Covariance / correlation matrix

↳ standardize data

↳ create cov matrix.

Covariance: Dataset with m samples, n features

$$\text{cov}(x_a, x_b) = \frac{1}{m} \sum_{i=1}^{m} (q_{i,a} - \bar{q}_a)(q_{i,b} - \bar{q}_b)$$

10 features ⇒ cov matrix: 10×10

XX^T ⇒ No. of samples × No. of samples.

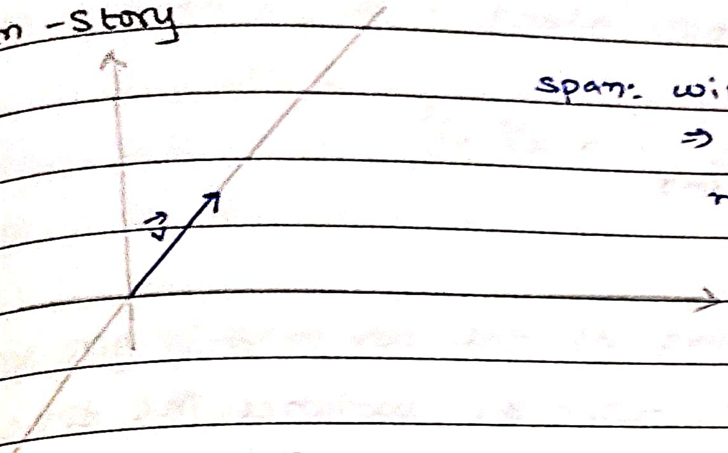
$$X = \begin{array}{cc} & \begin{array}{ccccc} x_1 & x_2 & x_3 & \cdots & x_n \end{array} \\ \begin{array}{c} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_m \end{array} & \begin{bmatrix} q_{11} & q_{12} & q_{13} & & q_{1n} \\ q_{21} & & & & \\ & & & & \\ & & & & \\ q_{m1} & & & & q_{mn} \end{bmatrix} \end{array}$$

$$C = \begin{bmatrix} \text{cov}(x_1, x_1) & \text{cov}(x_1, x_2) & \cdots & \text{cov}(x_1, x_n) \\ \text{cov}(x_2, x_1) & \text{cov}(x_2, x_2) & \cdots & \text{cov}(x_2, x_n) \\ \vdots & & & \\ \text{cov}(x_n, x_1) & \text{cov}(x_n, x_2) & \cdots & \text{cov}(x_n, x_n) \end{bmatrix} = \frac{1}{m} X^T X$$

n-dim Cov Matrix

Assume the matrix is cente

Eigen - story



span: will be on the same line
⇒ same dir^n
multiply with any number.

what happens with a matrix operation on the vector?

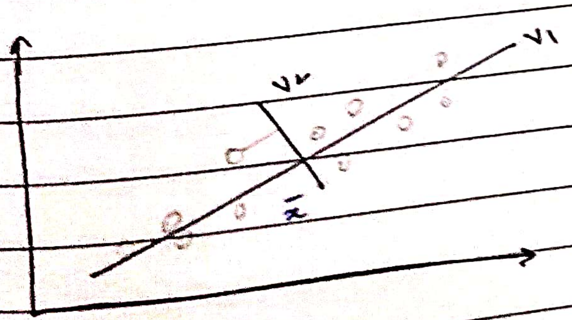
$$A\vec{v} = \begin{bmatrix} 3 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 8 \\ 5 \end{bmatrix}$$

- The vectors change their directions most of the time [the span changes]
- we have special cases where the vectors stay on the span [only getting scaled].

$$A\vec{v} = \begin{bmatrix} 1 & 3 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 4 \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

- The vector here is the eigenvector & the scaling 4 is the eigenvalue.

DEMO - PCA



$v_1 \rightarrow$ Most distribution and spread

→ consider the variation along a dir^n v among all the points

$$var(v) = \frac{1}{n} \sum_{x \text{ points}} |(x-\bar{x})^T \cdot v|^2$$

NOTE: The E. vectors of the cov matrix give you the direction that maximizes the variance. The direction of the green line is where the variance is max. Compare that with the projection on the V_2: the spread is small

→ Unit vector v maximizes var:

$$V_1 = \max_v \{var(v)\}$$

PCA :

large E. value capturing alot of variance.

most of the sample variance.

How many PC's

- A dataset with m samples & n features will give to a n×n cov matrix.

- The n×n covariance matrix will have n e.vectors, so n PC's.

n features → n Principal components.

Take dataset → Standardize features

numpy commands E.vectors/Value → chose how much value/ variance →

plot scree plot → 70% data ⇒ 7/8 P.C's.

where does dim reduction comes from?

↳ can always ignore the components of lesser significance.

→ we do lose some information, but if the E.values are small

Step by Step computation

numpy commands
e. values
e. vectors

steps (1) standardization of the data
(2) Compute the covariance matrix
(3) Calculate the E. values & vectors of cov matrix
(4) Compute the Principal components by selecting the first D E. vectors.
(5) Reduces the dimensions of the dataset.

The features in which data is most spread out.

PCA → UNSUPERVISED. LEARNING

Final prediction result ⇒ we only concentrate on "FEATURES"

PCA works best when → good amount of spread
→ relation among features

PCA → construct
SVD →

SINGULAR VALUE DECOMPOSITION

→ SVD gives the decomposition for any arbitrary matrix,
$M = U \Lambda V^T$

$M_{m \times n} = U_{m \times r} \Lambda_{r \times r} V_{r \times n}^T$

Λ ↳ diagonal matrix = Root of +ve e.values of M $[X^T X$ or $X X^T]$

- U & V → orthogonal matrices, $U^T U = I$; $V^T V = I$
- U consists of orthonormal eigenvectors of M $[XX^T]$.
- V consists of orthonormal eigenvectors of M^T $[X^T X]$.

cov (M) = X

→ The SVD of the data matrix, $X = U \Lambda V^T$

→ After standardization, the cov matrix of the data matrix, $\Sigma = \frac{1}{m} X^T X$.

$$\Sigma = \frac{1}{m} X^T X = \frac{1}{m} (U \Lambda V^T)^T (U \Lambda V^T) = \frac{1}{m} (V \Lambda^T U^T)(U \Lambda V^T)$$

$$= \frac{1}{m} (V \Lambda^T \Lambda V^T) = \frac{1}{m} (V (\Lambda)^2 V^T)$$

→ $(\Lambda)^2$ is a diagonal matrix whose entries are $\Lambda_{ii} = \lambda_i^2$, the squares of the E.values of the SVD of X.

→ We can run SVD on X without ever

→ Both X and $X^T X$ share the same e.vectors in their SVD.

Date
Page

For ex: Given an image of a woman; where given no. of features = 200

at PC = 0 Just black & white lines

PC = 10 blurry

⋮

PC = 50 Almost similar

PC → capturing most of the variances

SUMMARY

→ PCA allows us to find the highest variance (lowest square distance) direction to project to.

→ E-values gives an indication of the number of dimensions to choose.

→ Can be computed in multiple ways (SVD is popular)

→ It is an unsupervised algo.

→ Ensures pre-processing for effectiveness

→ Is used in a variety of applications.