SMAI-S25-L10: Principal Component Analysis (PCA)

C. V. Jawahar

IIIT Hyderabad

February 7, 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 (today)
- 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
  - Notion of Loss Function, (eg. MSE), Regularization.
  - Role of Optimization, Convex and non-Convex optimization
  - Closed form solution, Gradient Descent, Eigen vector solns.

PCA: Principal component Analysis

A classical, popular dimensionality reduction technique.

- Unsupervised
- Linear

1. PCA: Dimensions that preserve maximum variance
   - $\mathbf{x}' = \mathbf{W}\mathbf{x}$
   - Problem: How to find $\mathbf{W}$?
2. PCA as Compression
   - Dimensionality reduction that allow minimal loss in the data.
3. Eigen Faces
   - A powerful application of PCA
   - Face representation and compression.

Recap: Optimization problems with EVec as Solutions

Problem: Maximize $\mathbf{w}^T\mathbf{A}\mathbf{w}$ such that $\mathbf{w}^T\mathbf{w} = 1$ (or $||\mathbf{w}|| = 1$)

We form an objective with the help of a lagrangian (λ) as

$$J(\mathbf{w}, \lambda) = \mathbf{w}^T\mathbf{A}\mathbf{w} - \lambda(\mathbf{w}^T\mathbf{w} - 1)$$

Differentiating wrt $\mathbf{w}$ and equating to zero leads to:

$$\mathbf{A}\mathbf{w} = \lambda\mathbf{w}$$

Soln: $\mathbf{w}$ is the eigen vector corresponding to the largest eigen value.

Discussions

1. If we are given N points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N\}$ in d dimension (say $d = 2$), what will be a good representing point $\mathbf{p}$? (hint: optimize the sum of square distance to all the points!)
2. If we are given N points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N\}$ in d dimension (say d=1), what will be a good representing line (in 2D) (or hyper place in general) $\mathbf{p}$? (hint: optimize the sum of square distance to all the points!)
3. What happens when we approximate a point by projecting to a line? What is the approximation error? (If we approximate a d-dimensional data in d' dimension, (where $d' < d$) what is the error of approximation or dimensionality reduction?

Fitting a line that minimize Orthogonal Distances[1]

We want to find $\mathbf{w}$ that define the line and norm 1.0. Let us consider that all samples are mean subtracted.

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

Alternatively,

$$\max \sum_{i=1}^{N} (\mathbf{w}^T \mathbf{x_i})^2 \text{ subject to: } \mathbf{w}^T \mathbf{w} = 1$$

$$\max \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} \text{ subject to: } \mathbf{w}^T \mathbf{w} = 1$$

Solution to this is the eigen vector corresponding to the largest eigen value of $\mathbf{X}^T\mathbf{X}$ or $\mathbf{\Sigma}$

[1] https://www.youtube.com/watch?v=likh0NUdvnc

Appreciating PCA

Maximum Variance Direction: 1st PC a vector v such that projection on to this vector capture maximum variance in the data (out of all possible one dimensional projections)

$$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{v}^T\mathbf{x}_i)^2 = \mathbf{v}^T\mathbf{X}\mathbf{X}^T\mathbf{v}$$

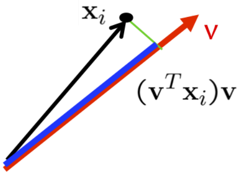
Minimum Reconstruction Error: 1st PC a vector v such that projection on to this vector yields minimum MSE reconstruction

$$\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i - (\mathbf{v}^T\mathbf{x}_i)\mathbf{v}\|^2$$

blue² + green² = black²

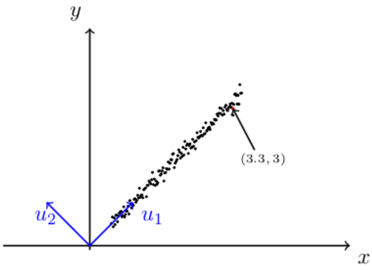
black² is fixed (it's just the data)

So, maximizing blue² is equivalent to minimizing green²



Slide from Nina Balcan

Reconstruction: Numerical Example



- $u_1 = [1, 1]$ and $u_2 = [-1, 1]$ are the new basis vectors
- Let us convert them to unit vectors $u_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$ & $u_2 = \begin{bmatrix} \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$

- Consider the point $x = [3.3, 3]$ in the original data
- $\alpha_1 = x^T u_1 = 6.3/\sqrt{2}$
  $\alpha_2 = x^T u_2 = -0.3/\sqrt{2}$
- the perfect reconstruction of x is given by (using $n = 2$ dimensions)

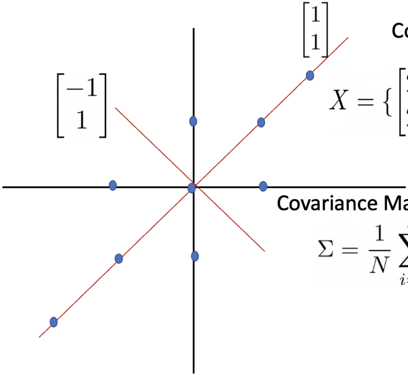
$$x = \alpha_1 u_1 + \alpha_2 u_2 = \begin{bmatrix} 3.3 & 3 \end{bmatrix}$$

- But we are going to reconstruct it using fewer (only $k = 1 < n$ dimensions, ignoring the low variance u_2 dimension)

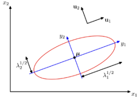
$$\hat{x} = \alpha_1 u_1 = \begin{bmatrix} 3.15 & 3.15 \end{bmatrix}$$

(reconstruction with minimum error)

Worked Out Example - I



Consider nine samples in 2D



$$X = \left\{ \begin{bmatrix}2\\2\end{bmatrix} \begin{bmatrix}1\\1\end{bmatrix} \begin{bmatrix}1\\0\end{bmatrix} \begin{bmatrix}0\\1\end{bmatrix} \begin{bmatrix}0\\0\end{bmatrix} \begin{bmatrix}-1\\0\end{bmatrix} \begin{bmatrix}0\\-1\end{bmatrix} \begin{bmatrix}-1\\-1\end{bmatrix} \begin{bmatrix}-2\\-2\end{bmatrix} \right\}$$

Covariance Matrix:

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}[x_i - \mu][x_i - \mu]^T$$

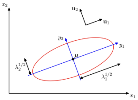
$$\Sigma = \frac{1}{9}\begin{bmatrix}12 & 10\\10 & 12\end{bmatrix}$$

Eigen vectors: $\begin{bmatrix}1\\1\end{bmatrix}$ and $\begin{bmatrix}-1\\1\end{bmatrix}$

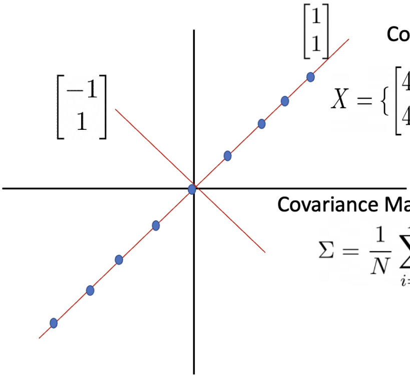
Eigen values: $\lambda_1 = 22$ and $\lambda_2 = 2$

Note: Eigen vectors are not normalized for simplicity

Worked Out Example - II



Consider nine samples in 2D



$$X = \left\{ \begin{bmatrix}4\\4\end{bmatrix} \begin{bmatrix}3\\3\end{bmatrix} \begin{bmatrix}2\\2\end{bmatrix} \begin{bmatrix}1\\1\end{bmatrix} \begin{bmatrix}0\\0\end{bmatrix} \begin{bmatrix}1\\1\end{bmatrix} \begin{bmatrix}-1\\-1\end{bmatrix} \begin{bmatrix}-2\\-2\end{bmatrix} \begin{bmatrix}-3\\-3\end{bmatrix} \begin{bmatrix}-4\\-4\end{bmatrix} \right\}$$

Covariance Matrix:

$$\Sigma = \frac{1}{N} \sum_{i=1}^{N} [x_i - \mu][x_i - \mu]^T$$

$$\Sigma = \frac{1}{9} \begin{bmatrix} 60 & 60 \\ 60 & 60 \end{bmatrix}$$

Eigen vectors: $\begin{bmatrix}1\\1\end{bmatrix}$ and $\begin{bmatrix}-1\\1\end{bmatrix}$

Eigen values: $\lambda_1 = 120$ and $\lambda_2 = 0$

Note: Eigen vectors are not normalized for simplicity

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: $x' = Ax$

How much to Reduce?

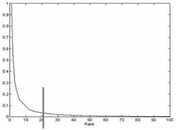
How many Eigen Values?

How many eigen values are required so that "most" of the information in the covariance/data is preserved? Consider the popular eigen expansion:

$$\Sigma = \sum_{i=1}^{D} \lambda_i v_i v_i^T$$
$$= \lambda_1 v_1 v_1^T + \lambda_2 v_2 v_2^T + \cdots + \lambda_{N-1} v_{N-1} v_{N-1}^T + \lambda_N v_N v_N^T$$

Eigen values (Lambda) are in decreasing order. Even if we discard the small eigen values, most information is preserved.

Typical Eigen Value Spectrum



$$d = \min_p \frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{D} \lambda_i} \geq 0.95$$

18

Eigen Faces

- All our faces look very different. But are they so different? Consider all our frontal faces are of size 100×100 or of size 10^4.
  - What will be the mean face?
  - What about eigen values, rank of Σ
  - What about eigen vectors?
- Is it possible to develop a representation for human faces? (for solving tasks like face recognition)
  - That is compact (much smaller than 10^4)
  - Different for different faces.
  - How to formulate it as $\mathbf{x}' = \mathbf{W}\mathbf{x}$

Faces and Eigen Vectors

Representation and Reconstruction of Face from 16 EVs

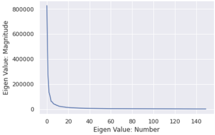
Eigen Faces

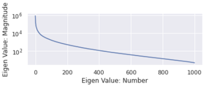
62 X 47 Image D = 2914; N = 1348
Vectors displayed as 2D Image

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

How many coeff. are required to represent a block?

144 to 60, 16, 6, 3

Figure: Original, Blocks of size 12 X 12 in 60, 16, 6 and 3

Problem 1

Consider images of size 100×100 and we have 200 such images. Assume means are subtracted.

1. What is the size of the covariance matrix?
2. What is the rank of the covariance matrix? (guess!)
3. What is the size of XX^T and X^TX and what are their ranks?
4. How are the Eigen values of X^TX and XX^T related?
5. How are the Eigen vectors of X^TX and XX^T related?

Problem 2

Consider images of size 100×100 ($O(10^4)$) and we have 200 ($O(10^2)$) such images. We know that computing eigen vectors is a costly operation. How do we compute $\mathbf{W}$?

Problem 3

In the context of regression (assume $\mathbf{x} \in R^1$ i.e., only one feature):

1. We know how to fit a line, even if it is not passing through origin, with a model $y = \mathbf{w}^T \mathbf{x}'$. where $\mathbf{x}'$ is defined as $\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$
2. How do we model the problem of fitting a quadratic (say a parabola) given a set of points?. What is $\mathbf{x}$? Is there a closed form expression?

Problem - 4

Matrix Completion Problem: Can we guess/compute/complete the missing elements of the matrix:

$$\begin{bmatrix} ? & 1 & 7 & ? & ? \\ ? & ? & ? & 8 & ? \\ 18 & ? & ? & 12 & 6 \\ ? & ? & ? & ? & 2 \\ ? & ? & 21 & 6 & ? \end{bmatrix}$$

if we know that this is a rank-1 matrix (or every row is a multiple of each other)[2]
Consider the following problem: Expalin the notations and what we try to find. (Q: What is this summation over? What is the min over?)

$$\min \sum_i \sum_j (A_{ij} - B_{ij})^2 \ s.t \ rank(B) = 1$$

[2] Read later: https://web.stanford.edu/class/cs168/l/l9.pdf

Problem - 5

Continuing the numerical example we saw in the class with eigen values and eigen vectors. Consider three points $\mathbf{x}_1 = [5.0, 5.0]^T$, $\mathbf{x}_2 = [3.0, 3.3]^T$, $\mathbf{x}_3 = [-3.0, 0.0]^T$

- Find the <u>low</u> dimensional $(d = 1)$ representations $\mathbf{x}' = \mathbf{W}\mathbf{x}$ What is $\mathbf{W}$?
- Find the reconstructed point and show the "error" magnitude in this new representation