Lecture 8

Unsupervised Learning

Ashish Dandekar

Lecture Overview

K-Means

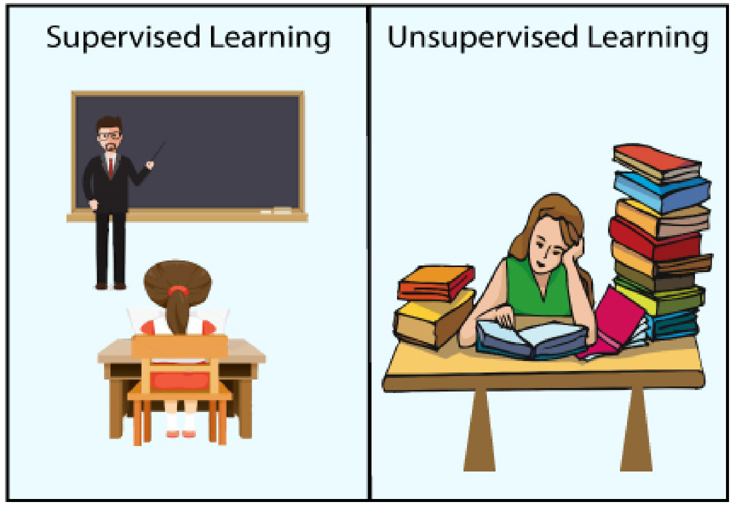
Kernel Density Estimation

Frequent Patterns Mining

Unsupervised learning

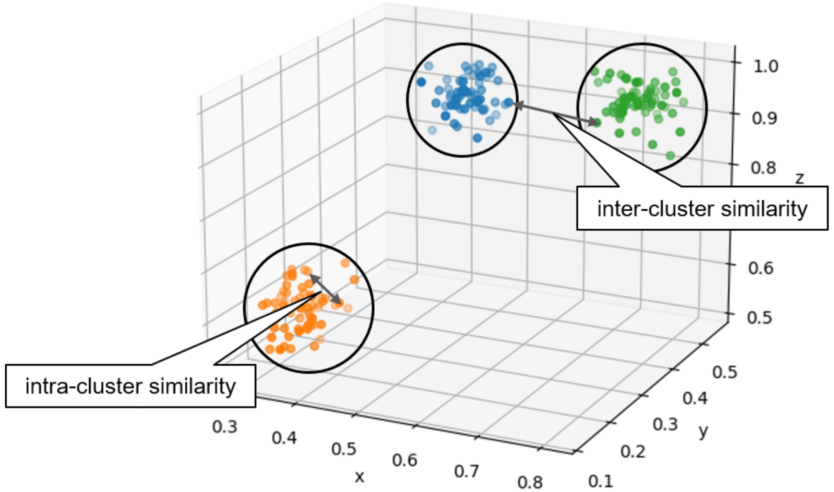
Unsupervised learning is one that lets us observe the data systematically, holistically, objectively, and often creatively to discover the nuances of the underlying process that generated the data, the grammar in the data, and insights that we didn't know existed in the data in the first place.

- Clustering
- Density estimation
- Pattern mining



Clustering

Introduction



Clustering aims at finding groups of **similar** objects in the **unlabeled** dataset.

- Maximise intra-cluster similarity
- Minimise inter-cluster similarity

Deciding the number of good/meaningful/useful set of clusters is not obvious!

Applications

Market segmentation

- Group customers based on behaviour and/or preferences
- Design targeted campaigns for customers according to the clusters

Recommendation systems

- Group items based on their attributes
- Recommend items from a cluster to user who has liked similar items

Web search diversification

- Group webpages based on the content
- Return search results from different clusters to ensure diversity

...

Ingredients

Representation

- Points in Euclidean Space
- Sets
- Vectors

Clustering algorithm

Similarity metrics

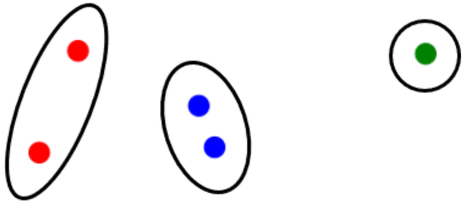
$$d_{euclidean}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

$$d_{jaccard}(A, B) = \frac{| A \cap B |}{| A \cup B |}$$

$$d_{cosine}(u, v) = \frac{u \cdot v}{// u // \; // v //}$$

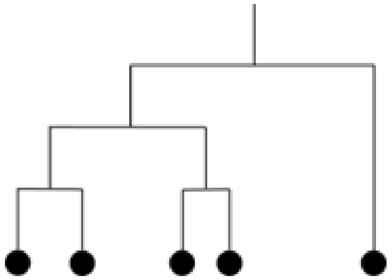
Types of Clustering

Partitional



- Non-overlapping clusters
- Each object exactly belongs to one cluster

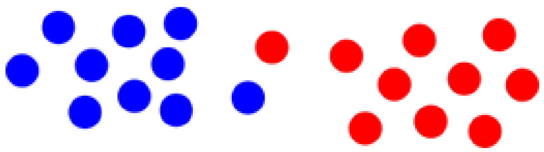
Hierarchical



- Clusters can be nested
- A point can belong to different clusters depending on the level

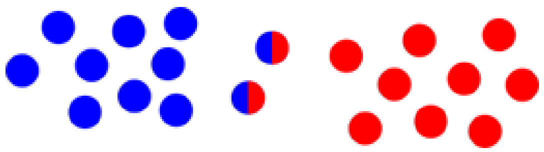
Types of Clustering

Exclusive



- Each object exactly belongs to one cluster

Overlapping



- A point can belong to more than one cluster at a time
- Fuzzy clustering: each object belongs to all clusters with a certain probability.

K-Means Clustering

Given an unlabeled dataset $D = \{x_1, x_2, ..., x_n\}$. We want to partition it in K clusters where $\{c_1, c_2, ..., c_K\}$ denote the *cluster representatives*.

Let, the membership of i^{th} datapoint to the j^{th} cluster is denoted as

$$\delta_{ij} = \begin{cases} 1 & x_i \in c_j \\ 0 & x_i \notin c_j \end{cases}$$

We want to minimise:

$$SSE = \sum_{i=1}^{n} \sum_{j=1}^{K} \delta_{ij} d(x_i, c_j)$$

- Finding optimal solution is NP-hard.
- We will resort to a greedy solution (which may be sub-optimal)

Greedy Solution

Let's use Euclidean distance: $d(x_i, y_j) = // x_i - y_j //^2$.

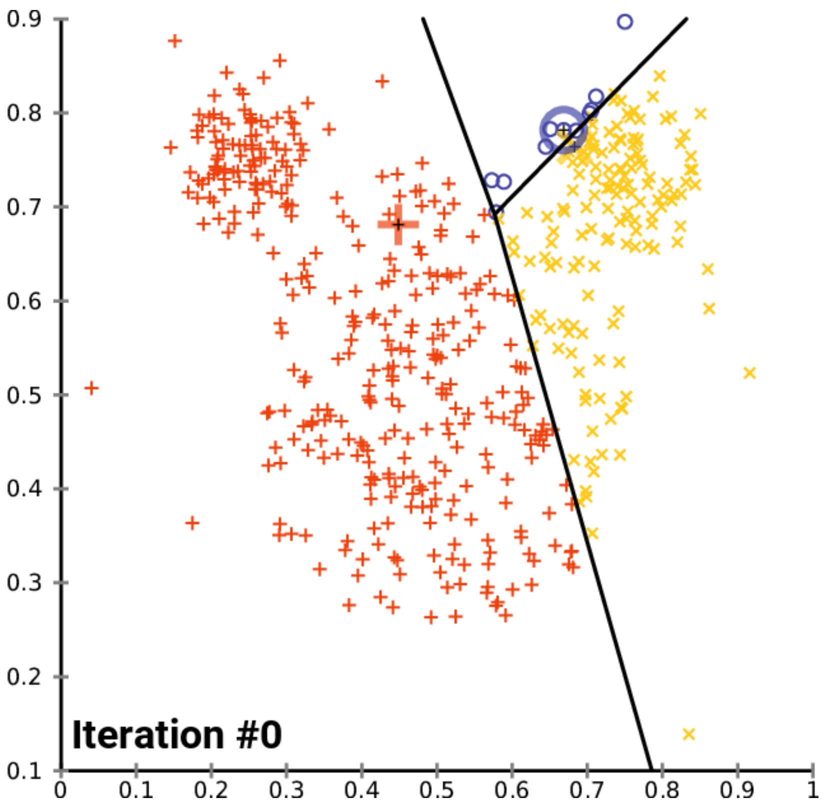
- **Assign** every point x_i to its closest cluster

$$\delta_{ij}^t \leftarrow 1 \quad \text{iff} \quad j = \underset{j \in [1...K]}{\arg\min}\, d(x_i, c_j^t)$$

- **Update** the cluster representatives (a.k.a centroid)

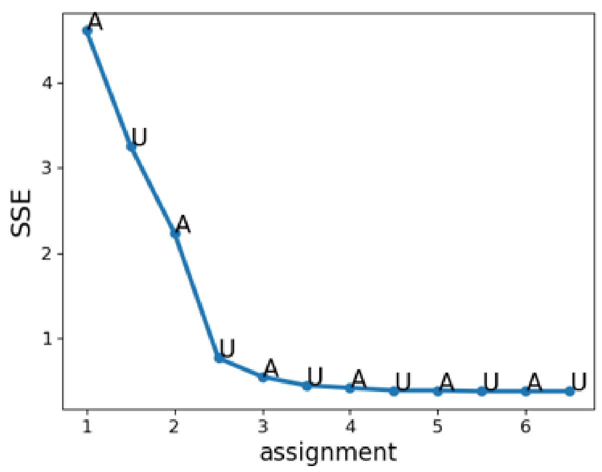
$$c_i^{t+1} = \frac{\sum_{x \in c_i^t} x}{\mid c_i^t \mid}$$

Simulation



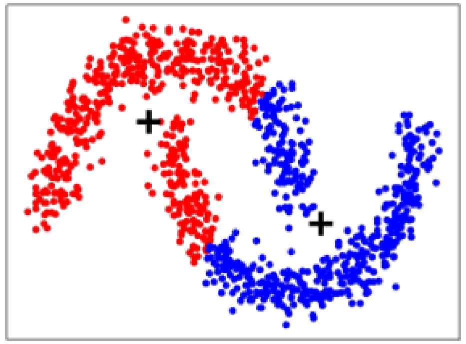
Convergence

- K-Means always converges.
- Though the process may converge to local optimal.
- Most improvement during the initial iterations.
- Initialisation of centroid may change the answers!

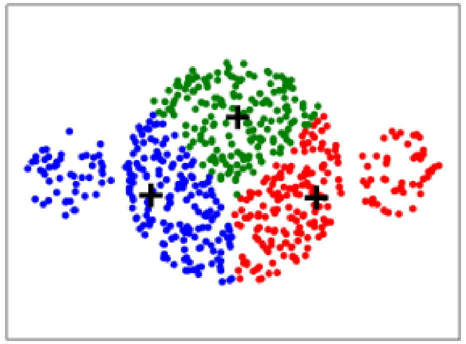


Limitations

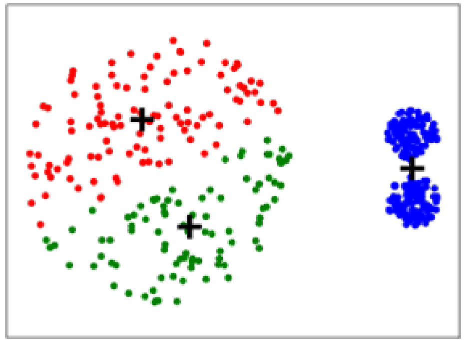
Non-spherical Clusters



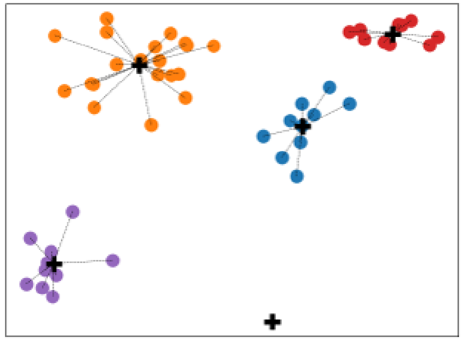
Clusters of different sizes



Clusters of different densities



Empty clusters



Variants

K-Means++

- It starts with one random point as a centroid, and sequentially chooses $K-1$ centroids that are well spread out.
- It tends to avoid empty clusters.

X-Means

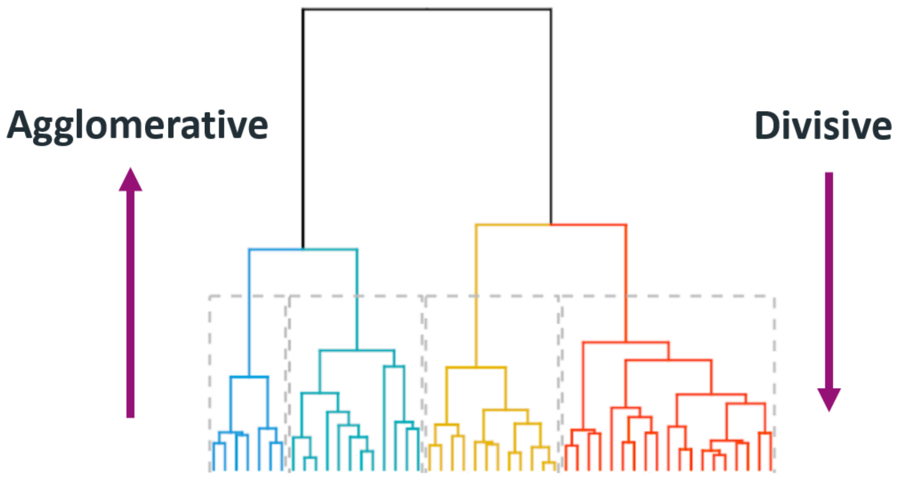
- Run K-Means with $K=2$.
- Iteratively run K_Means on each cluster with $K=2$.
- Split each cluster further using a scoring functions (such as Bayesian Information Criterion, Akaike Information Criterion, etc.)
- It automaticallly chooses the value of K at the end of the iterations.

Variants

K-Medoids.

- Centroids may not exist in the data whereas medoids are centroids that are chosen from the data points.
- Expensive Update. Swap medoid with each point in cluster and calculate change in the cost (SSE). Choose medoid that minimises the cost.
- More robus to outliers and noise.

Hierarchical Clustering



Agglomerative Clustering (AGNES)

- Every point is its own cluster.
- Two clusters are recursively merged based on a criterion.

Divisive Clustering (DIANA)

- Entire dataset lies in one big cluster.
- The cluster is recursively divided into two clusters based of paritioning criterion.

Linkage

Merge two clusters if *distance between them* is smaller than a threshold!

Single Linkage.

Distance between the closest points from the clusters.

Average Linkage.

Average pairwise distance between points in the two clusters.

Complete Linkage.

Distance between the farthest points from the clusters.

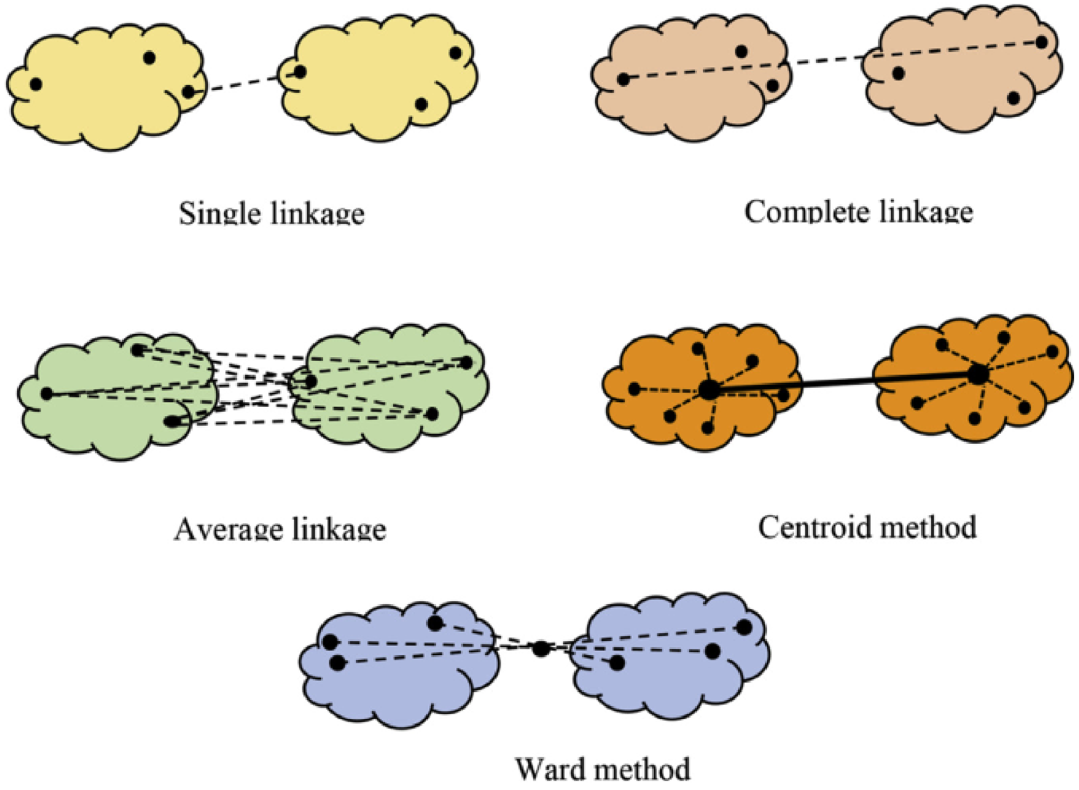
Centroid Linkage.

Distance between the centroids of the two clusters.

Ward Linkage

Change in the distance to the centroids if the clusters were merged.

Linkage



Single linkage
Complete linkage
Average linkage
Centroid method
Ward method

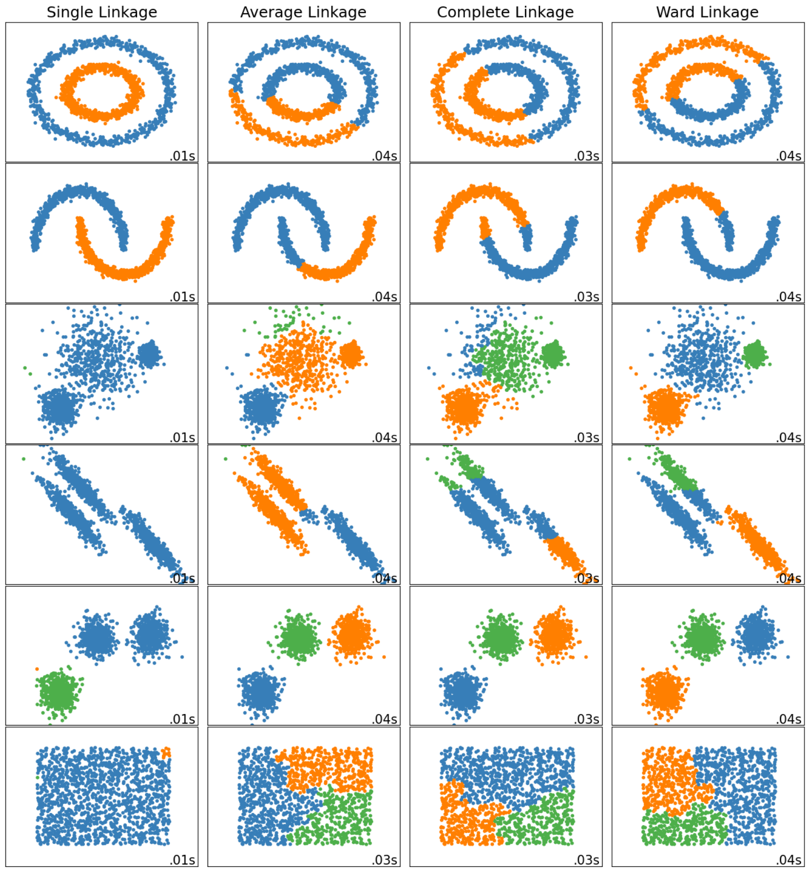
Linkage

Single Linkage

- Ability to handle non-globular clusters
- Very susceptible to noise (a single point may cause two clusters to be merged).

Complete Linkage

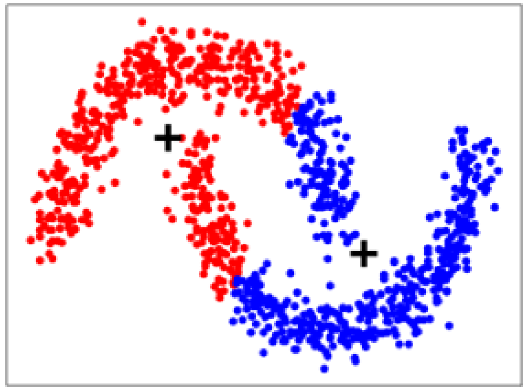
- Less susceptible to noise
- Bias towards globular clusters



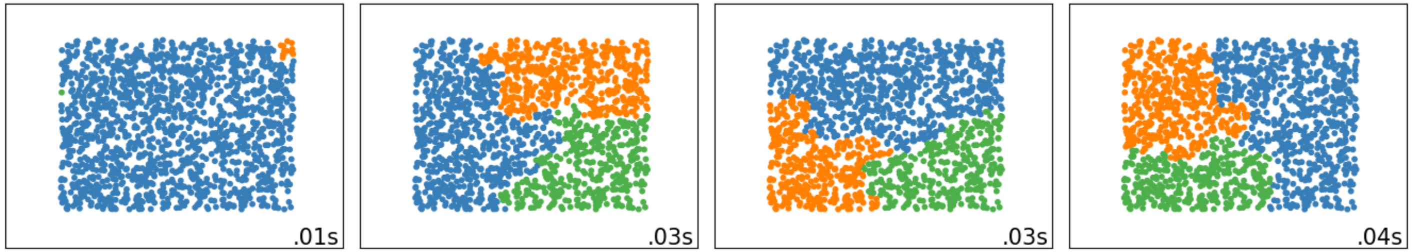
Is it always possible?

Eyeballing the clusters

Data is messy most of the times.



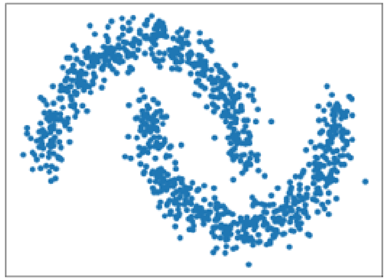
Algorithms always finds some clusters



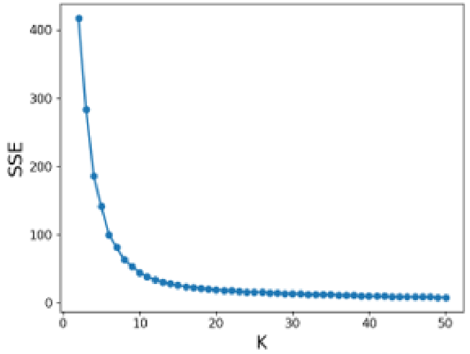
Elbow Method

Introduction
Ingredients
Types
K-Means
Hierarchical
❯ Evaluation

Input Data

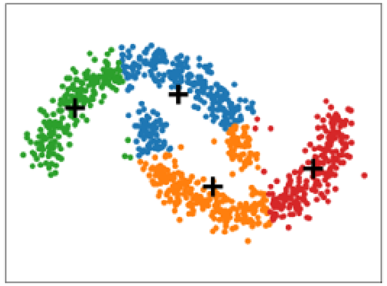


Check for various values of K

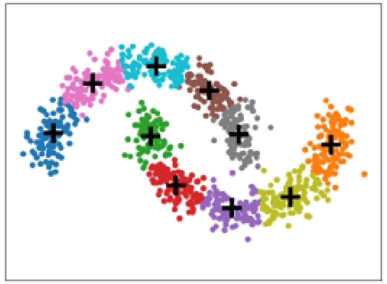


But K-Means inherently favours globular clusters!

K = 4



K = 10



General Approaches

Heuristics

- Fixed number of clusters
- Parameters defined by the task
- Focus on some clusters than overall effectiveness

External quality metrics

- Evaluate a clustering against a ground truth (if available).
- Use any metrics that you would use to evaluate classification.

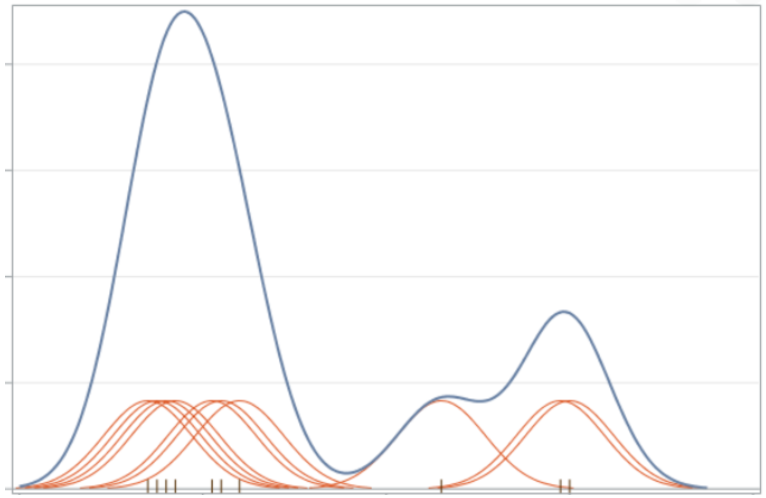
Internal quality metrics

- Intra-cluster distance (SSE)
- Inter-cluster distance

Kernel Density Estimation

Density Estimation

- **Introduction**
- Parzen Window
- Non-Parametric Learning



A wonderful website: KDE

Density Estimation provides a way of an exploratory analysis of the distribution of data. It can also be used to do

- Data imputation
- Outlier detection
- Data denoising

Parzen Window

Given a set of datapoints $x_1, x_2, ..., x_n$, the probability density of a new datapoint x is given as:

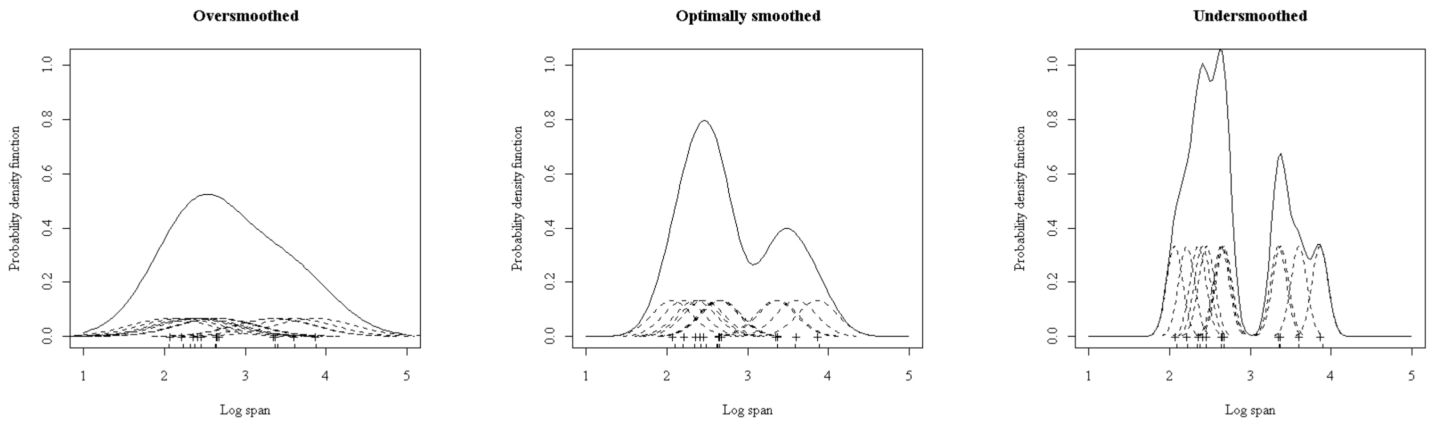
$$p(x) = \frac{1}{n} \sum_{i=1}^{n} K_h(x, x_i)$$

where K is called as the *kernel* with the *bandwidth* h.

For instance using the Gaussian Kernel

$$p(x) = \frac{1}{n\sqrt{2\pi h^2}} \sum_{i=1}^{n} exp\left(\frac{// x - x_i //^2}{2h^2}\right)$$

Effect of Bandwidth



Source: KDE

Smaller bandwidth implies narrower field of influence of individual points.

Non-Parametric Learning

Parametric models

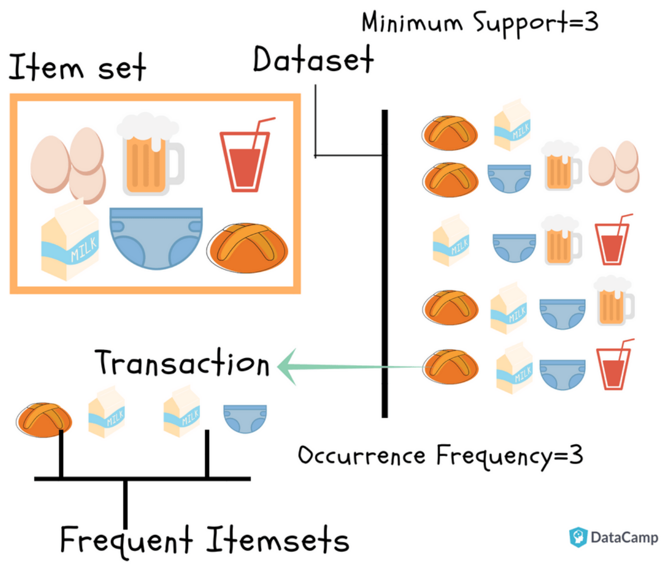
- They assume that the latent patterns in the data are captured by a finite set of parameters.
- They have a dedicated *training* phase to estimate the parameters.
- For instance: Linear regression, logistic regression, etc

Non-parametric models

- Parameters grow proportional to the number of datapoints. The parameters of the non-parametric models are called as *hyperparameters*.
- They are *lazy* learners.
- For instance: Parzen window estimation, decision trees, etc

Frequent Pattern Mining

Market Basket Analysis



Source: Datacamp

Finding the shopping patterns of the buyers to desing new marketing strategies.

- Changing store layouts
- Designing sales campaigns

Itemsets

Introduction
❯ Terminology
Apriori
Lift

Itemset

- It is the non-empty set of items.
- Examples. `{milk}`, `{yogurt, bread}`

K-itemset

- An itemset that contains K items.
- `{milk, bread, eggs}`, `{eggs, cheese, bread}` are examples of 3-itemsets.

TID	Items
1	bread, yogurt
2	bread, milk , cereal, eggs
3	bread, yogurt, milk, cheese, cereal
4	bread, milk, cereal

Support

Fraction of transactions containing the itemset.

$$support(\{milk, bread\}) = 3/4$$

Association Rules

Association Rule

- Inference of the form $X \Rightarrow Y$ where X and Y are itemsets.
- Examples. `{yogurt, milk}` $\Rightarrow$ `{milk}`

TID	Items
1	bread, yogurt
2	bread, milk , cereal, eggs
3	bread, yogurt, milk, cheese, cereal
4	bread, milk, cereal

Support of association rule

- Fraction of transactions containing the itemset from the association rule.

$$support(\{yogurt, milk\} \rightarrow \{bread\}) = 1/4$$

Confidence

Introduction
❯ Terminology
Apriori
Lift

Confidene

- Confidence of an association rule $X \Rightarrow Y$ is the probablity of Y given X

$$confidence(X \Rightarrow Y) = \frac{support(X \Rightarrow Y)}{support(X)}$$

TID	Items
1	bread, yogurt
2	yogurt, milk , cereal, eggs
3	bread, yogurt, milk, cheese, cereal
4	bread, milk, cereal

$$confidence(\{yogurt, milk\} \rightarrow \{bread\}) = \frac{1}{1}$$

Interesting Rules

	Low Support	High Support
Low Confidence	The items occur infrequently.	The items occur freuquently but if the items in X appear together, they often appear without the items in Y
High Confidence	The items occur infrequently.	The items occur frequently and if the items in X appear together, they often appear **with** the items in Y

We want to filter the association rules that have high support and high confidence!

Apriori Algorithm

The interestingness of the association rule can be quantified by two parameters:

Frequent itemset

An itemset X is said to be a frequent itemset if $support(X) \geq min\_sup$

Strong rule

An association rule $X \Rightarrow Y$ is said to be a strong rule if $confidence(X \Rightarrow Y) \geq min\_conf$

Apriori Algorithm

Introduction
Terminology
❯ Apriori
Lift

The interestingness of the association rule can be quantified by two parameters:

Frequent itemset

An itemset X is said to be a frequent itemset if $support(X) \geq min\_sup$

Strong rule

An association rule $X \Rightarrow Y$ is said to be a strong rule if $confidence(X \Rightarrow Y) \geq min\_conf$

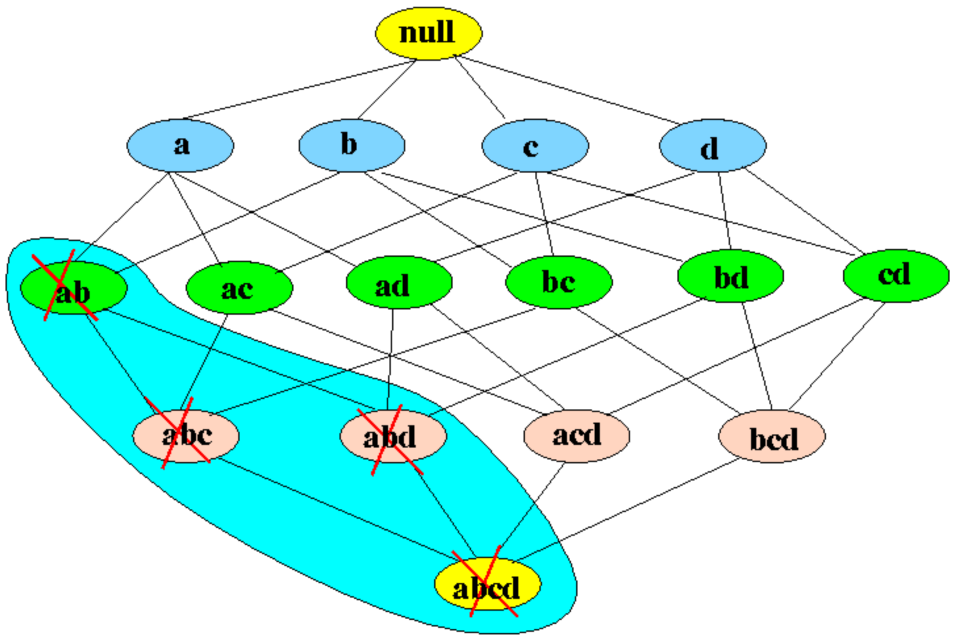
Brute-force approach of finding the frquent patterns grows exponential with the number of items :(

Apriori Algorithm

Introduction
Terminology
❯ Apriori
Lift

Apriori algorithm trick.

If an itemset X is infrequent so is any of its superset!



Lift

- Lift is an alternative measure to discover the intersestingness of the association rules.
- Lift measures the correlation (**not** causation) between item sets.

$$lift(X \Rightarrow Y) = \frac{support(X \Rightarrow Y)}{support(X) \times support(Y)}$$

- $lift < 1$. Asserts negative correlation between the itemsets.
- $lift = 1$. Asserts independence between the itemsets.
- $lift > 1$. Asserts positive correlation between the itemsets.

Example