COMPSCI762: Introduction to Machine Learning

Clustering

Jörg Simon Wicker and Katerina Taškova

The University of Auckland

SCIENCE

SCHOOL OF COMPUTER SCIENCE

This block will cover...

Unsupervised Learning

Clustering

- K-Means
- Density-Based Clustering
- Hierarchical Clustering
  - Agglomerative Clustering
  - Divisive Clustering
- Cluster Quality

Partly based on the lecture slides from University of British Columbia CPSC340

Unsupervised Learning

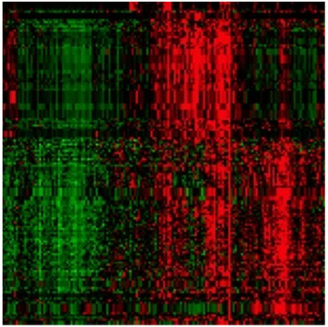
Unsupervised Learning

- Supervised learning
  - We have n instances in d-dimensional space X, $x_i < x_{i1}, \ldots, x_{ij}, \ldots, x_{id} >$, and class labels y_i, $1 \leq i \leq n$
  - Write a program that produces y_i from x_i
- Unsupervised learning
  - We **only have** x_{ij} **values**, but **no explicit target** (i.e. class labels)
  - You want to do "something" with them
- Some unsupervised learning tasks
  - Outlier detection: Is this a 'normal' x_i?
  - Similarity search: Which instances look like this x_i?
  - Association rules: Which feature values occur together?
  - Latent-factors: What 'parts' are the x_i made from?
  - Data visualization: What does the high-dimensional X look like?
  - Ranking: Which are the most important x_i?
  - Clustering: What types of x_i are there?

Clustering

Motivation – Classifying Cancer Types

- We collected gene expression data for 1000 cancer patients, can you find the different classes of cancer in the data?

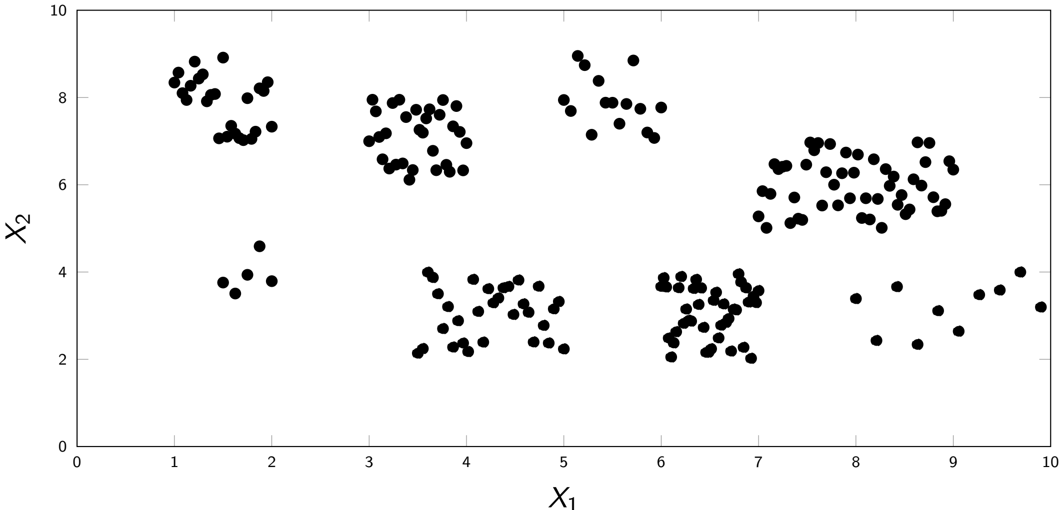


- We are not given the class labels y, but want meaningful labels
- An example of unsupervised learning

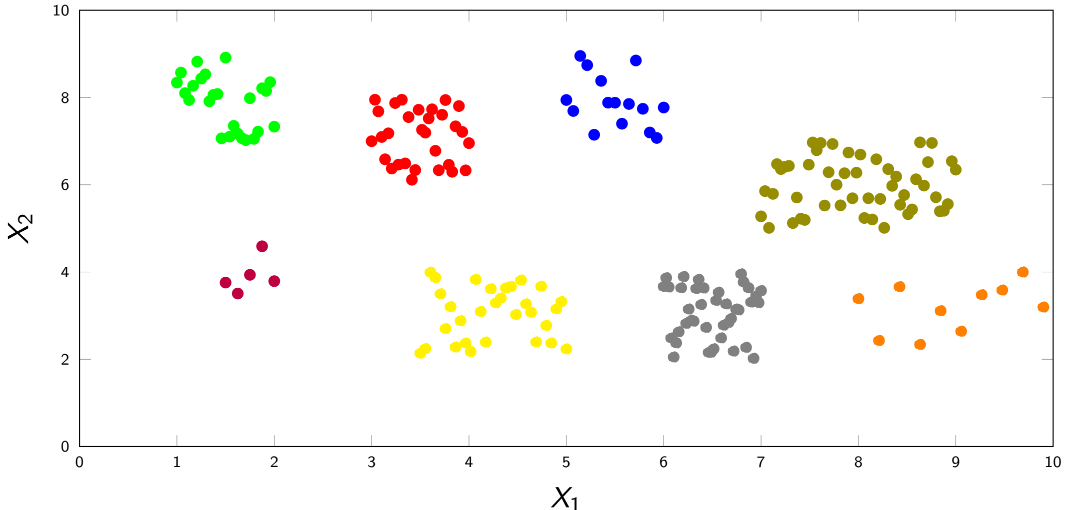
Clustering

- Input: set of instances described by d features
- Output: an assignment of instances to 'groups'
- Unlike classification, we are not given the 'groups'
  - Algorithm must discover groups
- Example of groups we might discover in e-mail spam:
  - 'Lucky winner' group
  - 'Weight loss' group
  - 'I need your help' group
  - 'Mail-order bride' group

Example



Example



What is Clustering?

- Cluster: A collection of data object
  - Similar (or related) to one another within the same group
  - Dissimilar (or unrelated) to the objects in other groups
- Clustering (aka cluster analysis, data segmentation, ...)
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters

What is Clustering?

- The **best** clustering is hard to define
  - We don't have a test error
  - Generally, there is **no best** method in unsupervised learning
    - So there are lots of methods: we will focus on important/representative ones.
- Typical applications
  - You could want to know **what the groups are**
  - You could want to find **the group for a new example** x_i
  - You could want to find **examples related to a new example** x_i
  - You could want a **prototype example for each group**

Applications – Data Understanding

- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults
- Climate: understanding earth climate, find patterns of atmospheric and ocean changes

Applications – Preprocessing

- Summarizing:
  - Preprocessing for regression, PCA, classification, and association analysis
- Compression:
  - Image processing: color quantization (computer graphics), i.e. task of reducing the color palette of an image to a fixed number of colors
- Outlier detection
  - Outliers are often viewed as those “far away” from any cluster

K-Means

The K-Means Algorithm

- Most popular clustering method
- Given number of clusters k (hyper-parameter), k-means is implemented in four steps:

  1. Initial guess of the centroid ("mean" or aka center) of each cluster
  2. Assign each instance to its closest cluster centroid (in terms of Euclidian distance)
  3. Update the cluster centroids based on the assignment in step 2
  4. Go back to step 2 and repeat until convergence

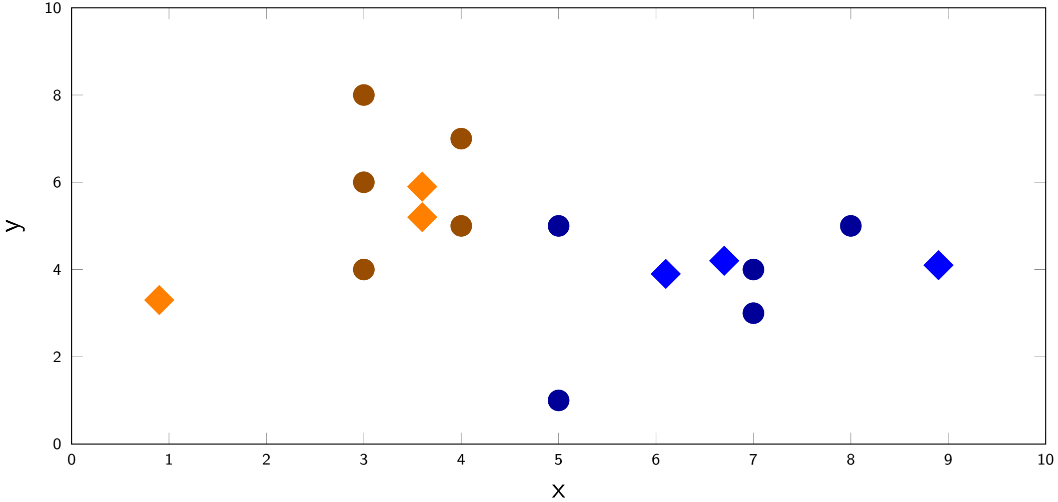
The K-Means Algorithm

Input: Data points $D = \{x_1, \ldots, x_n\}$, number of clusters k
Output: Partitioning of D into k mutually exclusive clusters $C = \{C_1, \ldots, C_k\}$
for $c = 1, \ldots, k$ **do**
  $w_c \leftarrow$ randomly chosen $x_i \in D$
end
while *changes in C happen* **do**
  //Assign instances to clusters based on Euclidian distance aka L2-norm:
  $dist(y, x) = \sqrt{\sum_{j=1}^{d} (y_j - x_j)^2} = ||y - x||_2$
  for $c = 1, \ldots, k$ **do**
    $C_c = \{x \in D | dist(w_c, x)^2 \leq dist(w_r, x)^2 \;\; \forall r = 1, \ldots, k, c \neq r\}$
  end
  //Update the cluster centers
  for $c = 1, \ldots, k$ **do**
    $w_c = \dfrac{\sum_{x \in C_c} x}{|C_c|}$
  end
end

Complexity

- k number of clusters
- n instances (each d-dimensional vector)
- I number of iterations
- Suggestions?
  - $O(nkdI)$
- Bottleneck: We need to compute distance from n instances to k clusters I times

K-Means – Example



Interactive Demo!

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

K-Means Issues

- Guaranteed to converge when using Euclidean distance
- Given a new test example
  - Assign it to the nearest (cluster) center to cluster it
- Assumes you **know number of clusters** k
  - Lots of heuristics to pick k, none satisfying
  - Cross-validation
- Each example is **assigned to one (and only one) cluster**
  - **No possibility for overlapping clusters** or leaving examples unassigned
- It may converge to **sub-optimal solution**

What is K-Means Doing?

- We can interpret K-means steps as minimizing an objective
  - Total sum of squared distances from each example x_i to its cluster center (i.e squared L2 norm)

$$f(w_1, \ldots w_k, \hat{y}_1, \ldots, \hat{y}_n) = \sum_{i=1}^{n} ||w_{\hat{y}_i} - x_i||_2^2$$

- The k-means steps:
  - Minimize f in terms of the $\hat{y}_i \in \{1, 2, ..., k\}$ (cluster assignments)
  - Minimize f in terms of the w_c (cluster centers)
- Termination of the algorithm follows because:
  - Each step does not increase the objective
  - There are a finite number of instance assignments to k clusters (i.e. k^n)

K-Medians Clustering

- With other distances k-means may not converge
  - But we can make it converge by changing the updates so that they are minimizing an alternative objective function
- E.g., we can use the L1-norm objective:

$$\sum_{i=1}^{n} ||w_{\hat{y}_i} - x_i||_1 = \sum_{i=1}^{n} \sum_{j=1}^{d} |w_{\hat{y}_i j} - x_{ij}|$$

- Minimizing the L1-norm objective gives the k-medians algorithm
  - Assign points to clusters by finding centers with smallest L1-norm distance
  - Update cluster centers as median value (dimension-wise) of each cluster (this minimizes the L1-norm distance to all the instances in the cluster)
- This approach is **more robust to outliers**

K-Medoids Clustering

- A disadvantage of k-means in some applications: **the cluster centers might not be valid data points.**

  E.g., consider document described by bag of words features like [0,0,1,1,0], that is words 3 and 4 appear in the document.
  - A cluster center from k-means might look like [0.1 0.3 0.8 0.2 0.3].
  - What does it mean to have 0.3 of word 2 in a document?
- Alternative to k-means is k-medoids:
  - Same algorithm as k-means, except the cluster centers must be data points in D.
  - Update the cluster center by finding instance in the cluster minimizing squared L2-norm distance to all points in the cluster.

Initialization

- K-means is **fast but sensitive to initialization**
- Classic approach to initialization: random restarts
  - Run to convergence using different random initializations
  - Choose the one that minimizes average squared distance of data to the cluster centers
- Newer approach: k-means++
  - Random initialization that prefers means that are far apart

K-Means++

- Steps of k-means++:
    1. Select initial cluster center w_1 as a random instance x_i in D
    2. Compute distance d_{ic} of each instance x_i to each cluster center w_c

$$d_{ic} = \sqrt{\sum_{j=1}^{d}(x_{ij} - w_{cj})^2} = ||x_i - w_c||_2$$

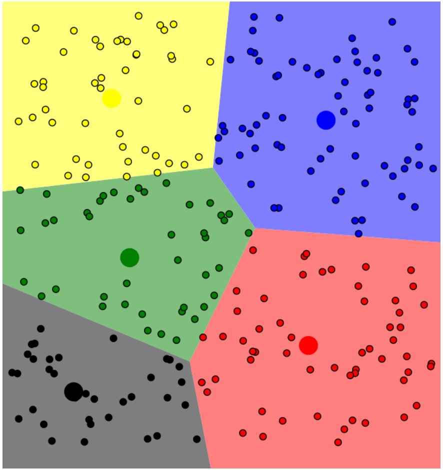
    3. For each instance x_i set d_i to the distance to the closest center $d_i = min_c\{d_{ic}\}$
    4. Choose the next cluster center by sampling an instance x_i proportional to $(d_i)^2$

$$p_i \propto d_i^2 \Rightarrow p_i = \frac{d_i^2}{\sum_{j=1}^{n} d_j^2}$$

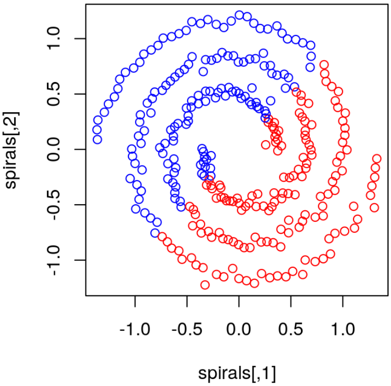
    5. Keep returning to step 2 until we have k cluster centers.
    6. Assign instances to clusters & update cluster centers until convergence

Shape of K-Means Clustering

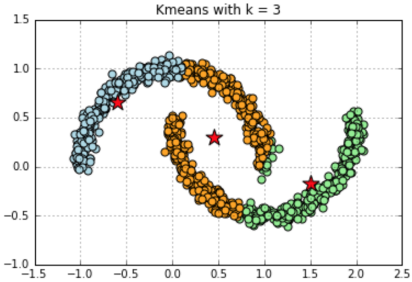
- K-Means partitions the space based on the closest mean
- Notice that the clusters are convex regions
  - A set is convex if any line between two points in the set stays in the set
- What are issues with that?
  - Clusters in the data might not be convex
  - How about outliers?



Non-convex data sets



Non-convex data sets



Partitioning Algorithms

- K-Means is a partitioning algorithm
- Partitioning a database D of n objects into a set of k clusters, such that within-cluster variation (the sum of squared distances of the object to the cluster centers) is minimized

$$E = \sum_{c=1}^{k} \sum_{x \in C_c} dist(x - w_c)^2,$$

  where w_c is the centroid or medoid of cluster C_c
- Given k, find a partition of k clusters that optimizes the chosen partitioning criterion
  - Global optimum: exhaustively enumerate all partitions
  - Local optimum: heuristics, such as k-means
- Suitable for detecting **similar-size non-overlapping clusters of spherical shape**
- There are other types of clustering algorithms, such as density-based and hierarchical clustering

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

SCIENCE
SCHOOL OF COMPUTER SCIENCE

Thank you for your attention!

https://ml.auckland.ac.nz