \sigma_1$, i.e., the ℓ_∞-norm of the singular values:

$$\|\boldsymbol{A}\| = \sqrt{\lambda_{\max}(\boldsymbol{A}^\top \boldsymbol{A})} = \sigma_1 = \sup_{\boldsymbol{x} \in \mathbb{R}^m, \boldsymbol{y} \in \mathbb{R}^n : \|\boldsymbol{x}\|_2 = 1, \|\boldsymbol{y}\|_2 = 1} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y}$$

- The Frobenius norm is the ℓ_2-norm of the singular values:

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |A_{ij}|^2} = \sqrt{\mathrm{tr}[\boldsymbol{A}^\top \boldsymbol{A}]} = \sqrt{\sum_{i=1}^{n} \sigma_i^2}$$

- The Schatten *p*-norm is defined to be the ℓ_p-norm of the singular values

Today's Lecture

- Introduction to Spectral Method
- SVD and Best-Fit Subspace
- k-Means Clustering
- Complexity of SVD

k-Means Clustering

Given m points $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_m\} \subset \mathbb{R}^n$, find k points $\mathcal{C} = \{\boldsymbol{c}_1, \dots, \boldsymbol{c}_k\} \subset \mathbb{R}^n$ such that

$$f_{\mathcal{X}}(\mathcal{C}) \coloneqq \sum_{i=1}^{m} \operatorname{dist}(\boldsymbol{x}_i, \mathcal{C})^2$$

is minimized

- **NP**-hard even for $k = 2$
- The optimal solution is obtained by k clusters of $\mathcal{X} = S_1 \uplus \cdots \uplus S_k$, and the cluster center $\boldsymbol{c}_j$ is the centroid of the points in S_j
  - For any set $\mathcal{S} = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_r\}$, let $\overline{\boldsymbol{y}} \coloneqq \frac{1}{r}(\boldsymbol{y}_1 + \cdots + \boldsymbol{y}_r)$. Then, for any $\boldsymbol{c} \in \mathbb{R}^n$, we have

$$\sum_{i=1}^{r} \|\boldsymbol{y}_i - \boldsymbol{c}\|_2^2 = \sum_{i=1}^{r} \|\boldsymbol{y}_i - \overline{\boldsymbol{y}}\|_2^2 + r\|\overline{\boldsymbol{y}} - \boldsymbol{c}\|_2^2 \geq \sum_{i=1}^{r} \|\boldsymbol{y}_i - \overline{\boldsymbol{y}}\|_2^2$$

k-Means Clustering

Let $g_X(V)$ be the sum of squared distance to a k-dimensional subspace V:

$$g_X(V) \coloneqq \sum_{i=1}^{m} \mathrm{dist}(\boldsymbol{x}_i, V)^2$$

$g_X(V)$ can be minimized using SVD. Moreover, it provides a lower bound for k-means clustering:

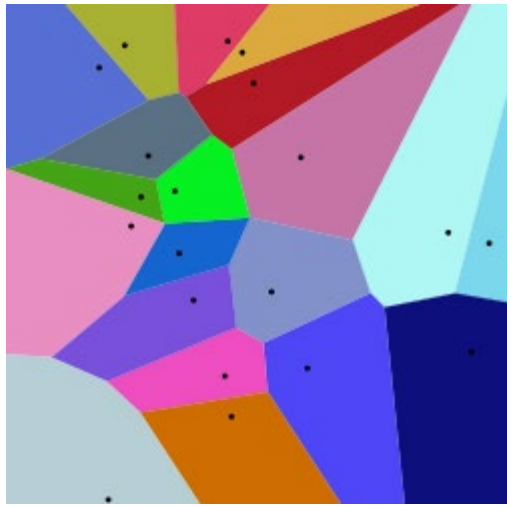
$$f_X(\mathcal{C}) \geq g_X\big(\mathrm{span}(\mathcal{C})\big)$$

- The best-fit subspace problem can be regarded as a relaxation of the k-means problem
- We can design a 2-approximation algorithm for k-means by:
  1. Identifying a best-fit subspace to reduce the dimensionality
  2. Exactly solving the k-means problem in this lower-dimensional subspace

k-Means Clustering

Theorem (Exact k-means)**.** The k-means problem can be solved in $\mathcal{O}\left(m^{k^2 d/2}\right)$ time when the input dataset $\mathcal{X} \subset \mathbb{R}^d$ of size m

- The most brute-force method is to enumerate all possible ways to form k clusters, and the complexity is $\mathcal{O}(k^m md) = \exp\left(\mathcal{O}(m \log k)\right) \gg \exp\left(\mathcal{O}(k^2 d \log m)\right)$
- k centers $\Leftrightarrow$ Voronoi diagram



$\mathbb{R}^d$ is partitioned into k cells:

$$R_i \coloneqq \left\{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{c}_i\|_2 \leq \left\|\boldsymbol{x} - \boldsymbol{c}_j\right\|_2 \ \forall j \neq i\right\}$$

- There are at most $\binom{k}{2}$ faces (boundary hyperplanes)
- We can enumerate these $\mathcal{O}(k^2)$ hyperplanes

k-Means Clustering

Since only the cluster assignments of the m points matter, we may freely translate or rotate any boundary hyperplane as long as it does not cross any data point

Algorithm (Voronoi-k-means).

1. Enumerate the sets of t hyperplanes such that each hyperplane contains exactly d data points, where $k \leq t \leq k^2/2$, and the total number of such sets is:

$$\sum_{t=k}^{k^2/2} \binom{\binom{m}{d}}{t} = \mathcal{O}\left(m^{dk^2/2}\right)$$

2. Check for each set whether the t hyperplanes form k cells
3. If so, assign the boundary data points to the left or the right cells

$$\#\text{assignments} = 2^{td}$$

4. Compute the centroid of the data points in each cell as the centers

k-Means Clustering

Algorithm (2-approximation for k-means).

1. Compute the rank-k best-fit subspace V by SVD
2. Let $\boldsymbol{x}_1^V, \boldsymbol{x}_2^V, \dots, \boldsymbol{x}_m^V$ be the projection of the dataset to V
3. Output $\mathcal{C}^V \leftarrow$ Voronoi-k-means$(\{\boldsymbol{x}_1^V, \boldsymbol{x}_2^V, \dots, \boldsymbol{x}_m^V\})$

- Let OPT be the optimal value of the k-means problem. We have

$$\mathrm{OPT} \geq \sum_{i=1}^{m} \left\| \boldsymbol{x}_i - \boldsymbol{x}_i^V \right\|_2^2$$

- We also have

$$\mathrm{OPT} = \sum_{i=1}^{m} d(\boldsymbol{x}_i, \mathcal{C}_\star)^2 \geq \sum_{i=1}^{m} d\left(\boldsymbol{x}_i^V, \mathcal{C}_\star^V\right)^2 \geq \sum_{i=1}^{m} d\left(\boldsymbol{x}_i^V, \mathcal{C}^V\right)^2$$

k-Means Clustering

- Let OPT be the optimal value of the k-means problem. We have

$$\mathrm{OPT} \geq \sum_{i=1}^{m} \left\| \boldsymbol{x}_i - \boldsymbol{x}_i^V \right\|_2^2$$

- We also have

$$\mathrm{OPT} = \sum_{i=1}^{m} d(\boldsymbol{x}_i, \mathcal{C}_\star)^2 \geq \sum_{i=1}^{m} d\left(\boldsymbol{x}_i^V, \mathcal{C}_\star^V\right)^2 \geq \sum_{i=1}^{m} d\left(\boldsymbol{x}_i^V, \mathcal{C}^V\right)^2$$

- Summing them together, we get

$$2\mathrm{OPT} \geq \sum_{i=1}^{m} \left(\left\| \boldsymbol{x}_i - \boldsymbol{x}_i^V \right\|_2^2 + d\left(\boldsymbol{x}_i^V, \mathcal{C}^V\right)^2 \right) = \sum_{i=1}^{m} d(\boldsymbol{x}_i, \mathcal{C}^V)^2 =: \mathrm{ALG}$$

- Thus, the algorithm can produce a 2-approximate solution in time

$$\mathcal{O}\left(mn^2 + m^{k^3/2}\right)$$

Today's Lecture

- Introduction to Spectral Method
- SVD and Best-Fit Subspace
- k-Means Clustering
- Complexity of SVD

Complexity of SVD

Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $m \geq n$

Practical version

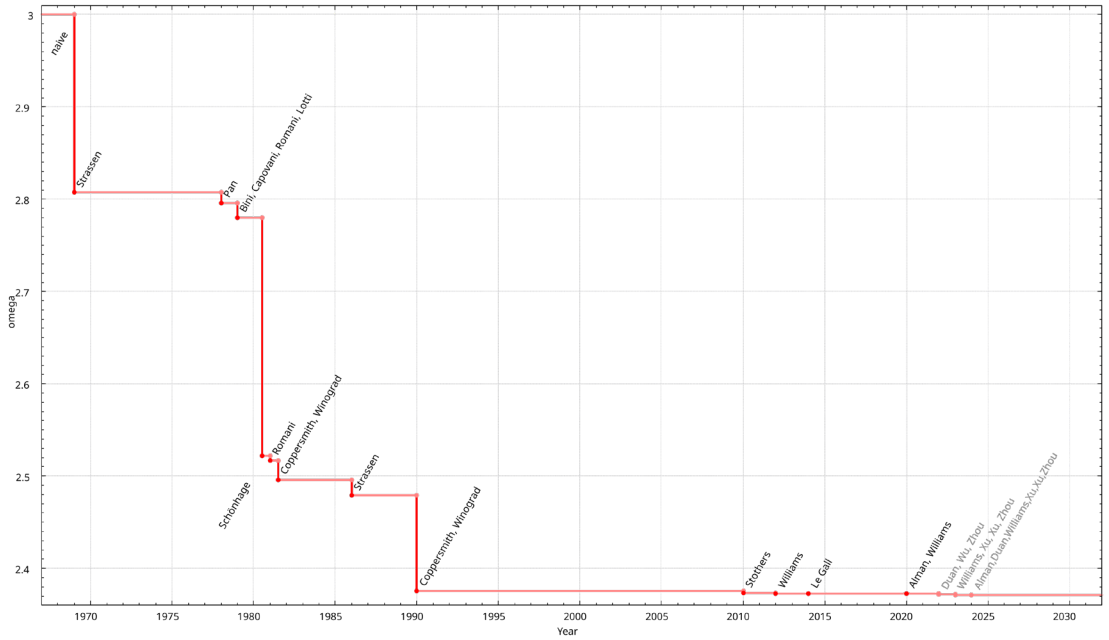
- Complexity: $\mathcal{O}(mn^2 + n^3)$
- $\boldsymbol{A} \longrightarrow \boldsymbol{A}^\top \boldsymbol{A} \longrightarrow$ eigendecomposition $\longrightarrow$ SVD
- LAPACK (used by Matlab and Numpy): highly optimized version of this approach

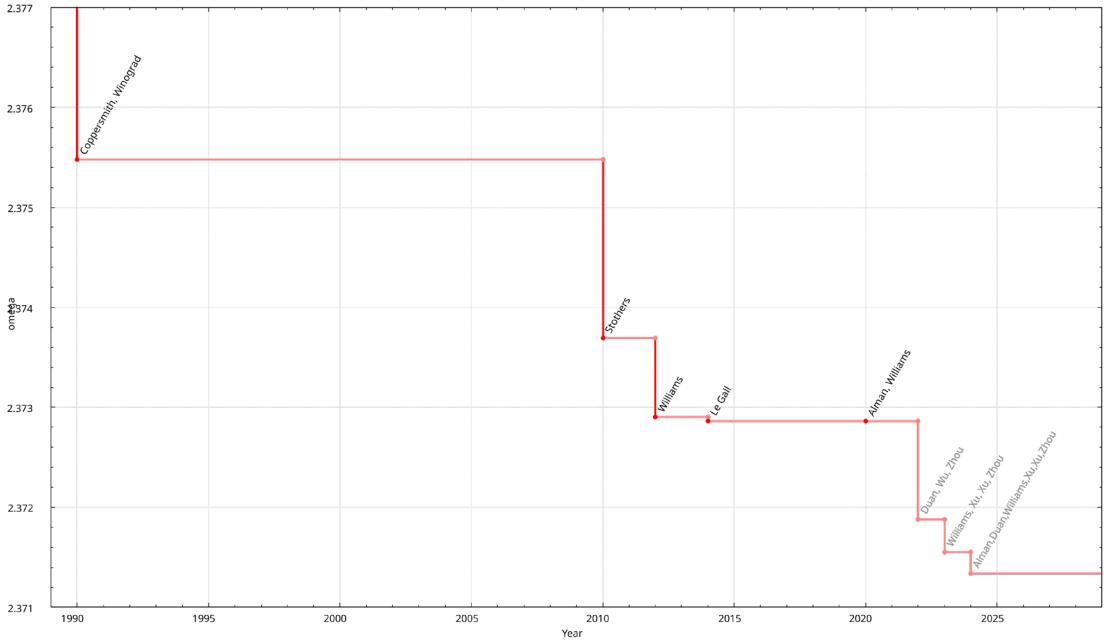
Complexity of SVD

Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $m \geq n$

Theoretical (TCS) version

- Complexity: $\mathcal{O}^*(mn^{\omega-1})$ where $\omega \approx 2.371339$ is the fast matrix multiplication
- See (Kacham-Woodruff '24, Appendix A) and (Banks-Garza-Vargas-Kulkarni-Srivastava '23)





Complexity of SVD

Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $m \geq n$. Suppose we only want to approximate the top-k singular values and singular vectors (e.g., low-rank approximation or PCA)

LazySVD

- $\boldsymbol{A} \longrightarrow$ top singular vector $\boldsymbol{v}_1 \longrightarrow$ left-project $(\boldsymbol{I} - \boldsymbol{v}\boldsymbol{v}^\top)\boldsymbol{A} \longrightarrow$ repeat k times
- Complexity: $\mathcal{O}\left(\frac{k \cdot \mathrm{nnz}(A) + k^2 n}{\sqrt{\epsilon}}\right)$ (Allen-Zhu-Li '17, Kacham-Woodruff '24)

How to compute the top-singular vector?

- Allen-Zhu-Li's algorithms use convex optimizers including accelerated gradient descent (AGD) and accelerated stochastic variance reduced gradient (SVRG)
- We'll see a simpler approach: the Power Method

Detour: Computational Models

In optimization and numerical linear algebra, we are dealing with real number operations

There are various computational models:

- Exact real arithmetic: infinite precision
- Variable precision rational arithmetic: rationals are stored exactly as numerators and denominators
- Finite precision arithmetic: real numbers are rounded to a fixed number of bits which may depend on the input size and accuracy

The third model is more realistic than the first, and potentially more efficient in terms of the bit complexity than the second

Detour: Finite Precision Arithmetic

- Numbers are stored and manipulated approximately up to some machine precision
$$\varepsilon \coloneqq \varepsilon_{\text{mach}}(n, \delta) > 0$$
which depends on the instance size n and the desired accuracy δ
- Every $x \in \mathbb{R}$ is stored as $\text{fl}(x) = (1 + \Delta)x$ for some adversarially chosen $\Delta \in \mathbb{R}$ such that $|\Delta| \leq \varepsilon$
- For each arithmetic operator $\circ \in \{+, -, \times, \div\}$, it holds that
$$\text{fl}(x \circ y) = (1 + \Delta)(x \circ y), \qquad |\Delta| \leq \varepsilon$$
- $\text{fl}(\sqrt{x}) = (1 + \Delta)\sqrt{x}$ for $|\Delta| \leq \varepsilon$

- The bit lengths of numbers stored in this form is fixed at $\log_2(1/\varepsilon)$
- The bit complexity of an algorithm is the number of arithmetic operations times $\tilde{\mathcal{O}}(\log_2(1/\varepsilon))$, the running time of standard floating-point arithmetic

Detour: Finite Precision Arithmetic

- An iterative algorithm that can be implemented in finite precision (i.e., bit length $\mathrm{polylog}(n/\delta)$) is called numerically stable
- Let $\boldsymbol{f}: \mathbb{R}^n \to \mathbb{R}^m$ be a function. Its condition number is defined as the smallest $\kappa_f \geq 0$ such that

$$\underbrace{\frac{\|\boldsymbol{f}(\boldsymbol{x}+\boldsymbol{\delta x})-\boldsymbol{f}(\boldsymbol{x})\|_2}{\|\boldsymbol{f}(\boldsymbol{x})\|_2}}_{\text{Forward error}} \leq \kappa_f \cdot \underbrace{\frac{\|\boldsymbol{\delta x}\|_2}{\|\boldsymbol{x}\|_2}}_{\text{Backward error}} + \mathcal{O}\left(\left(\frac{\|\boldsymbol{\delta x}\|_2}{\|\boldsymbol{x}\|_2}\right)^2\right)$$

- Backward stability is the gold standard for algorithms
  - Suppose an algorithm outputs $\boldsymbol{y}$
  - $\boldsymbol{\delta x} \coloneqq \arg\min\{\|\boldsymbol{\Delta}\|_2 : \boldsymbol{f}(\boldsymbol{x}+\boldsymbol{\Delta}) = \boldsymbol{y}\}$
  - Backward error $\coloneqq \frac{\|\boldsymbol{\delta x}\|_2}{\|\boldsymbol{x}\|_2} \leq \mathrm{poly}(n)\cdot\varepsilon + \mathcal{O}(\epsilon^2)$, then backward stable

Detour: Finite Precision Arithmetic

Examples

- Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$, there is an algorithm that outputs $\boldsymbol{C} \in \mathbb{R}^{n\times n}$ such that
$$\|\boldsymbol{C} - \boldsymbol{AB}\| \leq \mathrm{poly}(n) \cdot \varepsilon \cdot \|\boldsymbol{A}\|\|\boldsymbol{B}\|$$
on a floating-point machine with precision ε, in $\mathcal{O}^*(n^\omega)$ arithmetic operations (Demmel-Dumitriu-Holtz-Kleinberg '07)

- Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be an invertible matrix with condition number $\kappa(A) \coloneqq \frac{\sigma_1}{\sigma_n}$. There is an algorithm that outputs $C \in \mathbb{R}^{n\times n}$ such that
$$\|\boldsymbol{C} - \boldsymbol{A}^{-1}\| \leq \kappa(A)^{\mathrm{poly}\log n} \cdot \varepsilon \cdot \|\boldsymbol{A}^{-1}\|$$
on a floating-point machine with precision ε, in $\mathcal{O}^*(n^\omega)$ arithmetic operations (Demmel-Dumitriu-Holtz '07)

Complexity of SVD: Power Method

Let's first assume that $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is square symmetric and has the same left- and right-singular vectors:

$$\boldsymbol{A} = \sum_{i=1}^{r} \sigma_i \boldsymbol{v}_i \boldsymbol{v}_i^\top$$

- Squaring it gives

$$\boldsymbol{A}^2 = \left(\sum_{i=1}^{r} \sigma_i \boldsymbol{v}_i \boldsymbol{v}_i^\top\right)\left(\sum_{j=1}^{r} \sigma_j \boldsymbol{v}_j \boldsymbol{v}_j^\top\right) = \sum_{i,j=1}^{r} \sigma_i \sigma_j \, \boldsymbol{v}_i \left(\boldsymbol{v}_i^\top \boldsymbol{v}_j\right) \boldsymbol{v}_j^\top = \sum_{i=1}^{r} \sigma_i^2 \boldsymbol{v}_i \boldsymbol{v}_i^\top$$

- Multiplying k times gives

$$\boldsymbol{A}^k = \sum_{i=1}^{r} \sigma_i^k \boldsymbol{v}_i \boldsymbol{v}_i^\top, \qquad \frac{\boldsymbol{A}^k}{\|\boldsymbol{A}^k\|_F} = \sum_{i=1}^{r} \frac{\sigma_i^k}{\left(\sigma_1^{2k} + \cdots + \sigma_r^{2k}\right)^{1/2}} \boldsymbol{v}_i \boldsymbol{v}_i^\top \to \boldsymbol{v}_1 \boldsymbol{v}_1^\top \text{ if } \sigma_1 > \sigma_2$$

Complexity of SVD: Power Method

In general, let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ be rectangular. Then, we can apply the power method to $\boldsymbol{B} = \boldsymbol{A}^\top \boldsymbol{A}$:

$$\boldsymbol{B}^k = \left(\left(\sum_{i=1}^{r} \sigma_i \boldsymbol{v}_i \boldsymbol{u}_i^\top \right) \left(\sum_{j=1}^{r} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^\top \right) \right)^k = \left(\sum_{i=1}^{r} \sigma_i^2 \boldsymbol{v}_i \boldsymbol{v}_i^\top \right)^k = \sum_{i=1}^{r} \sigma_i^{2k} \boldsymbol{v}_i \boldsymbol{v}_i^\top$$

- As k increases, σ_i / σ_1 goes to zero, and $\boldsymbol{B}^k \approx \sigma_1^{2k} \boldsymbol{v}_1 \boldsymbol{v}_1^\top$
- However, matrix multiplication is expensive
- We can pick a vector x and compute $\boldsymbol{x}^{(k)} \coloneqq \boldsymbol{B}^k \boldsymbol{x} = \boldsymbol{A}^\top \boldsymbol{A} \boldsymbol{x}^{(k-1)}$
- Each iteration involves two Mat-Vec products, which take $\mathcal{O}\big(\mathrm{nnz}(\boldsymbol{A})\big)$ time
- $\boldsymbol{x}^{(k)} \approx \sigma_1^{2k} \boldsymbol{v}_1 \langle \boldsymbol{x}, \boldsymbol{v}_1 \rangle$. If $\langle \boldsymbol{x}, \boldsymbol{v}_1 \rangle \neq 0$, we can recover $\boldsymbol{v}_1$ from $\boldsymbol{x}^{(k)}$ by normalization

Complexity of SVD: Power Method

Lemma. Let $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$ be a unit d-dimensional vector picked at random from $\mathbb{S}^{n-1}$. The probability that $|x_1| \geq \frac{1}{20\sqrt{n}}$ is at least 0.9

Proof.

- Let $\alpha \coloneqq 1/(20\sqrt{n})$
- We first show that for a uniformly random $\boldsymbol{v} \sim B_2^n$, $\Pr[|v_1| \geq \alpha] \geq 0.9$
- Then, by taking $\boldsymbol{x} \coloneqq \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|_2}$, $\boldsymbol{x}$ is a uniformly random unit vector and $x_1 \geq v_1$

Complexity of SVD: Power Method

Lemma. Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ be a unit d-dimensional vector picked at random from $\mathbb{S}^{n-1}$. The probability that $|x_1| \geq \frac{1}{20\sqrt{n}}$ is at least 0.9

Proof.

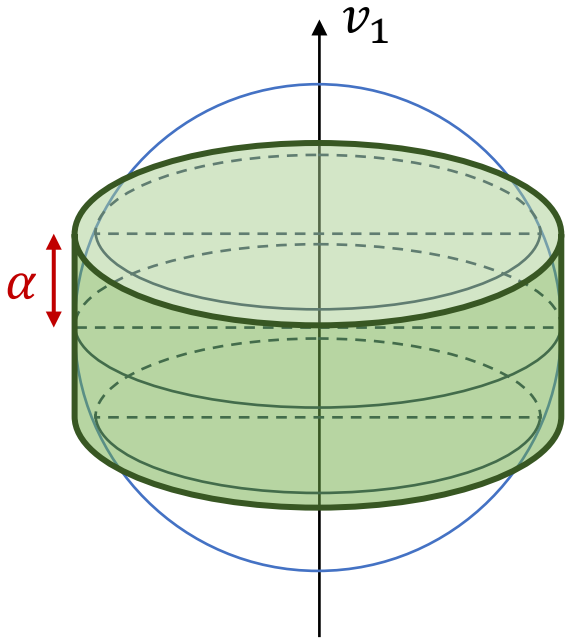
- Notice that

$$\Pr[|v_1| \leq \alpha] \leq \frac{2\alpha \cdot \mathrm{Vol}_{n-1}(B_2^{n-1})}{\mathrm{Vol}_n(B_2^n)}$$

$$\sim 2\alpha \left(\frac{2\pi e}{n-1}\right)^{(n-1)/2} \left(\frac{n}{2\pi e}\right)^{n/2}$$

$$= \frac{\alpha\sqrt{2n}}{\sqrt{\pi e}} \left(1 + \frac{1}{n-1}\right)^{(n-1)/2}$$

$$\sim \frac{\alpha\sqrt{2n}}{\sqrt{\pi}} = \frac{\sqrt{2n}}{20\sqrt{n\pi}} \approx 0.04$$



$$\Pr_{\boldsymbol{v} \sim B_2^n}[|v_1| \leq \alpha]$$

Complexity of SVD: Power Method

Theorem. Let $\boldsymbol{A}$ be an $m \times n$ matrix and $\boldsymbol{x}$ a random unit length vector. Let V be the space spanned by the left singular vectors of $\boldsymbol{A}$ corresponding to singular values greater than $(1-\epsilon)\sigma_1$. Let $k = \Omega\left(\frac{\ln(n/\epsilon)}{\epsilon}\right)$. Let $\boldsymbol{w}$ be unit vector after k iterations of the power method:

$$\boldsymbol{w} = \frac{\boldsymbol{x}^{(k)}}{\|\boldsymbol{x}^{(k)}\|_2} = \frac{(\boldsymbol{A}\boldsymbol{A}^\top)^k \boldsymbol{x}}{\|(\boldsymbol{A}\boldsymbol{A}^\top)^k \boldsymbol{x}\|_2}$$

Then, the probability that $d(\boldsymbol{w}, V) \geq \epsilon$ is at most $1/10$

- It does not rely on any explicit singular value gap assumption
- If we know that $\sigma_1 - \sigma_2 \geq \Delta$, then we can take $\epsilon = \Delta/2$ and recover v_1 with ϵ accuracy in $\tilde{\mathcal{O}}(\mathrm{nnz}(A)\Delta^{-1})$ time

Complexity of SVD: Power Method

Proof.

- Let $\boldsymbol{A} = \sum_{i=1}^{n} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathsf{T}}$ be the full SVD
- Expand $\boldsymbol{x}$ in the right singular vector basis: $\boldsymbol{x} = \sum_{i=1}^{n} c_i \boldsymbol{v}_i$ where $c_i \in \mathbb{R}$ and $\sqrt{c_1^2 + \cdots + c_n^2} = 1$
- Then, we have $\boldsymbol{x}^{(k)} = \sum_{i=1}^{n} \sigma_i^{2k} \boldsymbol{v}_i \langle \boldsymbol{x}, \boldsymbol{v}_i \rangle = \sum_{i=1}^{n} \sigma_i^{2k} c_i \boldsymbol{v}_i$
- Since $\boldsymbol{x}$ is a random unit vector, and $\{v_1, \ldots, v_n\}$ is a fixed orthonormal basis of $\mathbb{R}^n$
- We get that $\boldsymbol{c} \coloneqq (c_1, \ldots, c_n)$ is a random unit vector
- The lemma implies that $|c_1| \geq \frac{1}{20\sqrt{n}}$ with probability ≥ 0.9
- Suppose $\sigma_1 \geq \cdots \geq \sigma_s \geq (1-\epsilon)\sigma_1 > \sigma_{s+1} \geq \cdots \geq \sigma_n$, then we have

$$\left\| \boldsymbol{x}^{(k)} \right\|_2^2 = \sum_{i=1}^{n} \sigma_i^{4k} c_i^2 \geq \sigma_1^{4k} c_1^2 \geq \frac{\sigma_1^{4k}}{400n}$$

Complexity of SVD: Power Method

Proof.

- Suppose $\sigma_1 \geq \cdots \geq \sigma_s \geq (1-\epsilon)\sigma_1 > \sigma_{s+1} \geq \cdots \geq \sigma_n$, then we have

$$\left\|\boldsymbol{x}^{(k)}\right\|_2^2 = \sum_{i=1}^{n} \sigma_i^{4k} c_i^2 \geq \sigma_1^{4k} c_1^2 \geq \frac{\sigma_1^{4k}}{400n}$$

- Thus, for $V = \mathrm{span}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_s\}$, we have

$$d\left(\boldsymbol{x}^{(k)}, V\right)^2 = \sum_{i=s+1}^{n} \sigma_i^{4k} c_i^2 \leq (1-\epsilon)^{4k} \sigma_1^{4k} \sum_{i=s+1}^{n} c_i^2 \leq (1-\epsilon)^{4k} \sigma_1^{4k}$$

- Hence,

$$d(\boldsymbol{w}, V) = \frac{d\left(x^{(k)}, V\right)}{\left\|x^{(k)}\right\|_2} \leq \frac{(1-\epsilon)^{2k}\sigma_1^{2k}}{\sigma_1^{2k}/20\sqrt{n}} = 20(1-\epsilon)^{2k}\sqrt{n} \leq \epsilon$$

  if we take $k = \Omega(\ln(n/\epsilon)/\epsilon)$

■