School of Computing
UNIVERSITY OF GEORGIA

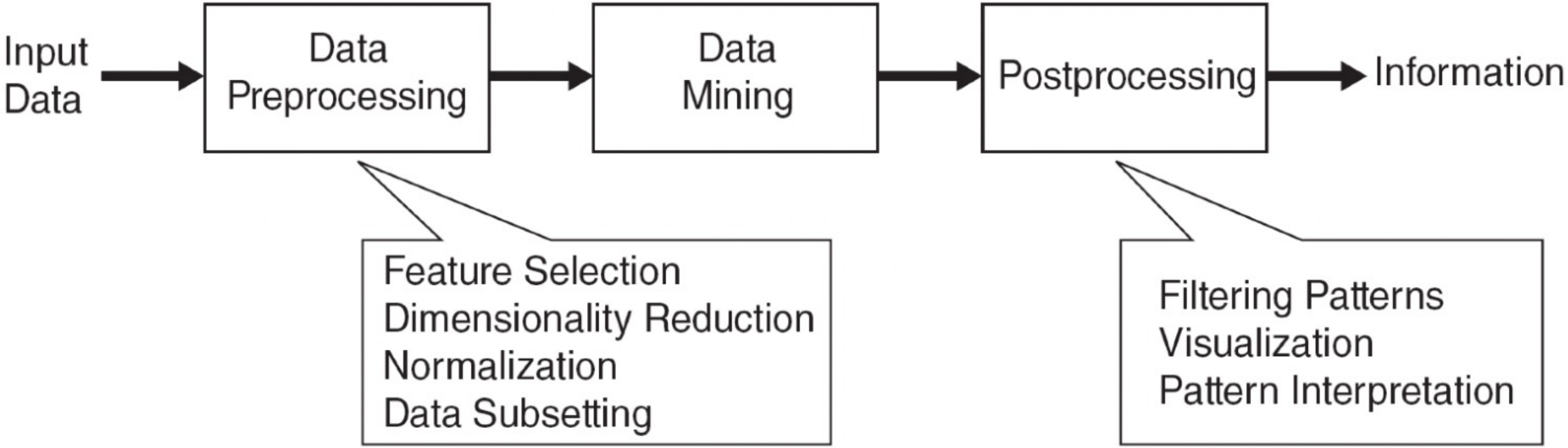
# CSCI 4380/6380 Data Mining

Fei Dou

Assistant Professor
School of Computing
University of Georgia

September 6, 2023

# Recap: Data Mining Process



# Data Preprocessing

# Data Preprocessing

- **Data cleaning**:
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- **Data integration**
  - Integration of multiple databases, data cubes, or files
- **Data reduction**
  - Dimensionality reduction
  - Numerosity reduction
  - Data compression
- **Data transformation and data discretization**
  - Normalization
  - Concept hierarchy generation

# Data Cleaning

# Data Cleaning

- **Data in the Real World Is Dirty**
  - **incomplete**: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data, e.g., Occupation=“ ” (missing data)
  - **noisy**: containing noise, errors, or outliers, e.g., Salary=“−10” (an error)
  - **inconsistent**: containing discrepancies in codes or names, e.g., Age=“42”, Birthday=“03/07/2010”
  - **Intentional** (e.g., disguised missing data) e.g., Jan. 1 as everyone’s birthday?

# Incomplete (Missing) Data

- Data is not always available

- Missing data may be due to
  - equipment malfunction
  - inconsistent with other recorded data and thus deleted data not entered due to misunderstanding
  - certain data may not be considered important at the time of entry
  - not register history or changes of the data

- Missing data may need to be inferred

# Missing Values - Methods

- Delete the object/instance
  - Pro: Easy to apply, does not tamper with the data.
  - Con: Can greatly reduce your sample size.

| Original Data set | | |
|---|---|---|
| Name | Age | Gender |
| Robin | 28 | Male |
| Heather | 29 | Female |
| Jamie | 22 | |
| Carl | 32 | Male |
| | 35 | Male |
| Sarah | 26 | Female |

| Data set after Listwise deletion | | |
|---|---|---|
| Name | Age | Gender |
| Robin | 28 | Male |
| Heather | 29 | Female |
| Carl | 32 | Male |
| Sarah | 26 | Female |

# Missing Values - Methods

- Imputation
  - Pro: No loss in sample size, preservation of data distribution, reduced bias.
  - Con: Complexity, potential for error, cannot be applied on All types of data.

- Mean: sum of all values in the column divided by the number of values present in the column

- Median: sort all values in the column, odd: (n+1)/2 th value; even: average of n/2-th and (n+2)/2-th values.

- Mode: the value that occurs the most often in the range of values

# Missing Values - Methods

- Imputation

- Model based approach: kNN, Autoregressive/Moving Average based prediction, linear/Neural Networks based interpolation.

- Last Observation Carried Forward (LOCF)

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Original Data Set | | | | | Data After LOCF | | | |
| 2 | Name | Visit | Month | Weight | | Name | Visit | Month | Weight |
| 3 | Robin | 1 | January | 65 | | Robin | 1 | January | 65 |
| 4 | Robin | 2 | February | 68 | | Robin | 2 | February | 68 |
| 5 | Robin | 3 | March | | | Robin | 3 | March | 68 |
| 6 | Robin | 4 | April | | | Robin | 4 | April | 68 |
| 7 | Robin | 5 | May | 72 | | Robin | 5 | May | 72 |
| 8 | Robin | 6 | June | 71 | | Robin | 6 | June | 71 |
| 9 | Heather | 1 | January | 52 | | Heather | 1 | January | 52 |
| 10 | Heather | 2 | February | 51 | | Heather | 2 | February | 51 |
| 11 | Heather | 3 | March | 56 | | Heather | 3 | March | 56 |
| 12 | Heather | 4 | April | 52 | | Heather | 4 | April | 52 |
| 13 | Heather | 5 | May | | | Heather | 5 | May | 52 |
| 14 | Heather | 6 | June | | | Heather | 6 | June | 52 |
| 15 | Jamie | 1 | January | | | Jamie | 1 | January | - |
| 16 | Jamie | 2 | February | 78 | | Jamie | 2 | February | 78 |
| 17 | Jamie | 3 | March | 81 | | Jamie | 3 | March | 81 |
| 18 | Jamie | 4 | April | | | Jamie | 4 | April | 81 |
| 19 | Jamie | 5 | May | | | Jamie | 5 | May | 81 |
| 20 | Jamie | 6 | June | 75 | | Jamie | 6 | June | 75 |

# Noise Data

- **Noise**: random error or variance in a measured variable

- **Incorrect attribute values** may be due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention

- Other data problems which require data cleaning
  - duplicate records
  - incomplete data
  - inconsistent data

# How to Handle Noisy Data?

- **Binning**
  - first sort data and partition into (equal-frequency) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- **Regression**
  - smooth by fitting the data into regression functions
- **Clustering**
  - detect and remove outliers
- **Combined computer and human inspection**
  - detect suspicious values and check by human (e.g., deal with possible outliers)

# Re-cap: Data Preprocessing

- **Data cleaning**:
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- **Data integration**
  - Integration of multiple databases, data cubes, or files
- **Data reduction**
  - Dimensionality reduction
  - Numerosity reduction
  - Data compression
- **Data transformation and data discretization**
  - Normalization
  - Concept hierarchy generation

# Data Integration

# Data Integration

- **Data integration:** Combines data from multiple sources into a coherent store
- Schema integration: e.g., A.cust-id and B.cust-number
- Entity identification problem: Identify real world entities from multiple data sources, e.g., Bill Clinton = William Clinton
- Detecting and resolving data value conflicts
  - For the same real-world entity, attribute values from different sources are different
  - Possible reasons: different representations, different scales, e.g., metric vs. British units.

# Handling Redundancy in Data Integration

- Redundant data occur often when integration of multiple databases
  - **Object identification**
  - **Derivable data**

- Redundant attributes may be able to be detected by correlation analysis and covariance analysis

- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

# Expected Value

- The ***expected value*** or ***expectation*** of a function $f(X)$ with respect to a probability distribution $P(X)$ is the average (mean) when $X$ is drawn from $P(X)$
- For a discrete random variable $X$, it is calculated as

$$\mathbb{E}_{X\sim P}[f(X)] = \sum_{X} P(X) f(X)$$

- For a continuous random variable $X$, it is calculated as

$$\mathbb{E}_{X\sim P}[f(X)] = \int P(X) f(X)\, dX$$

  - When the identity of the distribution is clear from the context, we can write $\mathbb{E}_X[f(X)]$
  - If it is clear which random variable is used, we can write just $\mathbb{E}[f(X)]$
- Mean is the most common measure of central tendency of a distribution
  - For a random variable: $f(X_i) = X_i \quad \Rightarrow \quad \mu = \mathbb{E}[X_i] = \sum_i P(X_i) \cdot X_i$
  - This is similar to the mean of a sample of observations: $\mu = \frac{1}{N} \sum_i X_i$
  - Other measures of central tendency: median, mode

# Variance

- ***Variance*** gives the measure of how much the values of the function $f(X)$ deviate from the expected value as we sample values of X from $P(X)$

$$\mathrm{Var}\big(f(X)\big) = \mathbb{E}[(f(X) - \mathbb{E}[f(X)])^2]$$

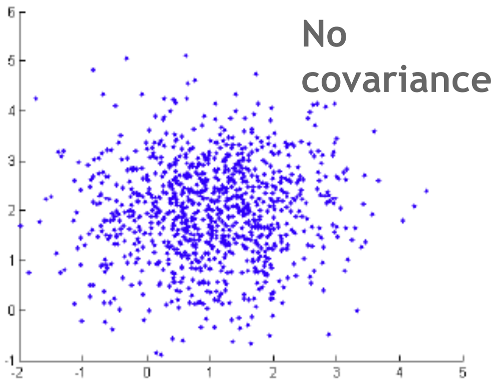
- When the variance is low, the values of $f(X)$ cluster near the expected value
- Variance is commonly denoted with $\sigma^2$
  - The above equation is similar to a function $X_i - \mu$
  - We have $\sigma^2 = \sum_i P(X_i) \cdot (X_i - \mu)^2$
  - This is similar to the formula for calculating the variance of a sample of observations: $\sigma^2 = \frac{1}{N-1} \sum_i (X_i - \mu)^2$
- The square root of the variance is the ***standard deviation***
  - Denoted $\sigma = \sqrt{\mathrm{Var}(X)}$

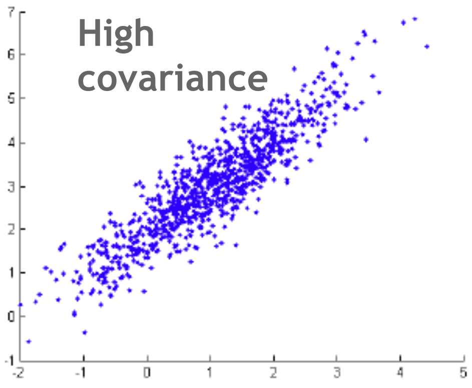
# Covariance

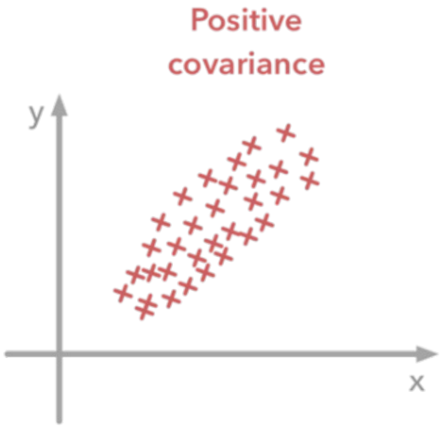
- ***Covariance*** gives the measure of how much two random variables are linearly related to each other

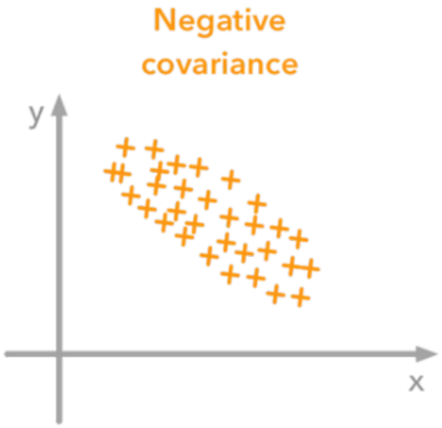
$$\mathrm{Cov}\big(f(X), g(Y)\big) = \mathbb{E}[(f(X) - \mathbb{E}[f(X)])(g(Y) - \mathbb{E}[g(Y)])]$$

- If $f(X_i) = X_i$ and $g(Y_i) = Y_i$
  - Then, the covariance is: $\mathrm{Cov}(X, Y) = \sum_i P(X_i, Y_i) \cdot (X_i - \mu_X) \cdot (Y_i - \mu_Y)$
  - Compare to covariance of actual samples: $\mathrm{Cov}(X, Y) = \frac{1}{N-1} \sum_i (Y_i - \mu_X)(Y_i - \mu_Y)$
- The covariance measures the tendency for $X$ and $Y$ to deviate from their means in same (or opposite) directions at same time



# Covariance Matrix

- ***Covariance matrix*** of a multivariate random variable $\mathbf{X}$ with states $\mathbf{x} \in \mathbb{R}^{\boldsymbol{n}}$ is an $n \times n$ matrix, such that

$$\mathrm{Cov}(\mathbf{X})_{i,j} = \mathrm{Cov}(\mathbf{x}_i, \mathbf{x}_j)$$

- I.e.,

$$\mathrm{Cov}(\mathbf{X}) = \begin{bmatrix} \mathrm{Cov}(\mathbf{x}_1, \mathbf{x_1}) & \mathrm{Cov}(\mathbf{x}_1, \mathbf{x_2}) & \cdots & \mathrm{Cov}(\mathbf{x}_1, \mathbf{x}_{\boldsymbol{n}}) \\ \mathrm{Cov}(\mathbf{x}_2, \mathbf{x_1}) & & & \mathrm{Cov}(\mathbf{x}_2, \mathbf{x}_{\boldsymbol{n}}) \\ \vdots & & \ddots & \vdots \\ \mathrm{Cov}(\mathbf{x}_n, \mathbf{x_1}) & \mathrm{Cov}(\mathbf{x}_n, \mathbf{x_2}) & \cdots & \mathrm{Cov}(\mathbf{x}_{\boldsymbol{n}}, \mathbf{x}_{\boldsymbol{n}}) \end{bmatrix}$$

- The diagonal elements of the covariance matrix are the variances of the elements of the vector

$$\mathrm{Cov}(\mathbf{x}_i, \mathbf{x}_i) = \mathrm{Var}(\mathbf{x}_i)$$

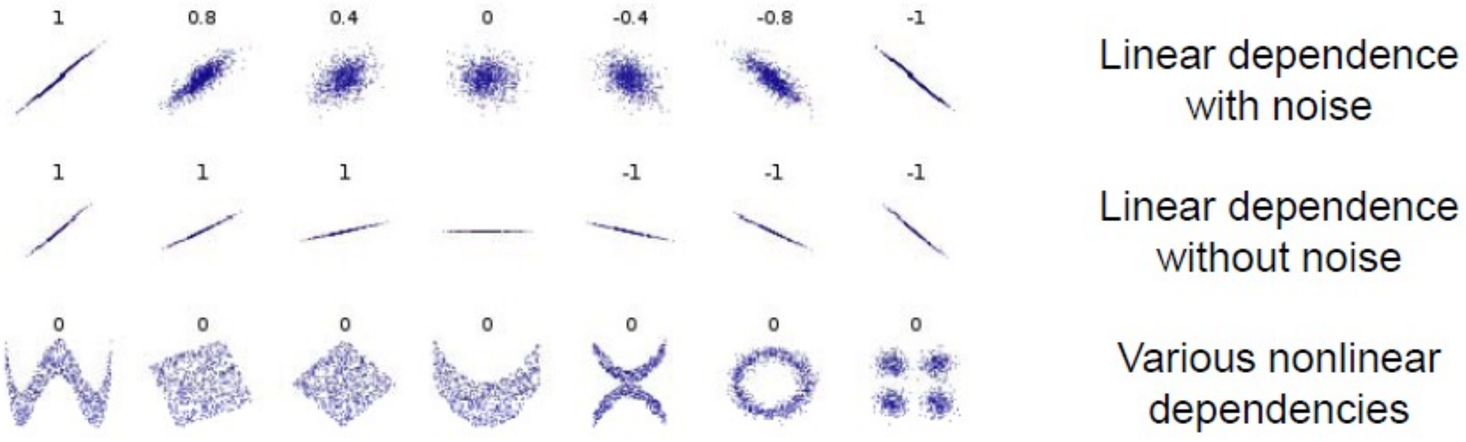
- Also note that the covariance matrix is symmetric, since $\mathrm{Cov}(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{Cov}(\mathbf{x}_j, \mathbf{x}_i)$

# Correlation

- ***Correlation coefficient*** is the covariance normalized by the standard deviations of the two variables

$$\mathrm{corr}(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \cdot \sigma_Y}$$

  - It is also called Pearson's correlation coefficient and it is denoted $\rho(X,Y)$
  - The values are in the interval $[-1,1]$
  - It only reflects linear dependence between variables, and it does not measure non-linear dependencies between the variables



# Data Reduction

# Data Reduction

- **Data reduction**: Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results

- **Why data reduction? —** A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to run on the complete data set.

# Data Reduction Strategies

- **Dimensionality reduction**, e.g., remove unimportant attributes
  - Principal Components Analysis (PCA)
  - Wavelet transforms
  - Feature subset selection, feature creation

- **Numerosity reduction** (some simply call it: Data Reduction)
  - Regression and Log-Linear Models
  - Histograms, clustering, sampling
  - Data cube aggregation

- **Data compression**

# Dimensionality Reduction

- Curse of dimensionality
  - When dimensionality increases, data becomes increasingly sparse

- Dimensionality reduction
  - Avoid the curse of dimensionality
  - Help eliminate irrelevant features and reduce noise
  - Reduce time and space required in data mining
  - Allow easier visualization

- Dimensionality reduction techniques
  - Principal Component Analysis
  - Fourier transforms and wavelet transform
  - Supervised and nonlinear techniques (e.g., feature selection)

# PCA

- **Motivation** With $n$ different data objects with $d$ attributes, we aim to learn a low dimensional representation

$$\mathbf{X} \in \mathbb{R}^{n \times d} \longrightarrow f(\mathbf{X}) \in \mathbb{R}^{n \times k}$$

  - Reduce curse of dimensionality problems
  - Reduce redundancies in the data
  - Increase storage and computational efficiency
  - Visualize data in 2D or 3D

# Re-cap: Eigen Decomposition

- ***Eigen decomposition*** is decomposing a matrix into a set of eigenvalues and eigenvectors
- ***Eigenvalues*** of a square matrix $\mathbf{A}$ are scalars $\lambda$ and ***eigenvectors*** are non-zero vectors $\mathbf{v}$ that satisfy

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

- Eigenvalues are found by solving the following equation

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0$$

- If a matrix $\mathbf{A}$ has $n$ linearly independent eigenvectors $\{\mathbf{v}^1, \dots, \mathbf{v}^n\}$ with corresponding eigenvalues $\{\lambda_1, \dots, \lambda_n\}$, the eigen decomposition of $\mathbf{A}$ is given by

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$$

  - Columns of the matrix $\mathbf{V}$ are the eigenvectors, i.e., $\mathbf{V} = [\mathbf{v}^1, \dots, \mathbf{v}^n]$
  - $\mathbf{\Lambda}$ is a diagonal matrix of the eigenvalues, i.e., $\mathbf{\Lambda} = [\lambda_1, \dots, \lambda_n]$

- To find the inverse of the matrix A, we can use $\mathbf{A}^{-1} = \mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{V}^{-1}$
  - This involves simply finding the inverse $\mathbf{\Lambda}^{-1}$ of a diagonal matrix

# Singular Value Decomposition

- ***Singular value decomposition*** (SVD) provides another way to factorize a matrix, into singular vectors and singular values
  - SVD is more generally applicable than eigen decomposition
  - Every real matrix has an SVD, but the same is not true of the eigen decomposition
    - E.g., if a matrix is not square, the eigen decomposition is not defined, and we must use SVD
- SVD of an $m \times n$ matrix $\mathbf{A}$ is given by

$$\mathbf{A} = \mathbf{U\Sigma V}^{\boldsymbol{T}}$$

  - $\mathbf{U}$ is an $m \times m$ matrix,$\boldsymbol{\Sigma}$ is an $m \times n$ matrix, and $\mathbf{V}$ is an $n \times n$ matrix
  - The elements along the diagonal of $\boldsymbol{\Sigma}$ are known as the singular values of $A$
  - The columns of $\mathbf{U}$ are known as the left-singular vectors
  - The columns of $\mathbf{V}$ are known as the right-singular vectors
- For a non-square matrix $\mathbf{A}$, the squares of the singular values $\sigma_i$ are the eigenvalues $\lambda_i$ of $\mathbf{A}^{\boldsymbol{T}}\mathbf{A}$, i.e., $\sigma_i^2 = \lambda_i$ for $i = 1,2, \ldots, n$
- Applications of SVD include computing the pseudo-inverse of non-square matrices, matrix approximation, determining the matrix rank

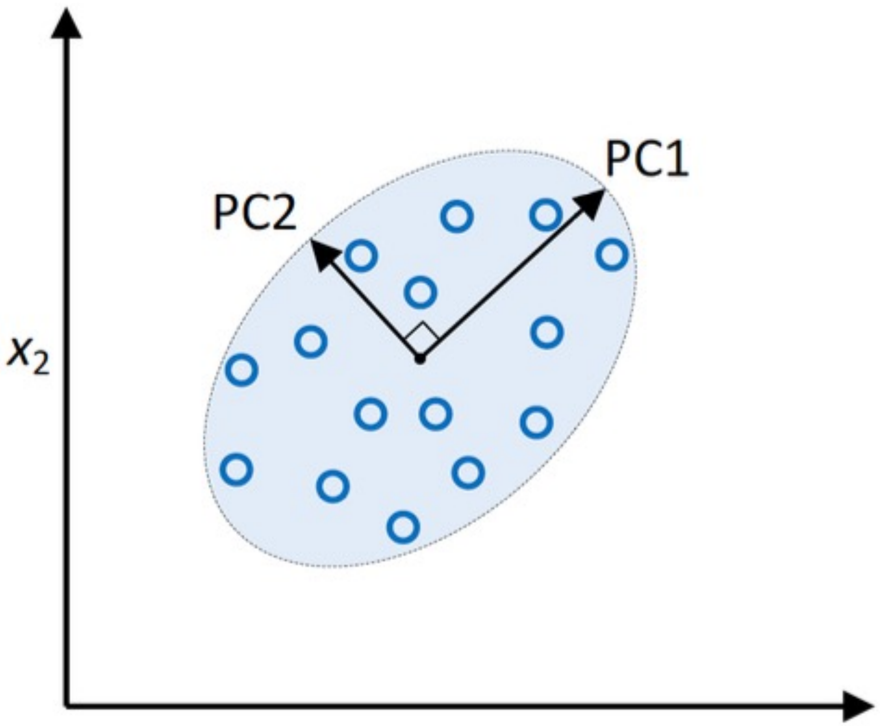
# PCA

**Algorithm**

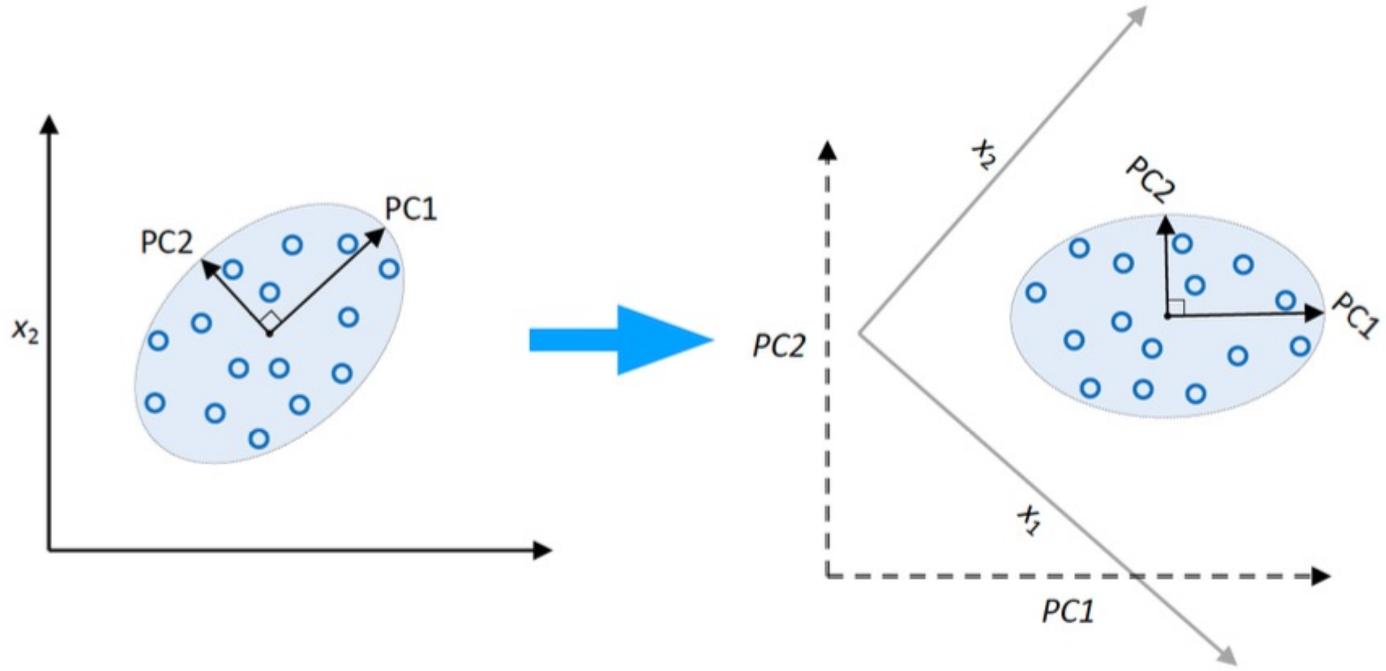
- Normalize the data to be zero mean. (m data objects with n features)
- Calculate the sample covariance matrix
- Find the n eigenvector -eigenvalue pairs of the sample covariance matrix
  - PCA basis vectors = the eigenvector
  - Larger eigenvalue $\Longrightarrow$ more important eigenvectors
- Choose the top k eigenvectors corresponding to the highest eigenvalues
- Project the data to the lower dimensional space.

# PCA

- **Intuition**
  - Step 1: Find directions of maximum variance



- **Intuition**
  - Step 2: Transform features onto directions of maximum variance



# PCA

- **Intuition**
  - Step 3: Usually consider a subset of vectors of most variance (DR)