Foundations of Data Science

Konda Reddy Mopuri
Dept. of AI, IIT Hyderabad

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Data-driven Intelligence & Learning Lab

Scope

1. Curse of dimensionality
2. Covariance
3. Correlation
4. Dimensionality Reduction
5. Principal Component Analysis (PCA)

1. Curse of Dimensionality

Curse of Dimensionality: What?

- Challenges that can arise in spaces of higher dimensions
- Important factor influencing the design of PR/ML techniques

Example-1
Classifying the Pipeline measurements

Pipeline measurements

- Each data point consists of **12D** vector of measurements
- Material (each data point) can be present in one of the three geometric configurations (labels)

Pipeline measurements

- Plot of 100 points w.r.t. two of the dimensions (x_6 and x_7)

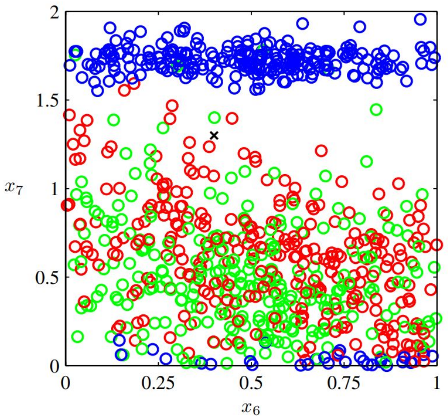


Figure: [PRML Book]

Pipeline measurements

- Points are labeled with their geometric configurations (i.e. labels)

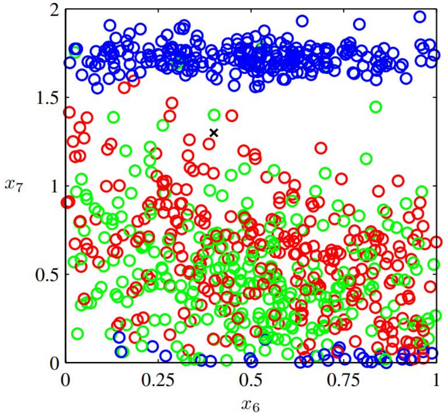


Figure: PRML Book

Pipeline measurements

- Goal: use this as training data and classify a test sample 'X'

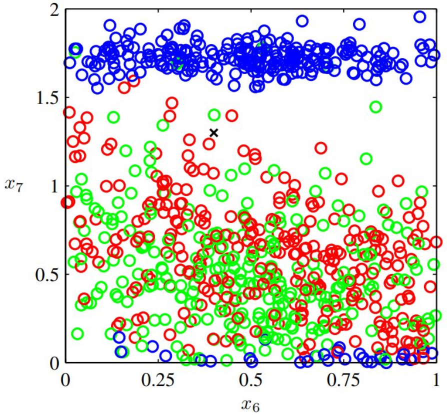


Figure: PRML Book

Pipeline measurements

- Approach: let's look at the neighbors
    - Intuition: identity of 'X' is determined more by its immediate neighbors from the training data than the distant ones

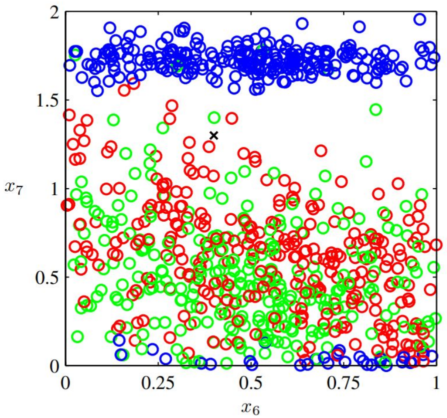
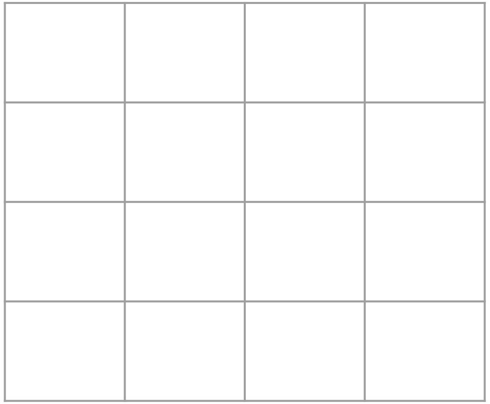


Figure: PRML Book

Pipeline measurements

- Let's turn this intuition into a learning algorithm
- How?
  - Divide the i/p space into regular cells



Pipeline measurements

- How?
  - Decide in which cell the test sample falls
  - In that cell observe which class has the most training data → majority voting

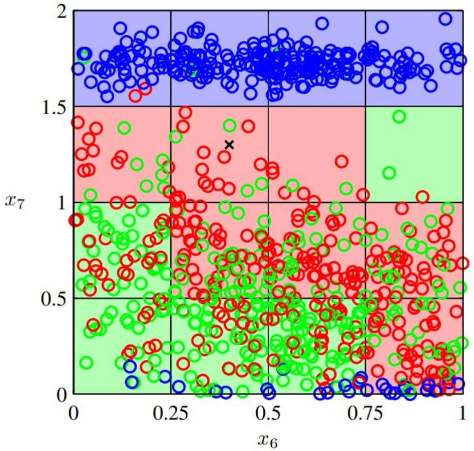


Figure: PRML Book

Issue with the approach

- Input spaces of higher dimensions
- As the dimension increases, the number of cells grows exponentially

Issue with the approach

- Input spaces of higher dimensions
- As the dimension increases, the number of cells grows exponentially

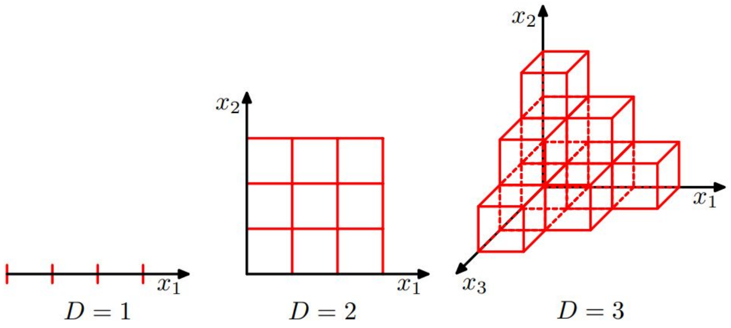


Figure: [PRML Book]

Issue with the approach

- Exponentially large no. of cells need exponentially large amount of training data
  - So that the they are not empty

Issue with the approach

- One has no hope of applying such a technique in a space of more than a few variables

Example-2
Polynomial Curve Fitting

Polynomial Curve Fitting

- Simple case of input having a single variable (*x*)
- Considering a polynomial of order M

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Polynomial Curve Fitting

- Extending to D variables with coefficients upto order 3

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D}\sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D}\sum_{j=1}^{D}\sum_{k=1}^{D} w_{ijk} x_i x_j x_k.$$

Polynomial Curve Fitting

- No. of coefficients grows proportional to ?

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D} \sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D} \sum_{j=1}^{D} \sum_{k=1}^{D} w_{ijk} x_i x_j x_k.$$

Polynomial Curve Fitting

- No. of coefficients grows proportional to D^3

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D} \sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D} \sum_{j=1}^{D} \sum_{k=1}^{D} w_{ijk} x_i x_j x_k.$$

Polynomial Curve Fitting

- In practice, we may need to use a higher-order polynomial of order M
  - No. of coefficients is proportional to D^M
  - Quickly goes out of hands and of limited practical utility

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D} \sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D} \sum_{j=1}^{D} \sum_{k=1}^{D} w_{ijk} x_i x_j x_k.$$

Difference of Geometric Intuitions b/w Lower and Higher dimensions

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Data-driven Intelligence & Learning Lab

Our intuitions fail

- We form intuitions from ‘easy to visualize’ spaces such as 3D
- They may fail badly in higher dimensional spaces

E.g. Fraction of the volume near the surface of the sphere

- Consider a sphere of unit radius in D dimensions
- What is the fraction of its volume that lies between radius $(1-\epsilon)$ and 1 ?

E.g. Fraction of the volume near the surface of the sphere

$$V_D(r) = K_D r^D$$

E.g. Fraction of the volume near the surface of the sphere

$$V_D(r) = K_D r^D$$

$$\frac{V_D(1) - V_D(1-\epsilon)}{V_D(1)} = 1 - (1-\epsilon)^D$$

E.g. Fraction of the volume near the surface of the sphere

In spaces of high dimensionality, most of the volume of a sphere is concentrated in a thin shell near the surface!

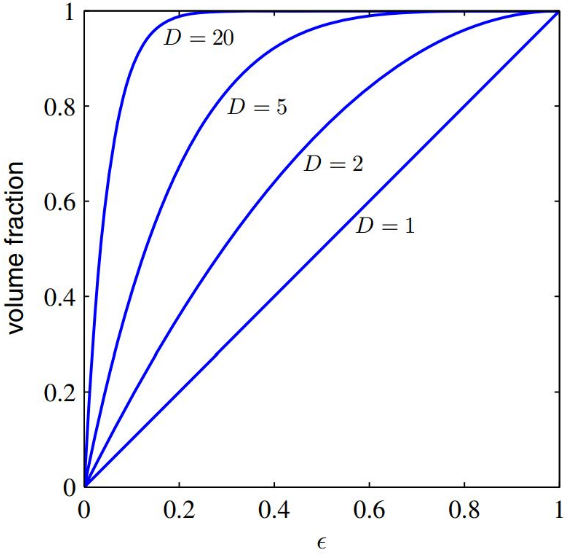


Figure: PRML Book

In short

- Not all intuitions developed in lower-dim spaces will generalize to spaces of many dimensions
- As the dimension grows
  - Volume of the space grows so fast that available data becomes sparse
  - Data looks dissimilar (prevents forming groups)
- Curse of dimensionality raises issues for learning

But, that doesn't prevent us building techniques in high-dimensions

How do we learn in higher dimensions?

1. Real data often confines to a lower dimensional subspace
    - Directions over which the target varies may be confined

How do we learn in higher dimensions?

2. Real data typically exhibits smoothness (locally)
    - Small changes in the input variables results in small changes in target
    - We can exploit techniques such as interpolation

E.g. Inferring Orientation of objects

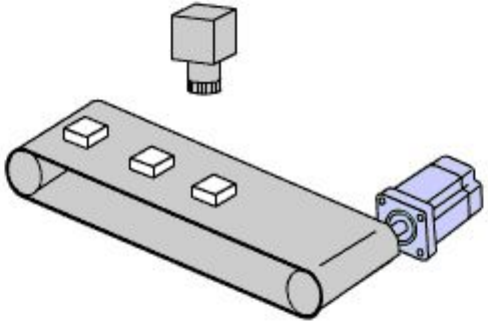


Figure: Oriental Motor

- Identical planar objects on a conveyor belt
- Goal: determine their orientation through images

E.g. Inferring Orientation of objects

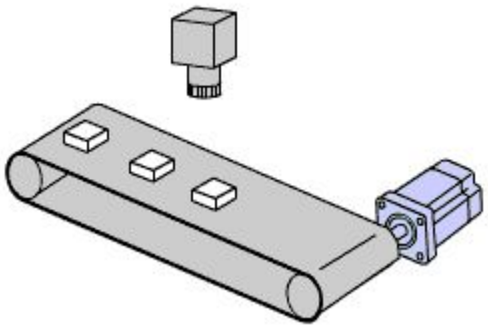


Figure: Oriental Motor

- Each image is a point in high-dim space
- They vary in ***position of the object***, ***orientation***, and ***pixel values***
- Hence, the set of images lie in a 3D manifold embedded in the high-dim space

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

DiL
Data-driven Intelligence & Learning Lab

E.g. Generative models

- GANs understanding the space of human faces (on a lower-dim manifold)
  - To generate real-looking faces

Figure credits

2. Covariance

Covariance

- Measure of the joint variability of two random variables
    - How much they co-vary (vary together)

Covariance

- Consider two random variables X, Y
  - With means E[Y] and E[Y]
- Their covariance is given by

$$COV(X,Y) = E[(X - E[X])(Y - E[Y])]$$

Covariance

- Take pairs of (X, Y)
- Take their differences from their means
- Take their product

$$COV(X,Y) = E[\boxed{(X - E[X])(Y - E[Y])}]$$

Covariance

- For a pair (x_1, y_1) this product is +ve
  - If the values of x and y have varied together in same direction from their means

$$COV(X,Y) = E[(X - E[X])(Y - E[Y])]$$

Covariance

- For a pair (x_1, y_1) this product is +ve
  - If the values of x and y have varied together in same direction from their means
- Larger the magnitude of the product, stronger the relationship!

$$COV(X,Y) = E[\boxed{(X - E[X])(Y - E[Y])}]$$

Covariance

- For a pair (x_2, y_2) this product is -ve
  - They have varied together in opposite directions (from their means)

$$COV(X,Y) = E[\boxed{(X - E[X])(Y - E[Y])}]$$

Covariance

- Covariance is the mean value of this product
  - Calculated with each pair of data points (x_i, y_i)

$$COV(X,Y) = E[(X - E[X])(Y - E[Y])]$$

Covariance

- What if the covariance is zero?
  - The +ve cases were offset by those in which it is -ve
  - There is no linear relationship between the two variables

$$COV(X,Y) = E[(X - E[X])(Y - E[Y])]$$

Covariance is +ve

- **Higher** than average values of **one** variable tend to pair with **higher** than average values of the **other** variable

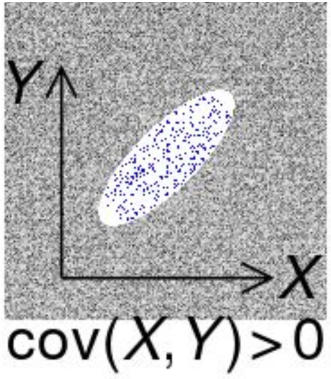


Figure: Wikipedia

Covariance is -ve

- **Higher** than average values of **one** variable tend to pair with **lower** than average values of the **other** variable

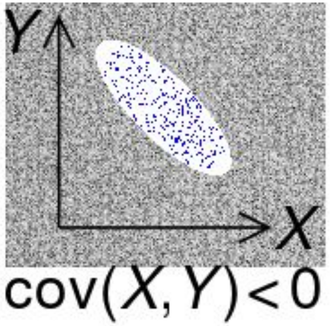


Figure: Wikipedia

Covariance and independence

- Variables for which the covariance is zero → Uncorrelated
- If two variables are independent, their covariance is zero
  - Converse need not be true

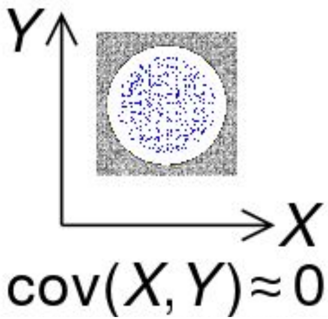


Figure: Wikipedia

Covariance

$$COV(X,Y) = E[(X - E[X])(Y - E[Y])]$$

$$COV(X,Y) = E[XY] - E[X]E[Y]$$

Covariance: Some properties

- $Cov(X,X) = Var(X)$
- $Var(X+Y) = Var(X) + Var(Y) + 2\,Cov(X,Y)$
- $Cov(aX,bY) = ab.Cov(X,Y)$

Covariance: Multivariate

- Two multivariate random variables $X \in R^m$ & $Y \in R^n$

$$COV(X,Y) = E[XY^T] - E[X]E[Y]^T = COV(Y,X)^T \in R^{m\times n}$$

Covariance: Multivariate

- When applied on a single random variable, tells its spread (variance)

$$
\begin{aligned}
\mathbb{V}_X[\boldsymbol{x}] &= \mathrm{Cov}_X[\boldsymbol{x}, \boldsymbol{x}] \\
&= \mathbb{E}_X[(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^\top] = \mathbb{E}_X[\boldsymbol{x}\boldsymbol{x}^\top] - \mathbb{E}_X[\boldsymbol{x}]\mathbb{E}_X[\boldsymbol{x}]^\top \\
&= \begin{bmatrix} \mathrm{Cov}[x_1, x_1] & \mathrm{Cov}[x_1, x_2] & \dots & \mathrm{Cov}[x_1, x_D] \\ \mathrm{Cov}[x_2, x_1] & \mathrm{Cov}[x_2, x_2] & \dots & \mathrm{Cov}[x_2, x_D] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}[x_D, x_1] & \dots & \dots & \mathrm{Cov}[x_D, x_D] \end{bmatrix} .
\end{aligned}
$$

Covariance: Limitations

- Depends on the units of the data
- Difficult to compare covariances among datasets (with different scales)
- A value that represents a strong linear relationship in one dataset may mean a weak relationship in another dataset

Covariance: Limitations

- Correlation coefficient addresses this
  - Normalize covariance to the product of individual standard deviations
  - Dimensionless quantity → facilitates comparison across datasets

3. Correlation

Correlation

- Is any statistical relationship between two random variables
    - Our interest is ‘linear’ relation
- Useful because it indicates a predictive relationship that can be exploited

Pearson’s Correlation Coefficient

- Familiar measure of Correlation b/w two random variables x, y

$$\mathrm{corr}[x, y] = \frac{\mathrm{Cov}[x, y]}{\sqrt{\mathbb{V}[x]\mathbb{V}[y]}} \in [-1, 1]$$

Pearson's Correlation Coefficient

- Corr(x,y) = Cov(x/σ(x) , y/σ(y))

$$\mathrm{corr}[x, y] = \frac{\mathrm{Cov}[x, y]}{\sqrt{\mathbb{V}[x]\mathbb{V}[y]}} \in [-1, 1]$$

Pearson's Correlation Coefficient

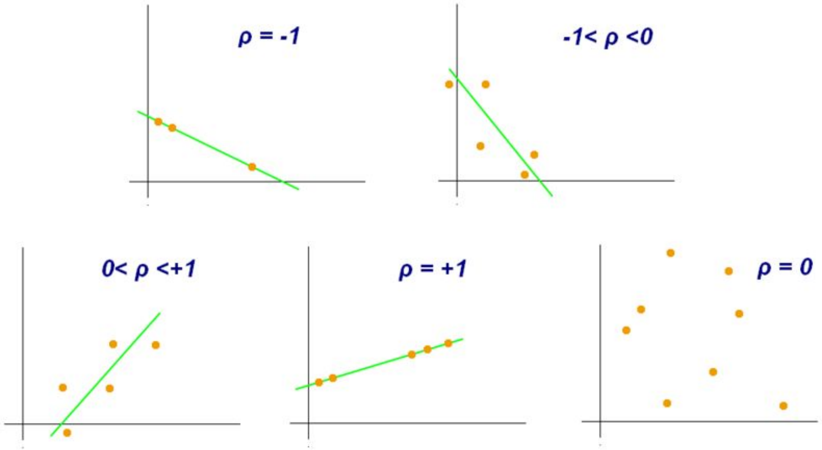


Figure: [Wikipedia]

Pearson's Correlation Coefficient

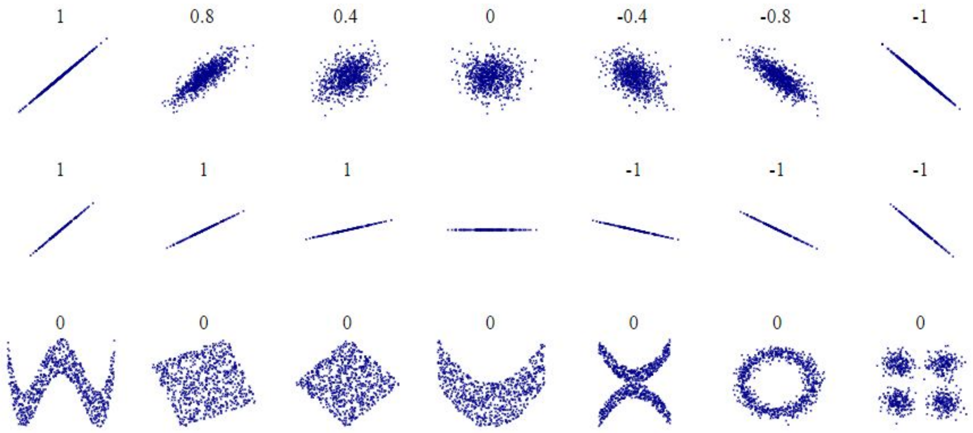


Figure: [Wikipedia]

4. Dimensionality Reduction

Issues with high-dim data

- Hard to analyze
  - Interpretation and visualization is challenging
- Storage and compute may become cumbersome

However, the high-dim data

- Overcomplete
    - Many dimensions are redundant
- Data possesses intrinsic lower-dimensional structure

Dimensionality Reduction

- Exploits the structure and correlation → compact representation of the data
    - With minimal information loss

Dimensionality Reduction

- PCA
- LDA
- t-SNE
- Autoencoders
- etc.

5. Principal Component Analysis (PCA)

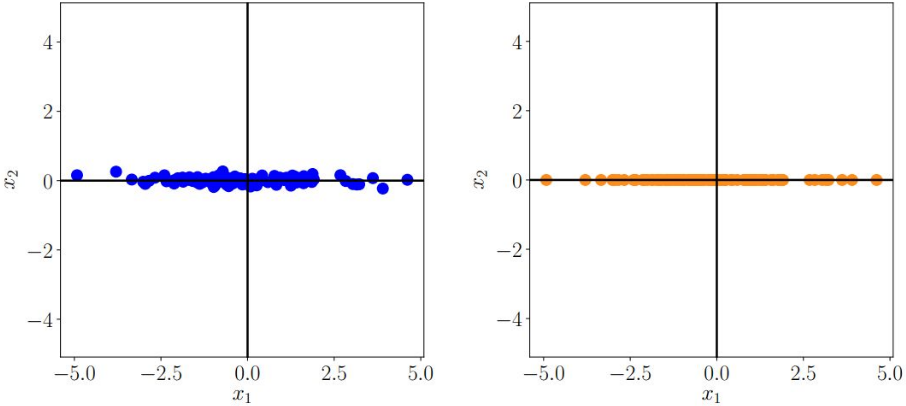
PCA

- Proposed by Pearson (1901) and Hotelling (1933)
- One of the most commonly used techniques for
  - data compression
  - Identification of patterns/structures
  - Visualization
- Also known as Karhunen-Loève (KL) transform

PCA from first principles

- Drawing from our understanding of
  - Basis
  - Projections
  - Eigen vectors
  - Constrained Optimization

Dimensionality Reduction



(a) Dataset with x_1 and x_2 coordinates.

(b) Compressed dataset where only the x_1 coordinate is relevant.

Figure: MML Book

PCA: problem setting

- **Goal**: find projections $\tilde{x}_n$ of data x_n that are as similar as possible, but with a lower intrinsic dimensionality

PCA: problem setting

- Consider iid data $\mathcal{X} = \{x_1, x_2, ... x_N\}$ in R^D with zero mean

PCA: problem setting

- Mean subtraction

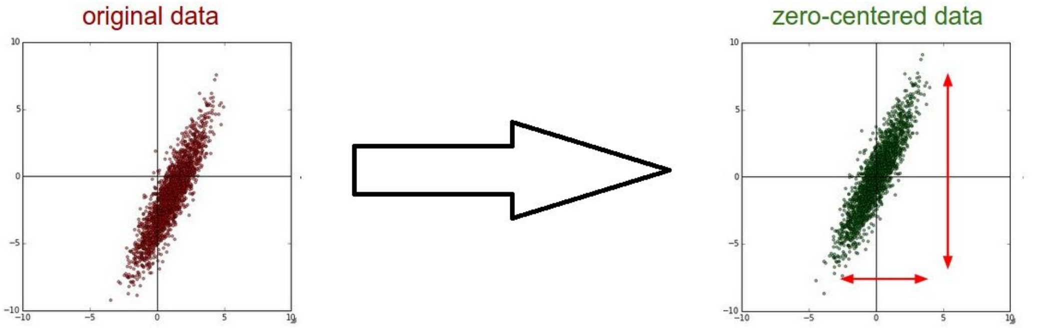


Figure: Ravindra Parmer

PCA: problem setting

- Covariance matrix

$$\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^\top$$

PCA: problem setting

- We assume that there exists a lower-dim compressed representation

$$z_n = B^\top x_n \in \mathbb{R}^M$$

of x_n, where we define the projection matrix

$$B := [b_1, \ldots, b_M] \in \mathbb{R}^{D \times M}$$

PCA: problem setting

- We assume that the columns of B are orthonormal

$$\boldsymbol{b}_i^\top \boldsymbol{b}_j = 0 \text{ if and only if } i \neq j \text{ and } \boldsymbol{b}_i^\top \boldsymbol{b}_i = 1$$

PCA: problem setting

- We seek an M-dimensional subspace in R^D with $dim(U) = M < D$
  - onto which we project the data

PCA: problem setting

- We denote the projected data as $\tilde{x}_n \in U$ and their coordinates w.r.t basis B as z_n
- Aim is to find the projections $\tilde{x}_n \in R^M$ that are similar to x_n and minimize the compression loss

PCA

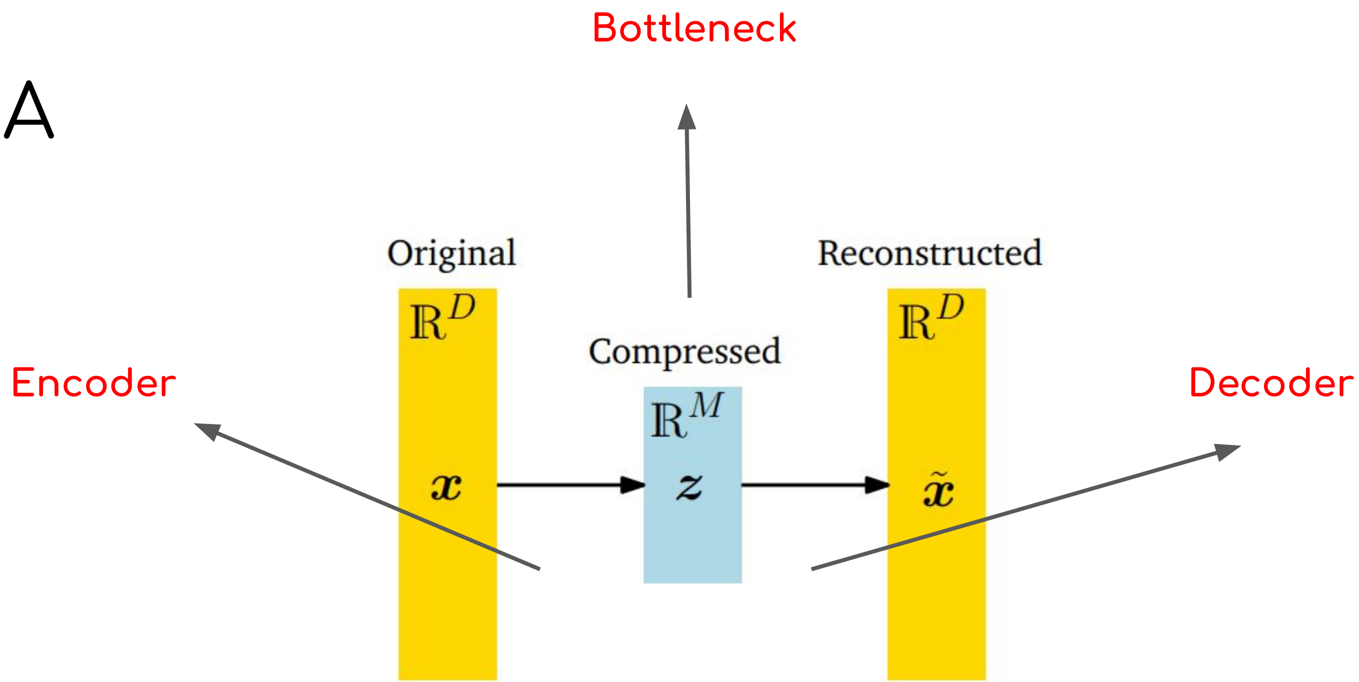
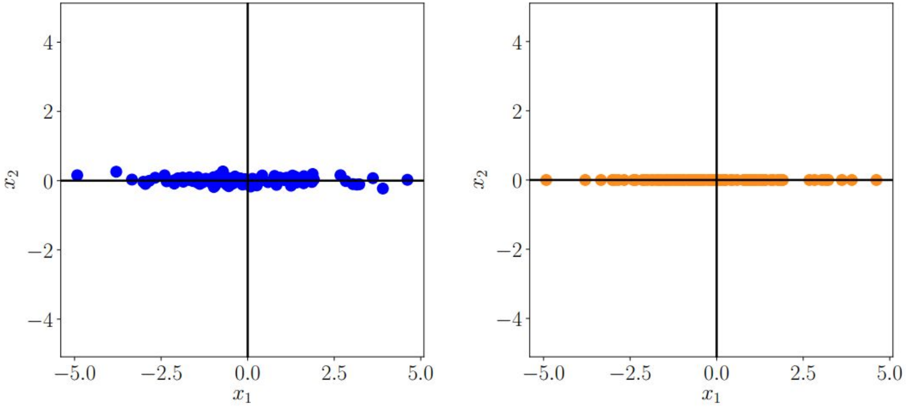


Figure: [MML Book]

PCA: maximum variance perspective



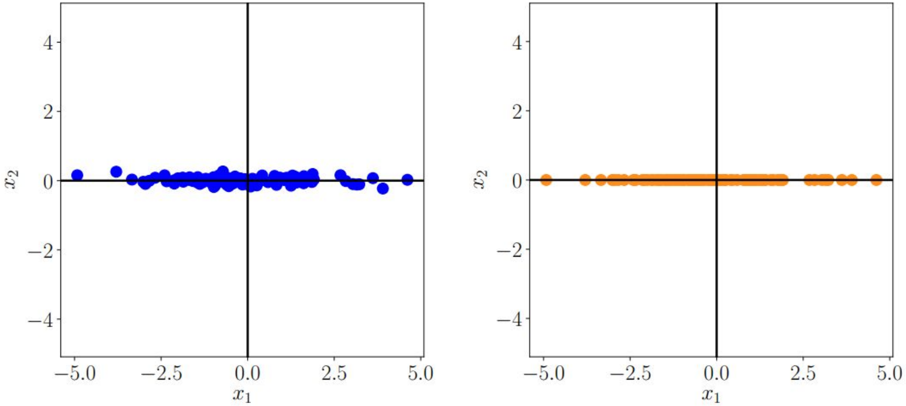
(a) Dataset with x_1 and x_2 coordinates.

(b) Compressed dataset where only the x_1 coordinate is relevant.

We chose to ignore x_2 because it did not add much information

Figure: [MML Book]

PCA: maximum variance perspective



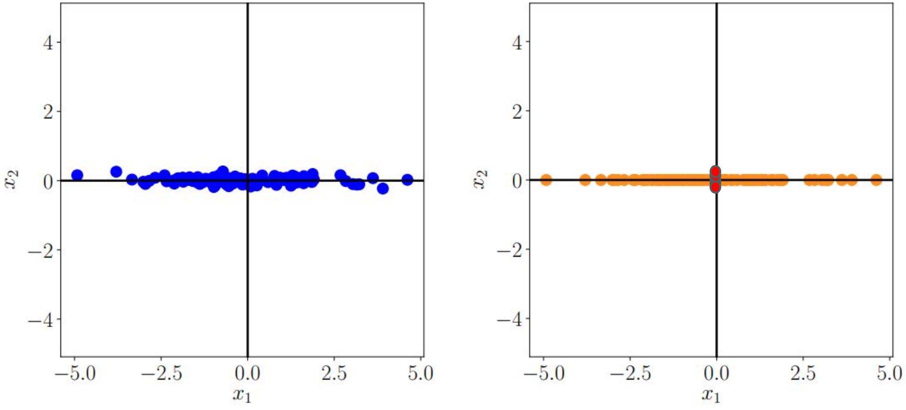
(a) Dataset with x_1 and x_2 coordinates.

(b) Compressed dataset where only the x_1 coordinate is relevant.

What if we ignore x_1?

Figure: MML Book

PCA: maximum variance perspective



(a) Dataset with x_1 and x_2 coordinates.

(b) Compressed dataset where only the x_1 coordinate is relevant.

What if we ignore x_1?

Much information would have been lost, and the compressed data would look very different

Figure: MML Book

Information content in data as its “spread“

- If we interpret the information content in the data as how much “space-filling” the dataset is
    - Then, it can be described by looking at its spread

Information content in data as its “spread“

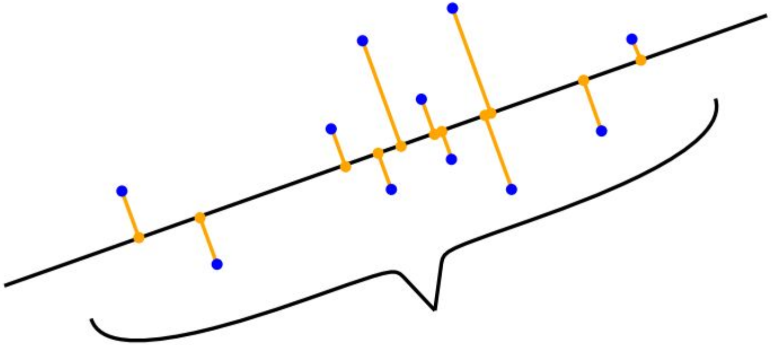


Figure: [MML Book]

Information content in data as its “spread“

- Variance is an indicator of spread in the data
- Hence, PCA maximizes the variance in the lower-dim representation of the data

Information content in data as its “spread“

- Retaining most information after data compression is equivalent to capturing the largest amount of variance in the lower-dim code

PCA: maximum variance perspective

- Let's maximize the variance in the lower-dim code
- We start by seeking a single vector $b_1 \in R^D$ that maximizes the variance of the projected data

$$V_1 := \mathbb{V}[z_1] = \frac{1}{N}\sum_{n=1}^{N} z_{1n}^2$$

$$z_{1n} = \boldsymbol{b}_1^\top \boldsymbol{x}_n$$

PCA: maximum variance perspective

$$V_1 = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{b}_1^\top \boldsymbol{x}_n)^2 = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{b}_1^\top \boldsymbol{x}_n \boldsymbol{x}_n^\top \boldsymbol{b}_1$$

$$= \boldsymbol{b}_1^\top \left(\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n \boldsymbol{x}_n^\top\right)\boldsymbol{b}_1 = \boldsymbol{b}_1^\top \boldsymbol{S}\boldsymbol{b}_1 \,,$$

S is the data covariance matrix

PCA: maximum variance perspective

$$V_1 = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{b}_1^\top \boldsymbol{x}_n)^2 = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{b}_1^\top \boldsymbol{x}_n \boldsymbol{x}_n^\top \boldsymbol{b}_1$$
$$= \boldsymbol{b}_1^\top \left(\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n \boldsymbol{x}_n^\top\right)\boldsymbol{b}_1 = \boldsymbol{b}_1^\top \boldsymbol{S} \boldsymbol{b}_1 \,,$$

- Increasing the magnitude of b_1 increases V_1
- Hence, we restrict the solutions to have unit norm → constrained optimization

PCA: maximum variance perspective

$$\max_{\boldsymbol{b}_1} \boldsymbol{b}_1^\top \boldsymbol{S} \boldsymbol{b}_1$$

$$\text{subject to } \|\boldsymbol{b}_1\|^2 = 1$$

PCA: maximum variance perspective

$$\mathfrak{L}(\boldsymbol{b}_1, \lambda) = \boldsymbol{b}_1^\top \boldsymbol{S} \boldsymbol{b}_1 + \lambda_1 (1 - \boldsymbol{b}_1^\top \boldsymbol{b}_1)$$

- Obtain the Lagrangian to solve this constrained optimization problem

PCA: maximum variance perspective

$$\mathfrak{L}(\boldsymbol{b}_1, \lambda) = \boldsymbol{b}_1^\top \boldsymbol{S} \boldsymbol{b}_1 + \lambda_1 (1 - \boldsymbol{b}_1^\top \boldsymbol{b}_1)$$

- Now, get the partial derivatives

$$\frac{\partial \mathfrak{L}}{\partial \boldsymbol{b}_1} = 2\boldsymbol{b}_1^\top \boldsymbol{S} - 2\lambda_1 \boldsymbol{b}_1^\top, \qquad \frac{\partial \mathfrak{L}}{\partial \lambda_1} = 1 - \boldsymbol{b}_1^\top \boldsymbol{b}_1$$

PCA: maximum variance perspective

$$\frac{\partial \mathfrak{L}}{\partial \boldsymbol{b}_1} = 2\boldsymbol{b}_1^\top \boldsymbol{S} - 2\lambda_1 \boldsymbol{b}_1^\top$$

$$\frac{\partial \mathfrak{L}}{\partial \lambda_1} = 1 - \boldsymbol{b}_1^\top \boldsymbol{b}_1$$

$$\Rightarrow$$

$$\boldsymbol{S}\boldsymbol{b}_1 = \lambda_1 \boldsymbol{b}_1 \,,$$
$$\boldsymbol{b}_1^\top \boldsymbol{b}_1 = 1 \,.$$

- Set them to 0

PCA: maximum variance perspective

$$\boldsymbol{S}\boldsymbol{b}_1 = \lambda_1 \boldsymbol{b}_1 \,,$$
$$\boldsymbol{b}_1^\top \boldsymbol{b}_1 = 1 \,.$$

- Comparing this to the Eigenvalue decomposition, clearly, b_1 is the eigenvector of the covariance matrix S

PCA: maximum variance perspective

$$V_1 = \boldsymbol{b}_1^\top \boldsymbol{S} \boldsymbol{b}_1 = \lambda_1 \boldsymbol{b}_1^\top \boldsymbol{b}_1 = \lambda_1$$

- Variance of the projected data on the 1D subspace = eigenvalue corresponding to b_1 (first eigenvector, also known as the first principal component)

PCA: maximum variance perspective

$$\tilde{\boldsymbol{x}}_n = \boldsymbol{b}_1 z_{1n} = \boldsymbol{b}_1 \boldsymbol{b}_1^\top \boldsymbol{x}_n \in \mathbb{R}^D$$

- Contribution of b_1 in the original data space is determined by z_{1n}
- Despite being a D-dim vector, it requires only one component to represent w.r.t. the basis vector b_1

Spectral Theorem

If $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is symmetric, there exists an orthonormal basis of the corresponding vector space V consisting of eigenvectors of $\boldsymbol{A}$, and each eigenvalue is real.

M-dim subspace with maximal variance

- Say, we have the first m-1 principal components
- How do we find the m^{th} Principal component?

M-dim subspace with maximal variance

- By subtracting the effect of the first m-1

$$\hat{\boldsymbol{X}} := \boldsymbol{X} - \sum_{i=1}^{m-1} \boldsymbol{b}_i \boldsymbol{b}_i^\top \boldsymbol{X} = \boldsymbol{X} - \boldsymbol{B}_{m-1} \boldsymbol{X}$$

Captures the remaining information

M-dim subspace with maximal variance

- To find the mth PC, we maximize the variance

$$V_m = \mathbb{V}[z_m] = \frac{1}{N}\sum_{n=1}^{N} z_{mn}^2 = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{b}_m^\top \hat{\boldsymbol{x}}_n)^2 = \boldsymbol{b}_m^\top \hat{\boldsymbol{S}} \boldsymbol{b}_m$$

M-dim subspace with maximal variance

- Turns out that

every eigenvector of $\boldsymbol{S}$ is an eigenvector of $\hat{\boldsymbol{S}}$

M-dim subspace with maximal variance

- Variance captured by a PC is equal to the eigenvalue

$$V_m = \mathbb{V}[z_m] = \frac{1}{N}\sum_{n=1}^{N} z_{mn}^2 = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{b}_m^\top \hat{\boldsymbol{x}_n})^2 = \boldsymbol{b}_m^\top \hat{\boldsymbol{S}} \boldsymbol{b}_m$$

$$V_m = \boldsymbol{b}_m^\top \boldsymbol{S} \boldsymbol{b}_m \quad = \quad \lambda_m \boldsymbol{b}_m^\top \boldsymbol{b}_m = \lambda_m$$

M-dim subspace with maximal variance

- Variance captured by PCA with M PCs (or, by projecting into an M-dim subspace)

$$V_M = \sum_{m=1}^{M} \lambda_m$$

M-dim subspace with maximal variance

- Relative variance captured by PCA with M PCs (or, by projecting into an M-dim subspace)

$$\frac{V_M}{V_D}$$

PCA steps

1. Standardize the data (mean subtraction and division by standard deviation)
2. Compute the covariance matrix
3. Compute the eigenvectors and eigenvalues of the covariance matrix → principal components
4. Decide how many principal components to keep
5. Transform the data using the principal components basis

Mathematics is the foundation for Data Science, Machine Learning, & Artificial Intelligence

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Data-driven Intelligence & Learning Lab

Foundation

- Linear Algebra & Matrix theory
- Vector Calculus
- Probability and Statistics
- Optimization
- etc.