Data Mining - Appunti

Francesco Lorenzoni

Febrero 2025

Contents

Part I

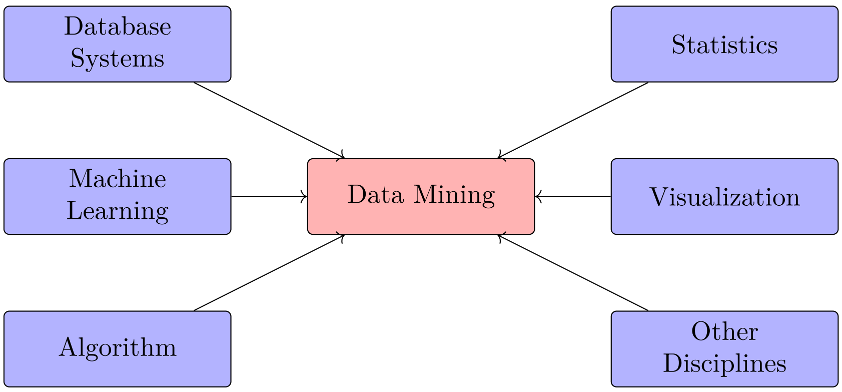
Introduction to Data Mining

Chapter 1

Introduction

Definition 1.1 (Data Mining) *is the use of efficient techniques for the analysis of very large collections of data and the extraction of useful and possibly unexpected patterns in data (hidden knowledge).* *The goal is to extract (human-readable) knowledge and insight from raw data.*

- Knowledge implies we are often not just trying to solve a task
- Insight implies that we should infer non-obvious knowledge
- Human-readable implies that knowledge should be (when possible) understood by humans: focus on interpretability!
- Raw data implies we'll need to clean it



1.1 Definitions

1.1.1 Knowledge Discovery Loop

Large collections tend to be heterogeneous in source, domain, language and refinement. The first step is to store the data, which however does not assess its heterogeneity. Data cleaning and integration tackle this problem, so that we get integrated sources, homogenous language, and data cleared of noise and outliers.

To look for insight on the data we have to answer questions on the data as a stakeholder. We may see patterns and ask ourselves their nature. Pattern extraction and validation lead to possible insight. Insight may lead to noticing that some data missing may be useful, and we may want to collect it, going back at previous steps.

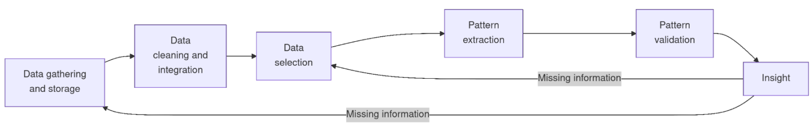


Figure 1.1: Knowledge Discovery Loop
This essentially summarizes the KDD process.

1.1.2 KDD Process

The KDD process consists of the following steps:

1. **Data Cleaning:** Remove noise and inconsistent data.

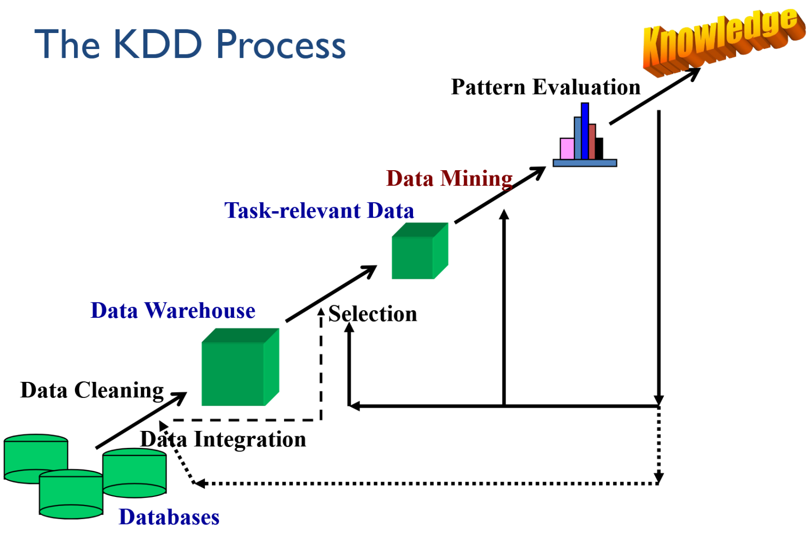
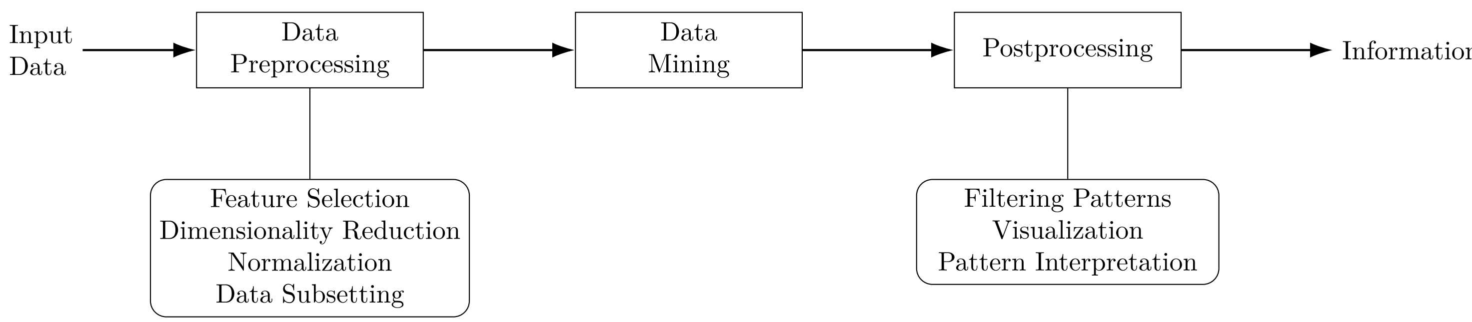


Figure 1.2: KDD Process

2. **Data Integration:** Combine multiple data sources.
   Involves the process of data understanding, data cleaning, merging data coming from multiple sources and transfoming them to load them into a **Data Warehouse**.
   Data Warehouse is a database targeted to answer specific business questions
3. **Data Selection:** Select relevant data for analysis.
4. **Data Transformation:** Transform data into suitable formats for mining (summary, aggregation, etc.).
5. **Data Mining:** Apply algorithms to extract patterns.
   - ◇ *Prediction Methods*
     Use some variables to predict unknown or future values of other variables.
   - ◇ *Description Methods*
     Find human-interpretable patterns that describe the data.
6. **Pattern Evaluation:** Identify truly interesting patterns.
7. **Knowledge Presentation:** Present the mined knowledge in an understandable way.

1.1.3 Data Mining Process



Definition 1.2 (Primary Data) *Original data that has been collected for a specific purpose.*
Primary data is not altered by humans

Definition 1.3 (Secondary Data) *Data that has been already collected and made available for other purposes.*
Secondary data may be obtained from many sources

Definition 1.4 (Association rule discovery) *Given a set of records each of which contain some number of items from a given collection.*
Produce dependency rules which will predict occurrence of an item based on occurrences of other items.

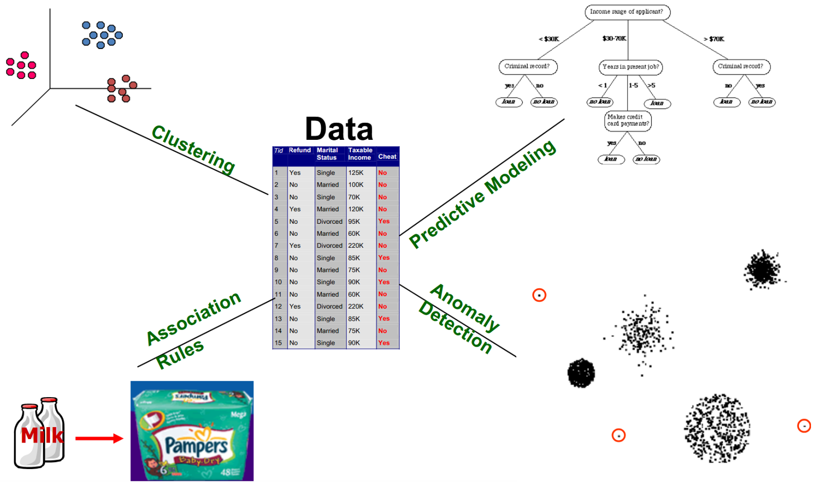


Figure 1.3: Data Mining methods

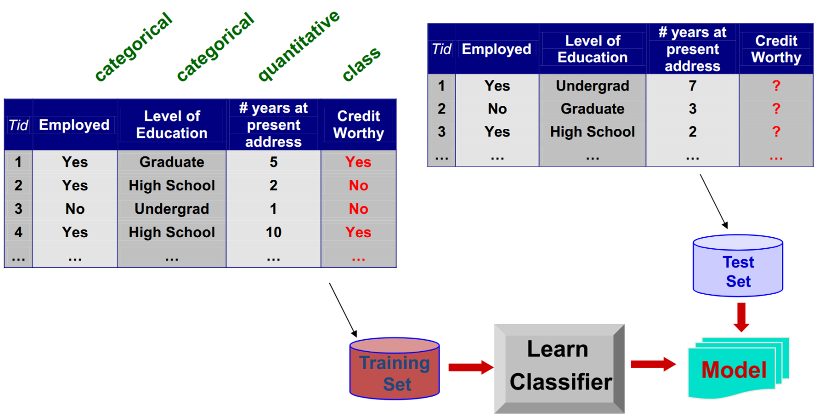


Figure 1.4: Classification Process

Association Use Cases

- **Market-basket analysis**
  Rules are used for sales promotion, shelf management, and inventory management
- **Telecommunication alarm diagnosis**
  Rules are used to find combination of alarms that occur together frequently in the same time period
- **Medical Informatics**
  Rules are used to find combination of patient symptoms and test results associated with certain diseases

1.2 Data Understanding

Definition 1.5 (Data) *Data is a collection of data objects and their attributes.*
An attribute is a property or characteristic of an object. A collection of attributes describe an object (record).

If data objects have the same fixed set of numeric attributes, then the data objects can be thought of as points in a multi-dimensional space, where each dimension represents a distinct attribute.
Such data set can be represented by an $m \times n$ matrix, where there are m rows, one for each object, and n columns, one for each attribute.

Data Types:
- Document data
- Transaction data
- Graph data
- Ordered data
  - Spatial data
  - Temporal data

The type of the attribute depends on the following properties:
- Distinctness: $=\neq$
- Order: $<>$
- Differences are meaningful: $+-$
- Ratios are meaningful: $*/$

Attribute types:
- Nominal/Categorical: attribute values in a finite domain (*distinctness*)
- Binary: special case of nominal with two values
- Ordinal: attribute values have a total ordering (*distinctness* and *order*)
- Numeric: quantity (integer or real-valued) (*distinctness*, *order*, *differences*)
- Ratio-Scaled: we can speak of values as being an order of magnitude larger than the unit of measurement (*all 4 properties*)
  length, counts, elapsed time (A baseball game lasting 3 hours is 50% longer than a game lasting 2 hours)
- Discrete/Continuous: attribute values are discrete (finite or countably infinite) or continuous (real-valued).

Attribute Type	Description	Examples	Operations
Nominal	Nominal attribute values only distinguish. $(=, \neq)$	zip codes, employee ID numbers, eye color, sex: {male, female}	mode, entropy, contingency correlation, χ^2 test
Ordinal	Ordinal attribute values also order objects. $(<, >)$	hardness of minerals, {good, better, best}, grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
Interval	For interval attributes, differences between values are meaningful. $(+, -)$	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation, Pearson's correlation, t and F tests
Ratio	For ratio variables, both differences and ratios are meaningful. $(*, /)$	temperature in Kelvin, monetary quantities, counts, age, mass, length, current	geometric mean, harmonic mean, percent variation

Table 1.1: Attribute types, examples and operations

1.2.1 Data Quality

Examples of data quality problems:

- Wrong data
- Duplicate data
- Noise and outliers
- Missing values

In order to know our data and discovery quality issues we need use descriptive statistics for getting a global picture and summarize properties of data and compare such statistics with the expected behaviour. All around we can exploit visualization techniques that can help in detecting general or unusual patterns and trends, as well as outliers.

1.2.1.1 Histograms

A histogram shows the frequency distribution for a numerical attribute. The range of the numerical attribute is discretized into a fixed number of intervals (**bins**).

The number of bins according to Sturges' rule is:

$$k = \lceil \log_2 n + 1 \rceil$$

where n is the number of records in the data set. Sturges' rule is suitable for data from normal distributions and from data sets of moderate size.

1.2.1.2 Statistics notions

Notoriouses Mean/Median/Mode...The degree in which data tend to spread is called the *dispersion*, or **variance** of the data.

The most common measures for data dispersion are **range** (The distance between the largest and the smallest values), **standard deviation**, the **five-number summary** (based on *quartiles*), and the **inter-quartile range**.

$$\text{variance}(x) = \sigma^2 = s_x^2 = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \bar{x})^2$$

Standard deviation σ is the square root of variance σ^2.

Because of outliers, other measures are often used:

- absolute average deviation (AAD)
$$\text{AAD}(x) = \frac{1}{m}\sum_{i=1}^{m}|x_i - \bar{x}|$$
- median average deviation (MAD)
$$\text{MAD}(x) = \text{median}\left(\{|x_1 - \bar{x}|, \dots, |x_m - \bar{x}|\}\right)$$

1.2.1.3 Box-Plot

1.3 Data Understanding - Lab

Data comes from diverse sources, and generally is not tailor-made for some downstream task. We need to start from basics:

- What features are available?
- What are they measuring, exactly?
- What properties do they have?
- What are their relations?
- Are there outliers?
- ...

Data can be of different nature which may co-occur:

- **Temporal**: the data describes events over time
- **Sequential**: the data spans some ordering
- **Relational**: the data describes event in between instances
- **Spatial**: the data describes space
- **Independent**: instances in data are independent observations

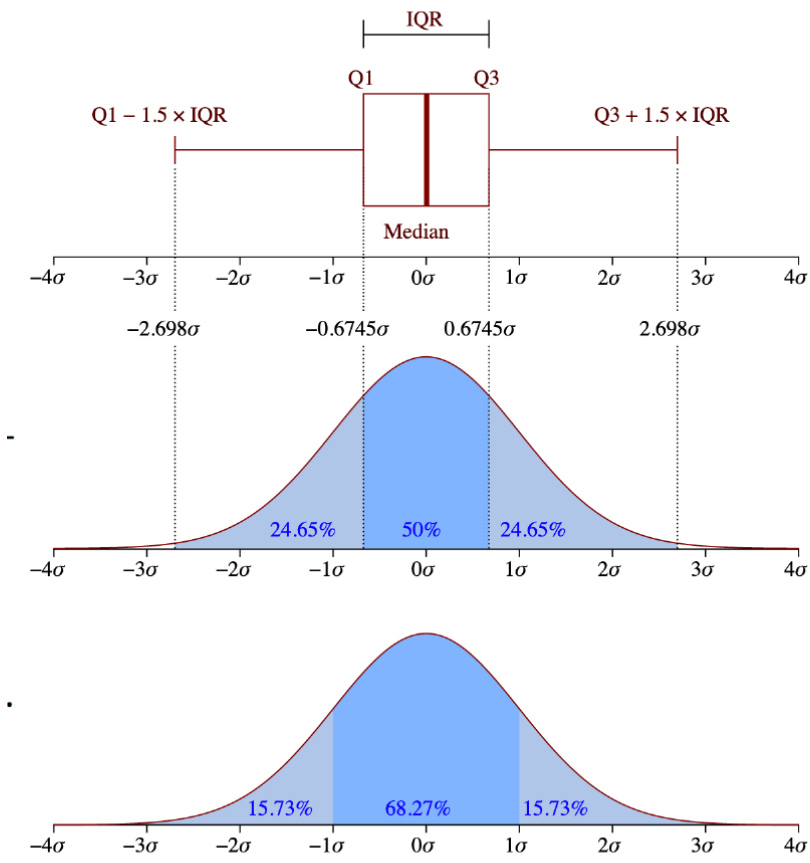


Figure 1.5: Box-Plot

1.3.1 Data Collections

We refer to single instances in the collections as objects/records/instances, which are described by attributes.

Id	Age	Income	Marital	Loan
0	30	2.5k	Married	Yes
1	24	1.4k	Single	No
...	...	...	...	...

Table 1.2: Grant Data

- Attributes: `Id , Age , Income , Marital,Loan grant`
- Records: `0, 30, 2.5k, Married, Yes , 1, 24,1.4k, Single, No`

1.3.2 Data Types

1.3.2.1 Tabular

When records are independent, and described by the same finite set of features, they are often represented in a tabular form: the data matrix. Each row is a record, each dimension is an attribute.

Records on the rows, attributes on the columns.

Id	Age	Bike used	Length	Duration	Date	Cyclist
0	28	Colnago VRS4	152.4	3:43:12	15-5-2025	Alessandro Covi
1	40	Cervelo RS5	72.4	2:55:01	4-3-2024	Gianni Affino

Table 1.3: Cyclist Data

1.3.2.2 Transaction

A feature contains a (multi)set of items.

PurchaseId	Cart	Bought on
0	Bread, Milk	17:12-15-5-2025
1	Notebook, Pens, Bread, Basil	8:04-4-3-2024

Table 1.4: Transaction Data

Records on the rows, attributes on the columns.

1.3.2.3 Graph

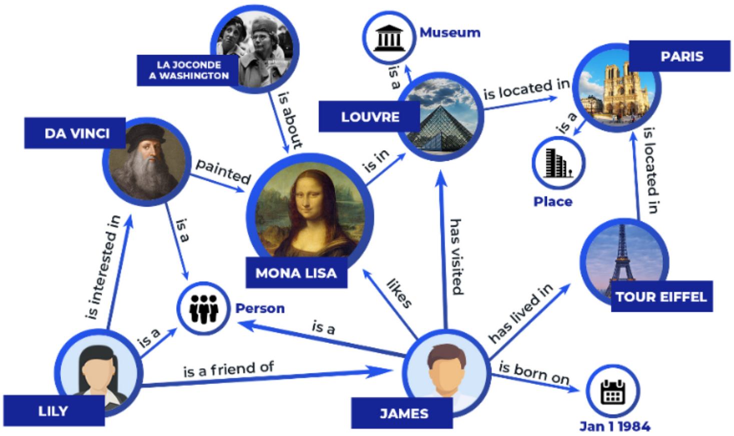


Figure 1.6: Graph Data

Data is linked, either on records or features. Records are nodes in a graph, attributes can vary wildly across records.

1.3.2.4 Sequential

Records are sequences (of variable length): attributes are indexed (order or time).

Image on the right lacks two images ☺

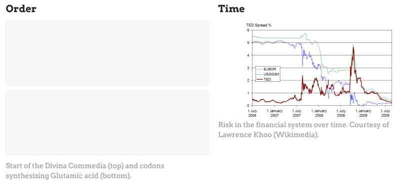


Figure 1.7: Sequential Data

1.3.2.5 Spatial

Records are associated with locations in space: attributes can include coordinates, regions, etc.

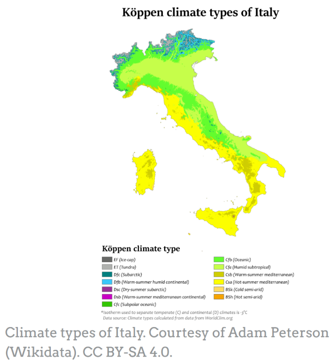


Figure 1.8: Spatial Data

1.3.2.6 Attribute types

Type	Description	Example
Numerical	Values have a total ordering, and represent some numerical quantity	Age, dates
Ordinal	Values have a total ordering, and represent some quantity	Dress size, Cup size
Binary	Values are one of two categories: no ordering	Boolean values
Categorical	Values of one of multiple categories: no ordering	Country, Job

Table 1.5: Types of Data

1.3.2.7 Values types

Values can be either:

- **Discrete** Defined in a finite or countably finite domain, e.g., country, job, cup size. Note: ordinal values may be discrete too!
- **Continuous** Defined in a continuous and infinite domain, e.g., distance.

1.3.3 Data Syntax and Semantics

Given the categorization of the records and attributes of your data, we can study its general behavior. We leverage some basic statistical tools, first of all by drawing the empirical distribution of the attributes.

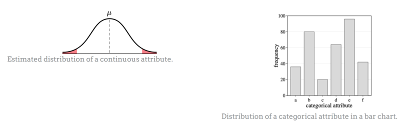


Figure 1.9: Data Syntax and Semantics

Useful statistics for data semantics

◇ **Expected value**

$$\mathbb{E}[X] = \sum_{x \in \mathrm{dom}(X)} \Pr(X = x)\, x$$

A statistic representative of the value of an attribute, weighing values and their probability

◇ **Variance**

$$\sigma^2(X) = \mathbb{E}\left[\sum_{x \in \mathrm{dom}(X)} \left(x - \mathbb{E}[X]\right)^2\right]$$

Distance from the expected value of all records: the data spread

◇ **Quantiles**

$$q^p = x \;\; s.t. \;\; \Pr(X \leq x) = q^p$$

Inflection points defining values for a threshold, e.g., if the 99-th percentile is , then we

◇ **Interquantile range**

$$q^{75} - q^{25}$$

Distance between quantiles: how spread are inflection points?

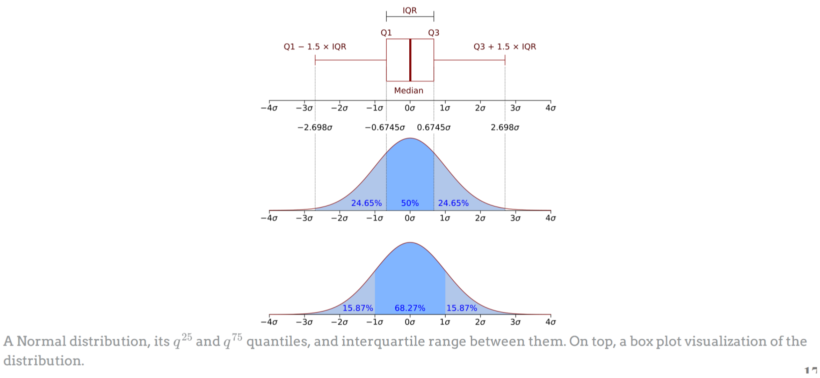


Figure 1.7: Statistics Graph

Statistical summary of the distribution are typically accompanied by visual and semantic one.
Erroneous or weird values to be cleaned later may already pop up in these basic steps. Outlier values typically skew statistics. Variance is often replaced by absolute/median average deviation

1.4 Data Cleaning

There are some concepts to be aware of when dealing with data quality, hence data cleaning.

Data accuracy is the degree to which data correctly describes the "real world" object or event being described.

- Syntactic: values outside domain, e.g., Eataly in Country
- Semantic: values in domain, but semantically wrong, e.g., age is 3, and weight is 82kg

Completeness is the degree to which all required data is known.
Some attributes are not collected, or are collected partially, e.g., temperature was not recorded by the sensor.

Biased gathering is the degree to which data may be over/under-representative, e.g., the bank may only provide data about successful loan applicants.

Timeliness is the degree to which data is up to date.

Remember: *garbage in, garbage out!*[1] In a task-agnostic view, we are interested in addressing the above by tackling:

- **Duplicates**: skews the data distribution

[1] i.e. if you have garbage data, you'll get garbage results

- **Missing values**: give false/partial information
- **Noise**: uninformative of the data
- **Poor accuracy**: gives wrong data
- **Outliers**: skews the data distribution and models of the data

1.4.1 Handling Duplicates

Remove them... when appropriate! Not all duplicates are garbage, it depends on what insight you can gather from it.

Case A
You have data on registration to your website, with several duplicate e-mails. Insights:

- The "Sign in" button is hard to find
- The "Sign in" button is less visible than the "Sign up" button
- Your site is so anonymous people forget they signed up already

Case B
You have data on credit account opening from Poste (Italian postal service) with several duplicate e-mails. Insights:

- The client hacked the database and added themselves to ask more credit (unlikely)
- Poste's tech staff is underwhelming (very likely)

1.4.1.1 Duplicate Features

Duplicate features may be more tricky. Features convey similar, although not equal, information to others.
Examples:

- Resting heart rate and heart rate under continuous high effort
- Education level and reading skills
- Rent and available bank deposit

These pairs of features are not per se one duplicate of the other, but are strongly related: when one grows, so does the other, and when one goes down, so does the other.

Linear (and rank) relationships between two features X, Y can be quantified with their correlation. Correlation ranges in $[-1, 1]$, from perfectly negative to perfectly positive correlation.
Given two lists of values x^{i^n}, y^{i^n} we can compute two main correlation types.

- **Pearson correlation**
$$\rho_P^{X,Y} = \frac{\mathbb{E}\left[(x^i - \mathbb{E}[X])(y^i - \mathbb{E}[Y])\right]}{\sigma_X \sigma_Y}$$
Measures linear correlation between two numerical features and their values.
- **Spearman correlation**
$$\rho_S = \rho_P^{\text{rank}(X),\text{rank}(Y)}$$
Measures monotonic correlation between two ordinal or numerical features.

1.4.2 Handling Missing Values

Data may be missing for any number of reasons (at random or not at random).

- A record has a large and/or significant set of missing attributes
- An attribute has a large percentage of missing values

We have two choices: **dropping** or **imputing**.

Dropping
If a record has a large and/or significant set of missing attributes, or an attribute has a large percentage of missing values, we can drop the record/attribute.

- High percentage of missing values
- Missing values in critical attributes, e.g., a patient in cardiology has no heart rate data

Imputing
Imputing means replacing the missing value with a "best guess" value.
If a record has a small set of missing attributes, or an attribute has a small percentage of missing values, we can impute the missing values. We have to create a model to predict the missing value.

- Low percentage of missing values
- Reasonably good understanding of the attribute semantics/distribution

◇ Presence of related attributes

1.4.3 Outliers

Quantiles and distributions inform us on what values may be outlier. They are typically dropped, and unlike missing values, almost never imputed. We'll tackle algorithms later in the course.

1.4.3.1 Flower Example

There is a dataset with 5 attributes: sepal length, sepal width, petal length, petal width, and species (type).

Sepal L.	Sepal W.	Petal L.	Petal W.	Type
5.1	3.5	1.4	0.2	Setosa
7.0	3.2	4.7	1.4	Versicolor
...	...	...	...	...

Table 1.6: Flower Dataset Example

Figure 1.8: Flower Data plotted

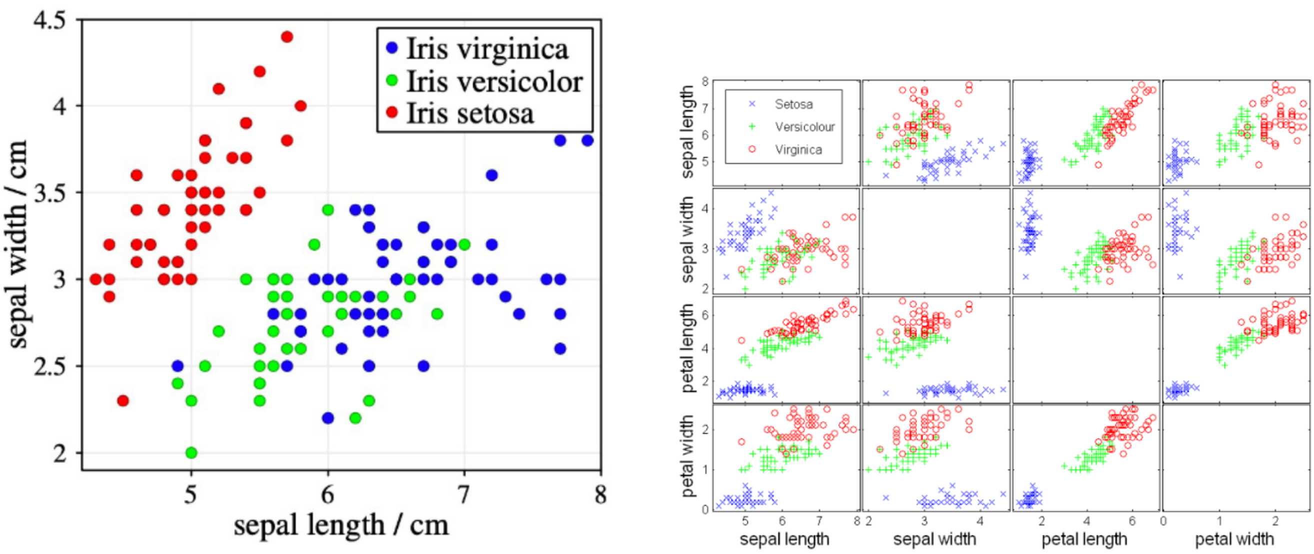


Figure 1.9: Scatter plot of sepal length and width, and scatter matrix: scatter plots of all pairs of attributes in the Iris dataset.

Plot bivariate (or trivariate) data, eyeing data correlation and outliers.

1.5 Data Preparation

We will delve into the following techniques of data preparation:

◇ Aggregation
◇ Data Reduction: Sampling
◇ Dimensionality Reduction

- ◇ Feature subset selection
- ◇ Feature creation
- ◇ Discretization and Binarization
- ◇ Attribute Transformation

1.5.1 Aggregation

Aggregation is the process of combining two or more attributes (or objects) into a single attribute (or object).

Purpose

- ◇ Data reduction
  - – Reduce the number of attributes or objects
- ◇ Change of scale
  - – Cities aggregated into regions, states, countries, etc.
  - – Days aggregated into weeks, months, or years
- ◇ More "stable" data
  - – Aggregated data tends to have less variability

1.5.2 Reduction

Reduction is simply reducing the amount of data We may reduce the number of **records** by sampling or clustering, or the number of **attributes** (*columns*) by selecting a subset of them, or by creating a new —smaller— set of attributes from the old one.

1.5.2.1 Sampling

Sampling is the main technique employed for data reduction.
It is often used for both the preliminary investigation of the data and the final data analysis.

Sampling is typically used in data mining because processing the entire set of data of interest is too expensive or time consuming.

The key principle for effective sampling is the following:

- ◇ Using a sample will work almost as well as using the entire data set, if the sample is representative
- ◇ A sample is representative if it has approximately the same properties (of interest) as the original set of data

- ◇ **Simple Random Sampling**
  - – There is an *equal probability* of selecting any particular item
  - – Sampling **without replacement**
    - ◦ As each item is selected, it is removed from the population
  - – Sampling **with replacement**
    - ◦ Objects are not removed from the population as they are selected for the sample.
    - ◦ In sampling with replacement, the same object can be picked up more than once
  - – **Stratified sampling**
  - – Split the data into several partitions; then draw random samples from each partition
  - – Approximation of the percentage of each class
  - – Suitable for distribution with peaks: each peak is a **layer**

1.5.2.2 Dimensionality Reduction

This consists in reducing the number of attributes (or features) in the data. We want a selection of a subset of attributes that is as small as possible and sufficient for the data analysis.

- ◇ removing (more or less) irrelevant features
  - – Contain no information that is useful for the data mining task at hand
  - – Example: students' ID is often irrelevant to the task of predicting students' GPA
- ◇ removing redundant features
  - – Duplicate much or all of the information contained in one or more other attributes
  - – Example: purchase price of a product and the amount of sales tax paid

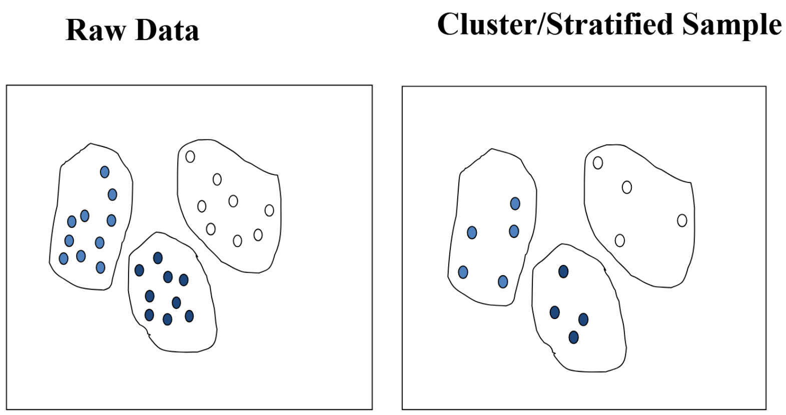


Figure 1.10: Stratified Sampling

Curse of Dimensionality

When dimensionality increases, data becomes **increasingly sparse** in the space that it occupies.

Definitions of density and distance between points, which are critical for clustering and outlier detection, become less meaningful.

This phenomenon is known as the **curse of dimensionality**.

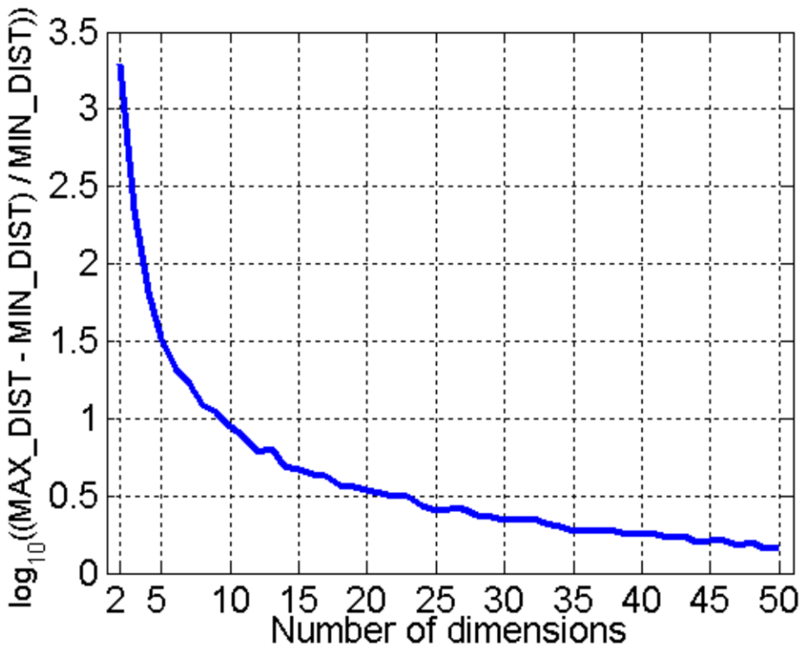


Figure 1.11: $log_{(MAXDIST-MINDIST)/MINDIST}$ decreases as the dimensionality increases, meaning that the difference between the farthest and nearest neighbor distances becomes less significant

Purposes of dimensionality reduction include:

- ◇ Avoid curse of dimensionality
- ◇ Reduce amount of time and memory required by data mining algorithms
- ◇ Allow data to be more easily visualized
- ◇ May help to eliminate irrelevant features or reduce noise

Techniques to do so include:

- ◇ Principal Components Analysis (PCA)
- ◇ Singular Value Decomposition
- ◇ Others: supervised and non-linear techniques

1.5.2.3 Feature Subset Selection

Feature subset selection consists in selecting a subset of the original features. The goal is to find a minimal subset of features that is as good as the entire set of features for the data analysis task at hand.

For removing irrelevant features, it is needed a **performance measure** indicating how well a feature or subset of features performs w.r.t. the considered data analysis task.

For removing **redundant features**, either a *performance measure* for subsets of features or a *correlation measure* is needed.

Filter Methods

- Selection after analyzing the **significance** and **correlation** with other attributes
- Selection is independent of any data mining task
- The operation is a pre-processing

Wrapper Methods

- Selecting the top-ranked features using as reference a DM task
- Incremental Selection of the “best” attributes
  “Best” = with respect to a specific measure of statistical significance (e.g.: information gain)

Embedded Methods

- Selection as part of the data mining algorithm
- During the operation of the DM algorithm, the algorithm itself decides which attributes to use and which to ignore (e.g. Decision tree)

Feature Selection Techniques

- **Selecting the top-ranked features**: Choose the features with the best evaluation when single features are evaluated.
- **Selecting the top-ranked subset**: Choose the subset of features with the best performance. This requires exhaustive search and is impossible for larger numbers of features. (For 20 features there are already more than one million possible subsets.)
- **Forward selection**: Start with the empty set of features and add features one by one. In each step, add the feature that yields the best improvement of the performance.
- **Backward elimination**: Start with the full set of features and remove features one by one. In each step, remove the feature that yields to the least decrease in performance.

Chapter 2

Data Representation

- By **correlation**
  I want to represent data according to the correlation of the dataset
  Algorithm: `PCA`
- By **neighborhood**
  I want to represent the data so that similar instances are similar
  Algorithm: `t-SNE`
- By **manifold**
  I want to represent the data so that its manifold is preserved
  Algorithm: `UMAP`

2.1 Principal Component Analysis (PCA)

PCA is a linear dimensionality reduction technique that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.

Essentially, PCA exploits spectral decomposition of the whole dataset to find a new basis for the data.

Data can often be correlated, and linear dependencies can exist among variables, e.g.,

- Rent is linearly dependent on salary and food expenses
- Bank deposit is linearly dependent on salary and work
- Cardio is linearly dependent on VO_2max

Vectors are m-dimensional elements in a field, and enjoy both addition and multiplication by scalar.
Composing these two, we can generate an infinite number of vectors: this is a **vector space**, and is defined by the basis vectors involved in the composition.

A matrix A defines a space...and thus a linear transformation! Av linearly combines the columns of with coefficients given by v.

Eigenpairs (λ, v) of a square matrix A are defined by the equation $Av = \lambda v$.

The eigenvectors $v_1, \dots, v_m$ of a matrix A define the stretching of the space, and their eigenvalues $\lambda_1 > \cdots > \lambda_m$ define the stretching factor.

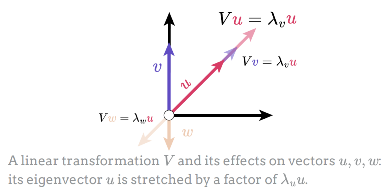
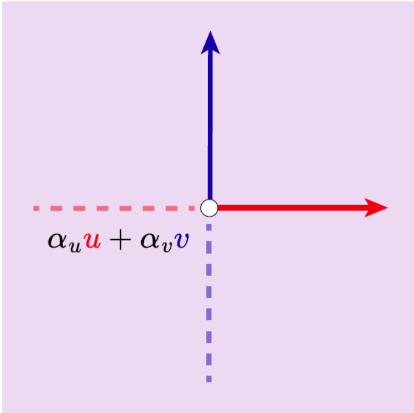


Figure 2.2: Eigenvectors of a matrix

PCA projects some data X to $\hat{X}$ through a linear transformation A: $AX = \hat{X}$.

Fun fact #1: for a mean-centered $\bar{X}$, the slope is directly proportional to the covariance!



Two vectors u, v (in red and blue), and the plane spanned by all their linear combinations $\alpha_u u + \alpha_v v$ (in purple).

Figure 2.1: Vector space spanned by two vectors

$$\bar{\Sigma} = \begin{bmatrix} \sigma^2_{\bar{X}^1} & \cdots & \text{cov}(\bar{X}^1, \bar{X}^n) \\ \cdots & \cdots & \cdots \\ & & \sigma^2_{\bar{X}^n} \end{bmatrix}$$

There is some pretty complicated linear algebra behind PCA, but the main steps are the following:

1. Mean-center your data X to get $\bar{X}$
2. Compute its eigenvectors matrix $\bar{V}$
3. Transpose V to obtain the transformation V^T
4. Project the data: $\bar{X}$ through $V^T\bar{X}$, obtaining the PCA-transformed data $\hat{X}$

In simple terms: PCA looks at your data and finds the "most important directions" - imagine you have a cloud of points and you want to find the best line that captures the main trend. PCA finds not just one line, but multiple directions ordered by importance. It then rotates your data so that the first dimension captures the most variation, the second dimension captures the second most variation. This allows you to keep only the first two dimensions while retaining most of the information, effectively reducing the complexity of your data while preserving its essential structure.

2.1.1 Observations

- PCA redefines data by removing collinearity: if your data has low covariance, the transformation will have minimal effect.
- PCA performs a linear transformation to tackle linear relationships between variables. Nonlinear relationships are not influenced.
- PCA does not work very well for high complexity data.

Uses

- **Feature selection**: high covariance of a feature may indicate disposability.
- **Dimensionality reduction**: trimming columns of lets us reduce the dimension of the resulting data.
- **Clustering preprocessing**: correlated features inflate object similarity

2.2 t-SNE

t-SNE (t-distributed Stochastic Neighbor Embedding) is a nonlinear dimensionality reduction technique particularly well suited for embedding high-dimensional data for visualization in a low-dimensional space of two or three dimensions. It works by modeling the data as a distribution of points in a high-dimensional space and then finding a lower-dimensional representation that preserves the pairwise similarities between points.

`t-SNE` focuses on data clusters rather than subspace representation, and again maps the original data X to a representation $\hat{X}$.

`t-SNE` tackles this problem in two phases:

1. **Similarity phase** In the original space $\mathcal{X}$, how similar is x_i to x_j?

How similar is x_i to x_j? Even better, what is the probability that x_i is a neighbor of x_j?

2. **Embedding phase** In the mapped space $\hat{\mathcal{X}}$, how similar is $\hat{x}_i$ to $\hat{x}_j$?

2.2.1 Similarity phase

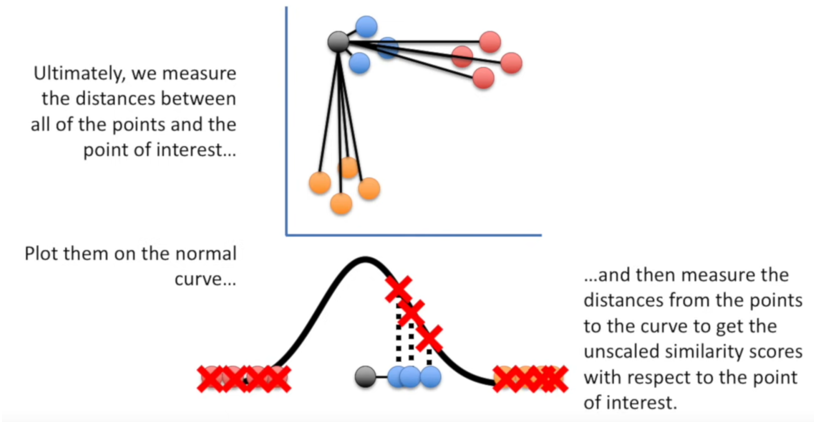


Figure 2.2: t-SNE plotting distance on an X axis and then projecting it on a normal distribution curve, to get the probability of being a neighbor

The similarity phase computes the similarity between points in the original high-dimensional space. This is typically done by converting the Euclidean distances between points into conditional probabilities that represent similarities. The probability that point x_j is a neighbor of point x_i is given by a Gaussian distribution centered at x_i. The variance of this Gaussian is controlled by a parameter called **perplexity**, which can be thought of as a smooth measure of the effective number of neighbors.

This yields a neighboring matrix P where each entry p_{ij} represents the probability that point x_j is a neighbor of point x_i in the original high-dimensional space.

The conditional probability is computed as:

$$p_{j|i} = \frac{\exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-||x_i - x_k||^2/2\sigma_i^2)}$$

where σ_i is the variance of the Gaussian centered at point x_i. The perplexity parameter determines σ_i through a binary search to match the desired effective number of neighbors.

2.2.2 Embedding phase

In the embedding phase, t-SNE defines a similar probability distribution over the points in the low-dimensional map. However, instead of using a Gaussian distribution, it uses a Student's t-distribution with one degree of freedom (heavy-tailed distribution) to avoid the "crowding problem" where moderate distant points are forced to be too far apart in the low-dimensional representation.

The probability in the low-dimensional space is:

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k\neq l}(1 + ||y_k - y_l||^2)^{-1}}$$

where y_i and y_j are the low-dimensional counterparts of x_i and x_j.

2.2.3 Optimization

t-SNE minimizes the Kullback-Leibler divergence between the probability distributions P (high-dimensional) and Q (low-dimensional):

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

The algorithm uses gradient descent to find the low-dimensional representation Y that minimizes this cost function, effectively preserving the neighborhood structure of the original high-dimensional data.

Key advantages:

- ◇ Excellent for visualization of high-dimensional data
- ◇ Preserves local neighborhood structure
- ◇ Can reveal clusters and patterns not visible in linear methods

Key limitations:

- ◇ Computationally expensive (quadratic in the number of points)
- ◇ Non-deterministic (different runs can give different results)
- ◇ Sensitive to hyperparameters, especially perplexity
- ◇ Not suitable for embedding new data points (no explicit mapping function)

2.3 UMAP

UMAP (Uniform Manifold Approximation and Projection) is a nonlinear dimensionality reduction technique that is particularly effective for visualizing high-dimensional data in a low-dimensional space. It is based on manifold learning and topological data analysis, aiming to preserve both local and global structure of the data.

The computed distances induce a connectivity graph, and thus an adjacency matrix A, its edges measuring distances among instances. After turning distances into probabilities, UMAP optimizes a distance on A, to make it so that all and only the edges on the original manifold also appear in the transformed manifold with the same magnitude.

For the set of edges E, UMAP minimizes

$$-\sum_{e \in E} \left(\underbrace{\Pr(e; X) \log(\Pr(e; Z))}_{\text{existing edges}} + \underbrace{(1 - \Pr(e; Z)) \log(1 - \Pr(e; X))}_{\text{non-existing edges}} \right),$$

where $\Pr(e; X), \Pr(e; Z)$ indicate the probability of edge e in the original and transformed representation, respectively.