SMAI-S25-L11: PCA and SVD

C. V. Jawahar

IIIT Hyderabad

February 11, 2025

Recap:

- Problems of interest:
  - Learn a function $y = f(\mathbf{W}, \mathbf{x})$ from the data.
    - (a) Classification (b) Regression
  - Learn Feature Transformations $\mathbf{x}' = \mathbf{W}\mathbf{x}$ or $\mathbf{x}' = f(\mathbf{W}, \mathbf{x})$
    - (a) Feature Normalization (b) PCA
- Algorithms/Approaches:
  - Nearest Neighbour Algorithm
  - Linear Classification: $sign(\mathbf{w}^T\mathbf{x})$
  - Decide as ω_1 if $P(\omega_1|\mathbf{x}) \geq P(\omega_2|\mathbf{x})$ else ω_2.
  - Linear Regression: (a) closed form and (b) GD
- Supervised Learning:
  - Notion of Training, Validation and Testing
  - Performance Metrics
  - Low rank data matrix and SVD
  - Notion of Loss Function, (eg. MSE), Regularization.
  - Role of Optimization, Convex and non-Convex optimization
  - Closed form solution, Gradient Descent, Eigen vector solns.

PCA: Algorithm

1. Input: N samples of D dimension
2. Compute Covariance Matrix $$\Sigma = \frac{1}{N}\sum_{i=1}^{N}[x_i - \mu][x_i - \mu]^T$$
3. Compute Eigen Values and Eigen Vectors of Covariance Matrix
4. Select `d' eigen vectors corresponding to 'd' largest eigen values
5. Arrange them as rows of 'A' (a 'd' X 'D' matrix)
6. Find lower dimensional representations as: $x' = Ax$

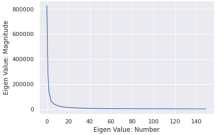
Eigen Faces

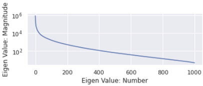
62 X 47 Image D = 2914; N = 1348
Vectors displayed as 2D Image

Mean
2914 X 1

Covariance Matrix is
2914 X 2914

First 16 Eigen Vectors
out of 2914





Log-scale

https://towardsdatascience.com/eigenfaces-recovering-humans-from-ghosts-17606c328184

23

Note on Reconstruction

Consider a sample in two coordinate systems. Let $\mathbf{x} = [x^1, x^2, \ldots, x^d]^T$

1. Standard basis: $\mathbf{u_1}, \mathbf{u_2}, \ldots \mathbf{u_d}$
   - $\mathbf{u_i} = [0, 0, \ldots, 1, \ldots, 0, 0,]^T$. WIth '1' at the i th place
   - $\mathbf{x} = (\mathbf{x}^T\mathbf{u}_1)\mathbf{u}_1 + (\mathbf{x}^T\mathbf{u}_2)\mathbf{u}_2 + \ldots + (\mathbf{x}^T\mathbf{u}_d)\mathbf{u}_d$
   - $\mathbf{x} = x^1\mathbf{u}_1 + x^2\mathbf{u}_2 + \ldots + x^d\mathbf{u}_d$
2. Defined by Eigen Vectors: $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_d}$
   - $(\lambda_i, \mathbf{v}_i)$ are the eigen value and eigen vectors of the covariance matrix.
   - $\mathbf{x} = (\mathbf{x}^T\mathbf{v}_1)\mathbf{v}_1 + (\mathbf{x}^T\mathbf{v}_2)\mathbf{v}_2 + \ldots + (\mathbf{x}^T\mathbf{v}_{d'})\mathbf{v}_{d'} + \ldots + (\mathbf{x}^T\mathbf{v}_d)\mathbf{v}_d$
   - $\mathbf{x} = \alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \ldots + \alpha_{d'}\mathbf{v}_{d'} + \ldots + \alpha_d\mathbf{v}_d$

If we reduce to d' dimension from d, we loose some information/signal.

PCA as Minimizing Reconstructing Loss

Let $\mathbf{u_1}, \ldots \mathbf{u_d}$ be d orthonormal vectors. $\mathbf{x}$ is $\sum_{i=1}^{d} \alpha_i \mathbf{u}_i$, where $\alpha_i = \mathbf{x}^T \mathbf{u_i}$. Problem: Find the dimension $\mathbf{u}$ that has highest variance (or minimal reconstruction error) or minimize $\mathbf{x} - \mathbf{u}\mathbf{u}^T\mathbf{x}$. Problem is to minimze:

$$\sum_{i=1}^{N} ||\mathbf{x_i} - \mathbf{u}\mathbf{u}^T\mathbf{x}_i||^2 = \sum_{i=1}^{N} \left(\mathbf{x}_i^T \mathbf{x}_i + (\mathbf{u}\mathbf{u}^T\mathbf{x}_i)^T (\mathbf{u}\mathbf{u}^T\mathbf{x}_i) - 2\mathbf{x}_i^T \mathbf{u}\mathbf{u}^T\mathbf{x}_i \right)$$

First term is positive and independent of $\mathbf{u}$. And $\mathbf{u}^T\mathbf{u} = 1$. Minimize:

$$\sum_{i=1}^{N} (\mathbf{x_i}^T \mathbf{u}\mathbf{u}^T \mathbf{u}\mathbf{u}^T \mathbf{x_i} - 2\mathbf{x}_i^T \mathbf{u}\mathbf{u}^T \mathbf{x}_i) = \sum_{i=1}^{N} -\mathbf{x}_i^T \mathbf{u}\mathbf{u}^T \mathbf{x}_i = \sum_{i=1}^{N} -\mathbf{u}^T \mathbf{x}_i \mathbf{x}_i^T \mathbf{u} = -\mathbf{u}^T \Sigma \mathbf{u}$$

Thus the problem reduces to:

$$\text{Maximize } \mathbf{u}^T \Sigma \mathbf{u} \text{ such that } \mathbf{u}^T \mathbf{u} = 1$$

This reduces the solution as the eigen vectors corresponding to the largest eigen values.

Eigen Decomposition

Popular eigen expansion

$$\Sigma = \sum_{i=1}^{d} \lambda_i \mathbf{v}_i \mathbf{v}_i^T$$

tells us what variation to retain and what to approximate.

$$\Sigma = \sum_{i=1}^{d'} \lambda_i \mathbf{v}_i \mathbf{v}_i^T + \sum_{i=d'+1}^{d} \lambda_i \mathbf{v}_i \mathbf{v}_i^T$$

We chose the reduced dimension based on the eigen value spectrum.
Also for a general PSD matrix

$$A = U \Lambda U^T$$

is known as the eigen decomposition.

SVD

Formally, the **singular value decomposition** (SVD) of an $m \times n$ real or complex matrix M is a factorization of the form

$$M = UDV^T$$

- U is $m \times n$, $U^T U = I_n$,
- D is diagonal $n \times n$ and
- V is $V^T V = VV^T = I_n$.

$$M = \sum_i D_i u_i v_i^T$$

SVD

$$\underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \bullet & & \\ & \bullet & \\ & & \bullet \\ & & \\ & & \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} \star & \star & \star \\ \star & \star & \star \\ \star & \star & \star \end{bmatrix}}_{V^T}$$

$$\underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \star & \star & \star \\ \star & \star & \star \\ \star & \star & \star \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \bullet & & & & \\ & \bullet & & & \\ & & \bullet & & \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \end{bmatrix}}_{V^T}$$

Numerical Example

Example 1

$$A = \begin{bmatrix} \mathbf{3} & \mathbf{0} \\ \mathbf{4} & \mathbf{5} \end{bmatrix}$$

$$\boldsymbol{U} = \frac{1}{\sqrt{10}}\begin{bmatrix} 1 & -3 \\ 3 & 1 \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \sqrt{45} & \\ & \sqrt{5} \end{bmatrix} \quad \boldsymbol{V} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}.$$

Example 2

$$\begin{bmatrix} -3 & 1 \\ 6 & -2 \\ 6 & -2 \end{bmatrix} = \begin{bmatrix} 1/3 & 2/\sqrt{5} & 2/\sqrt{5} \\ -2/3 & 1/\sqrt{5} & 0 \\ -2/3 & 0 & 1/\sqrt{5} \end{bmatrix} \begin{bmatrix} 3\sqrt{10} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -3/\sqrt{10} & 1/\sqrt{10} \\ 1/\sqrt{10} & 3/\sqrt{10} \end{bmatrix}^T$$

$$\begin{bmatrix} -3 & 1 \\ 6 & -2 \\ 6 & -2 \end{bmatrix} = \begin{bmatrix} 1/3 \\ -2/3 \\ -2/3 \end{bmatrix} 3\sqrt{10} \begin{bmatrix} -3/\sqrt{10} \\ 1/\sqrt{10} \end{bmatrix}^T = \begin{bmatrix} 1 \\ -2 \\ -2 \end{bmatrix} [-3, 1]$$

$$\sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathrm{T}} + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^{\mathrm{T}} = \frac{\sqrt{45}}{\sqrt{20}} \begin{bmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{3} & \mathbf{3} \end{bmatrix} + \frac{\sqrt{5}}{\sqrt{20}} \begin{bmatrix} \mathbf{3} & \mathbf{-3} \\ \mathbf{-1} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix} = A.$$

Eigen vs SVD

We know that:

- Eigen Decomposition of Symmetric Matrix **S**

$$\mathbf{S} = \mathbf{Q}\Lambda\mathbf{Q}^T = \sum_{i=1}^{n} \lambda_i \mathbf{q}_i \mathbf{q}_i^T$$

- SVD of **A**

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T = \sum_{i=1}^{n} d_i \mathbf{u}_i \mathbf{v}_i^T$$

- The non-zero singular values of M (found on the diagonal entries of D) are the square roots of the non-zero eigenvalues of both $M^T M$ and MM^T.

Low Rank Approximation using SVD

Problem: Given a matrix **A**, find its low rank (k) approximation.

1. Find SVD of **A** as $A \rightarrow UDV^T$
2. Make all singular values beyond k in D as zero to obtain $\hat{D}$
3. Reconstruct k rank matrix

$$\hat{A} \leftarrow U\hat{D}V^T$$

Latent Semantic Indexing (LSI)

Consider a term document matrix ($N \times d$) where there are N documents and d is the vocabulary size. We argued that this matrix is often low-rank in practice.

Consider a collection of documents from "Technology" and "Food".

- How many "concepts" topics are there? How do we find them?
- How do we handle words like "apple" which may have multiple meaning?

LSI

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

U	1	2	3	4	5
ship	−0.44	−0.30	0.00	0.00	0.00
boat	−0.13	−0.33	0.00	0.00	0.00
ocean	−0.48	−0.51	0.00	0.00	0.00
wood	−0.70	0.35	0.00	0.00	0.00
tree	−0.26	0.65	0.00	0.00	0.00

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00

Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

Σ_2	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00

LSI: Why and How LSI works?

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

C_2	d_1	d_2	d_3	d_4	d_5	d_6
ship	0.85	0.52	0.28	0.13	0.21	−0.08
boat	0.36	0.36	0.16	−0.20	−0.02	−0.18
ocean	1.01	0.72	0.36	−0.04	0.16	−0.21
wood	0.97	0.12	0.20	1.03	0.62	0.41
tree	0.12	−0.39	−0.08	0.90	0.41	0.49

- How close are document d_2 and d_3 in the original and new space?

Problem 1

If $\mathbf{A} = \mathbf{UDV^T}$, Find the SVD of $\mathbf{A}^{-1}$.

Problem 2

1. What is left and right inverse of a matrix? How do we compute using SVD?

Problem 3

If $\mathbf{A} = UDV^T$.

- What are the eigen values and eigen vectors of AA^T?
- What are the eigen values and eigen vectors of A^TA?

Problem 4

Consider a matrix A formed by first 25 integers in row order $1, \ldots, 25$. Find a rank-1 and rank-2 approximation of A. What is the error in this approximation? (use an online SVD tool for computing SVD)

Problem 5

(based on the LSI example) We have a large number of documents. They are represented in a Term Document Matrix. Given a query document, find the most semantically similar document. Given matrix **A** is a term document matrix (with columns as document d_1, d_2, d_3 etc.). Given a new document as $[1, 0, 1, 0, 0]^T$, find the cosine similarity with all the six documents and find the nearest document
in the latent space of dimension 2 as well as 1.

Hint: map query and documents as $d^T U_k D_K^{-1}$