BIBA: Business Intelligence and Big Data

2018-10-10

Jens Ulrik Hansen

Clustering and Classification

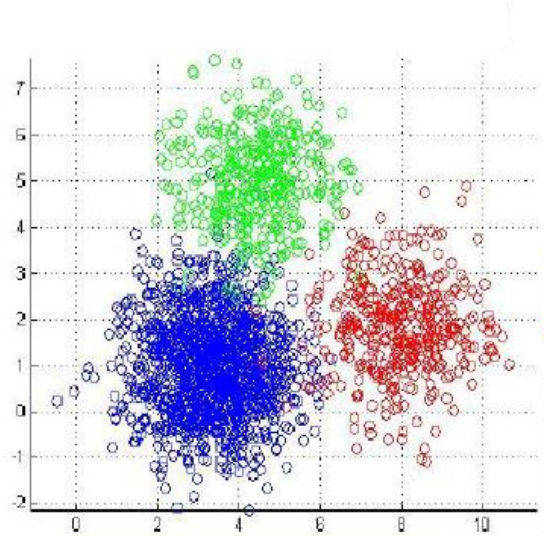
Today’s program

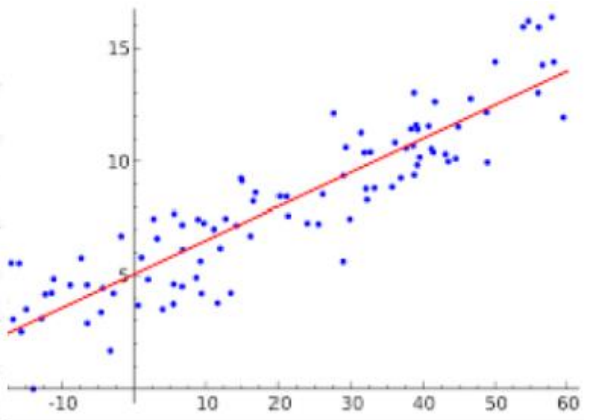
- Machine learning – supervised and unsupervised learning, training and testing
- Clustering – k-means and hierarchical clustering
- Classification – k-nearest neighbor and decision trees

Machine learning – supervised and unsupervised learning, training and testing

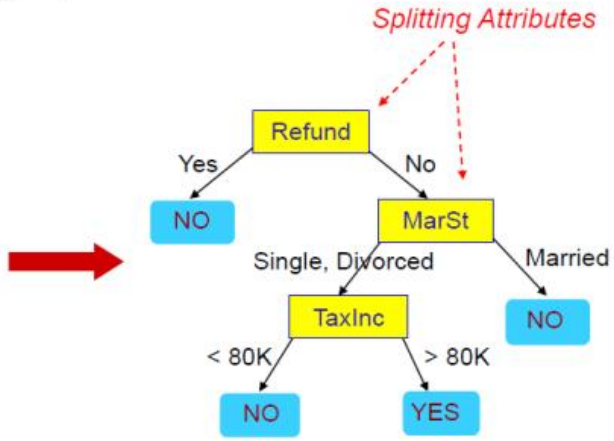
Types of machine learning

- ***Supervised learning*** – data contains values for what we want to predict
  - Regression (linear regression)
  - Classification (k nearest neighbor, decision tree)
- ***Unsupervised learning*** – data does not contain answers to what we want
  - Association Rule Mining (a prior algorithm)
  - Clustering (k-means clustering)
- ***Reinforcement learning***



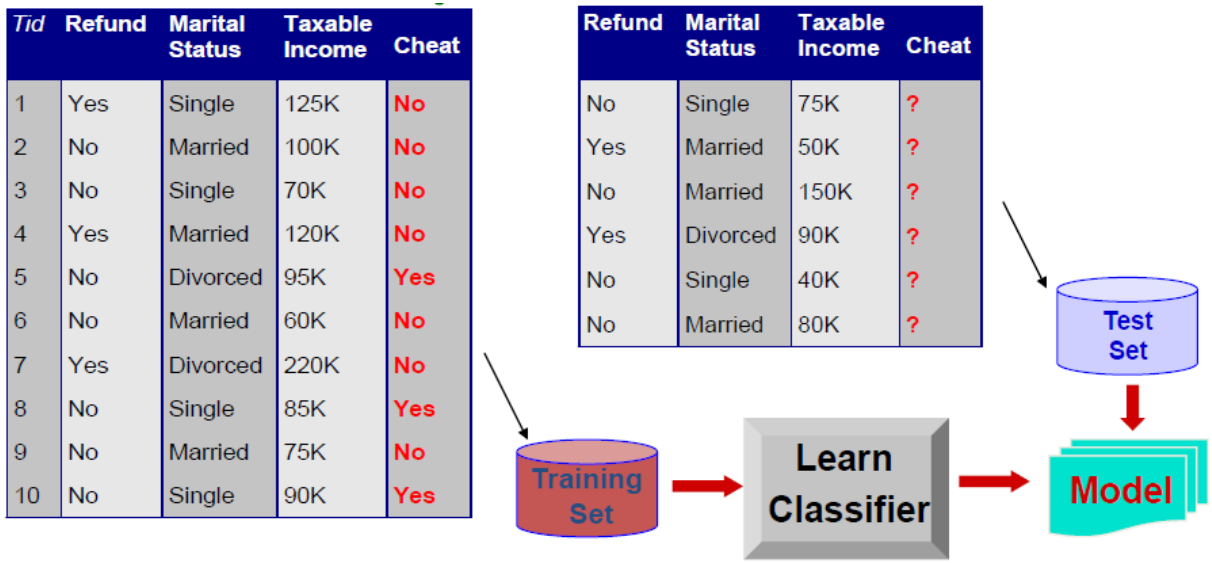


Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Evaluating machine learning algorithms

- To test machine learning models we usually split the data set up in a ***training set*** and a ***test set***
- A common split is 70% for the training set and 30% for the test set
  - Note, if we have a small amount of data, splitting it up into training and test sets will decrease the amount of valuable training data. In this case, more elaborate methods such as cross-validation can be used.

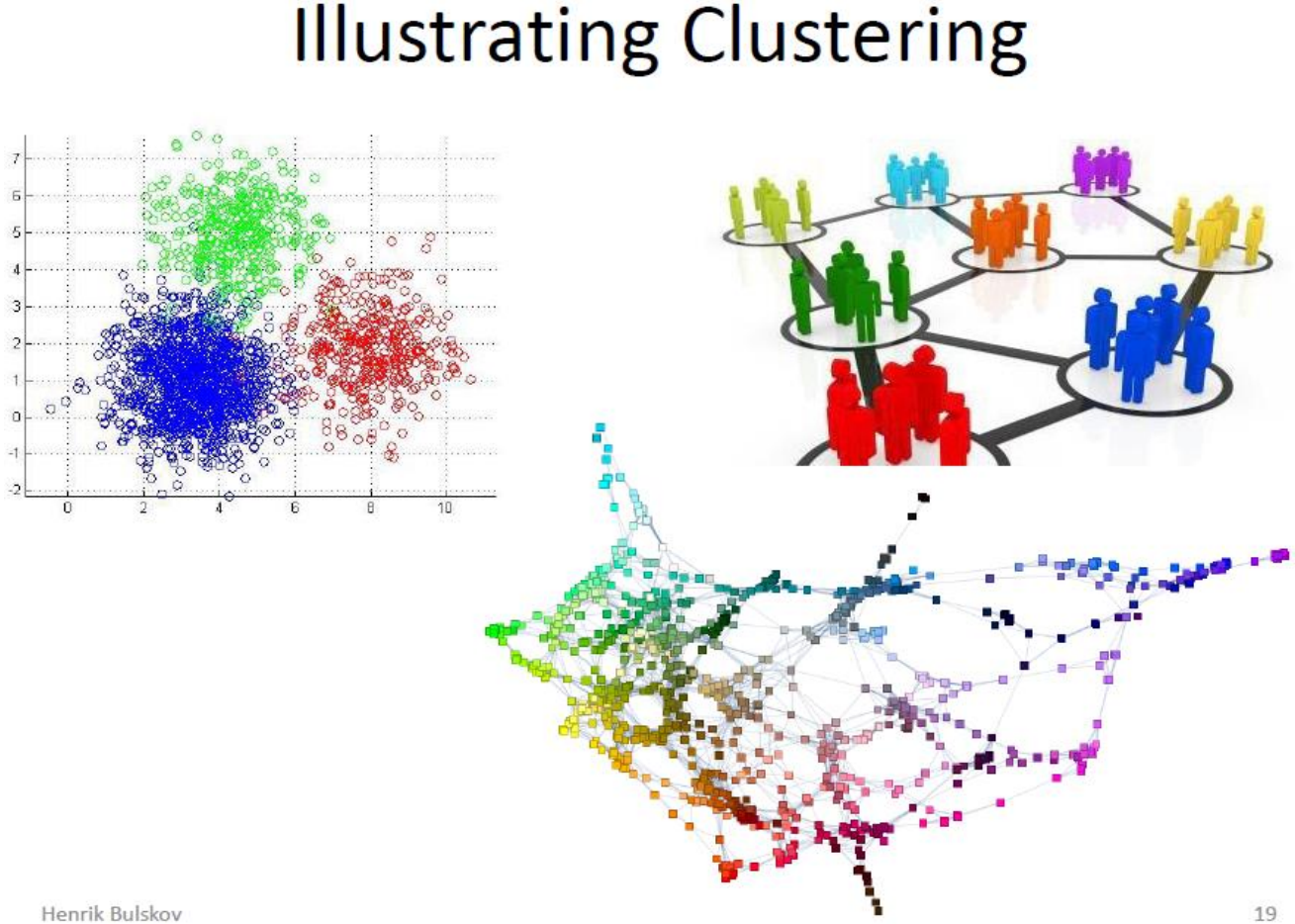


- See the notebook “5.1 Evaluating machine learning models.pynb”

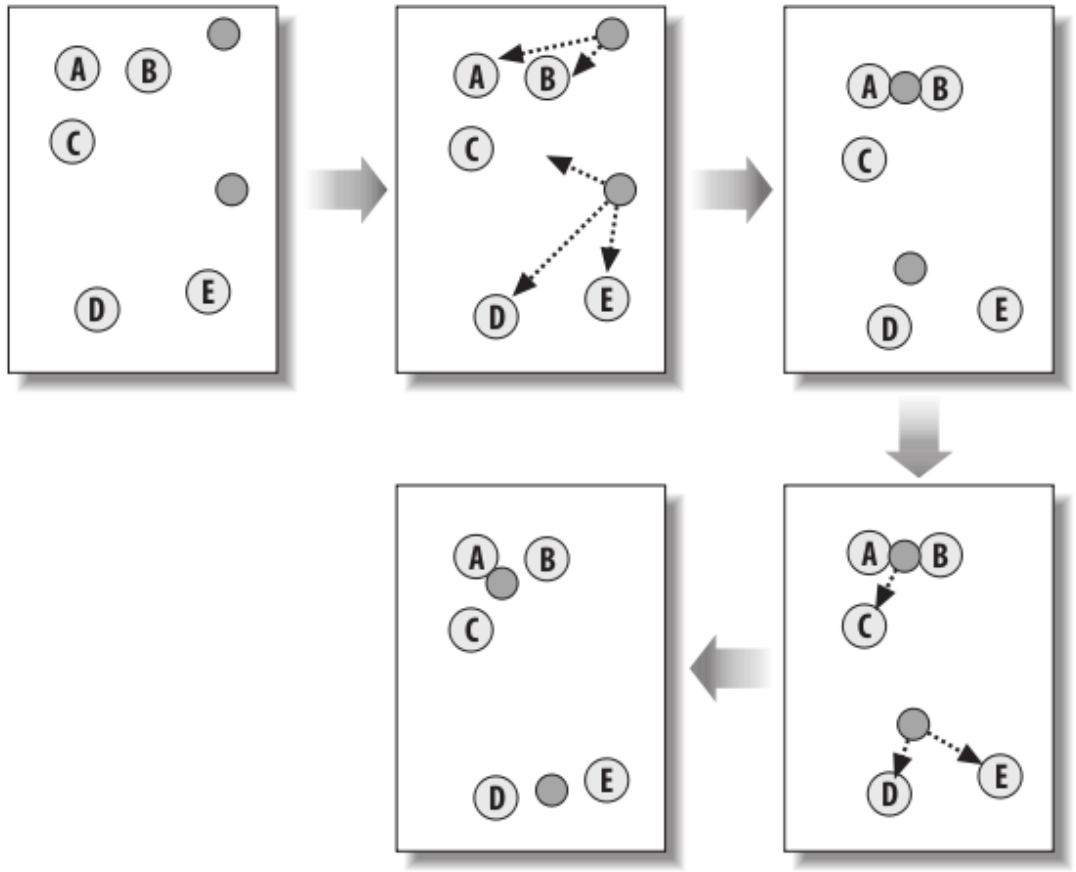
Clustering – k-means and hierarchical clustering

Clustering in general

- Cluster data points into groups or clusters
- An unsupervised learning task
  - We do not know which clusters there are or how many
- Applications
  - Market segmentation
  - Placing cellular towers or distribution centers



The K-means clustering algorithm



- The algorithm
  1. Specify the number of clusters (k)
  2. Randomly place k cluster centers in the feature space (or chose centers among the data points)
  3. Assign each data point to a cluster based on the center closest to it
  4. Recalculate the “center”/mean of each cluster
  5. Repeat 3 and 4 until no more points are moved between clusters
- Note that, there is no guarantee that the algorithm will always find the same clustering – a non-deterministic algorithm
- There is no guarantee that the algorithm will not get stuck in a local optimum
- “closest” means closest with respect to the Euclidean distance:

  $d((x1, y1), (x2, y2)) = \sqrt{((x1-x2)^2 + (y1-y2)^2)}$

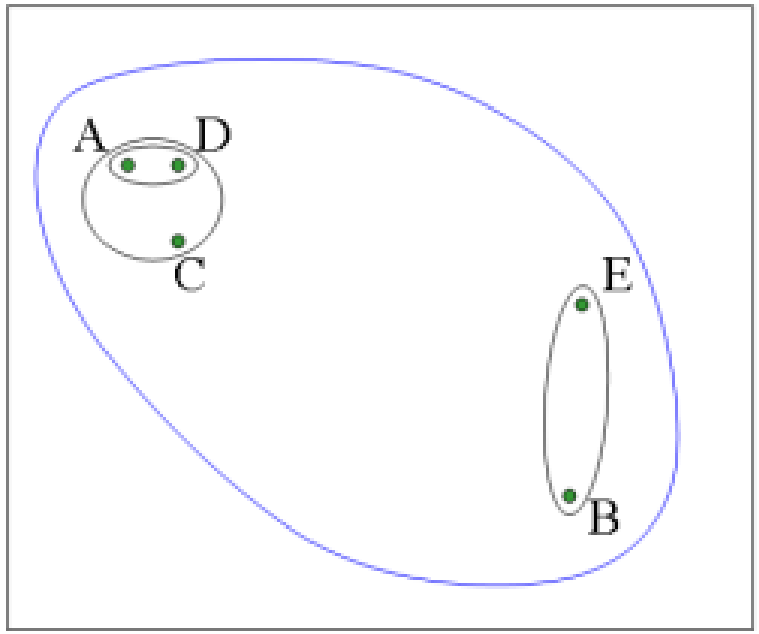
K-means clustering

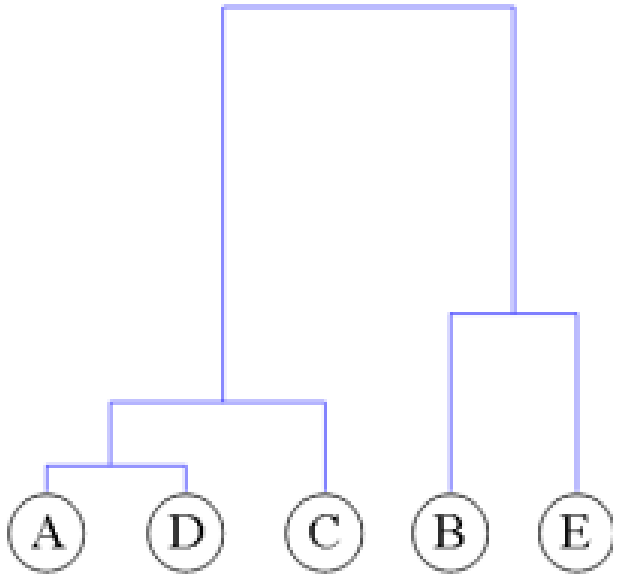
- Comments
  - You specify the number of clusters (k) you want from the beginning
  - Aims at reducing with cluster sum of squared error
    - This can always be achieved by having a cluster for each data point
  - Because of the use of Euclidian distance (or a distance measure in general), the algorithm is sensitive to the scale of the variables.
    - Thus, always normalize (scaling and centering) all the variables before using the k-means clustering

Hierarchical clustering

- We do not specify the number of clusters in advance, but we pick the number of clusters we want once the algorithm is done
- There two approaches to Hierarchical clustering:
  - ***Divisive clustering*** starts top-down with all the data points in a single cluster and keep splitting up into more and more clusters until each data point is in its own cluster
  - ***Agglomerative clustering*** stars bottom-up which each data point in a cluster and pairs clusters into fewer and fewer clusters until all data points are in one cluster
- In either case, you can pick a stage in the process and take the clusters created there as your output.

(Agglomerative) Hierarchical clustering algorithm





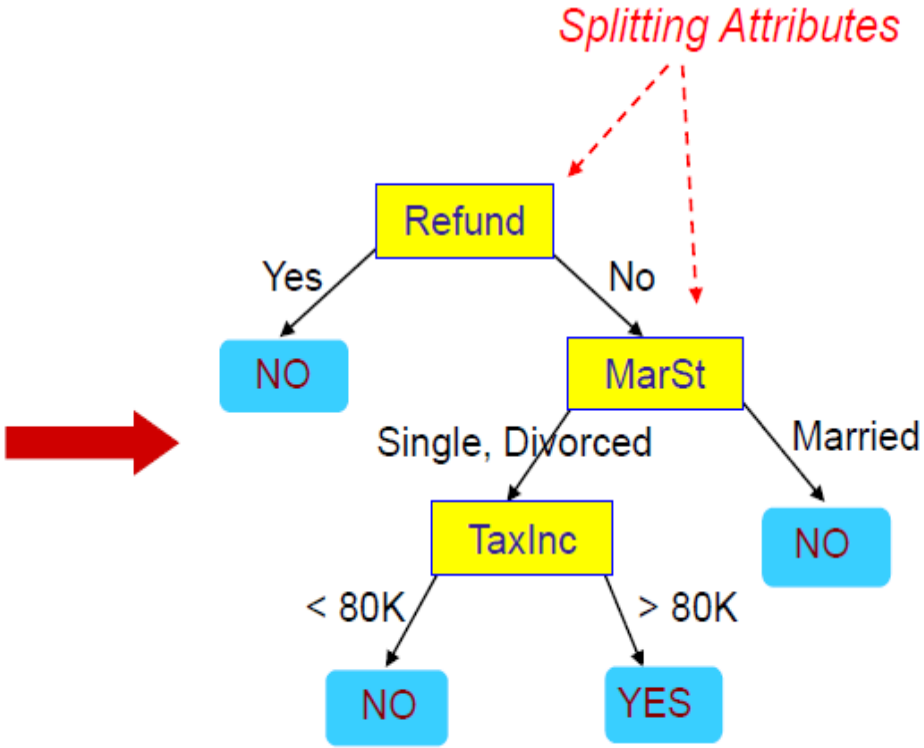
- Algorithm
  1. Start with each data point in its own cluster
  2. Take the two closest clusters and merge them into on cluster
  3. Recalculate the distance between clusters
  4. Repeat 2 and 3 until there is only one cluster left

Classification – k-nearest neighbor and decision trees

Classification in general

- Cluster data points into potential groups known in advance
- Applications
  - Classify emails as spam or not
  - Classify a credit card transaction as fraud or not
  - Classify a tumor as malignant or benign
  - Classifying images
- See the slides
  - “classification.html”

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Exercises

- Use the iris dataset in R and run both hierarchical and k-means clustering on it (excluding at least the “Species” column).
  - To make it easy to visualize, you can chose to use two columns only, for instance the “Petal.Length” and “Petal.Width” columns.
- The iris data set also contains a column called “Species”. Use this as the response variable in a classification task.
- Go to the UCI Machine Learning Repository and click “view all data sets”. In the left menu under “Default Task” select “Classification” or “Clustering” and do classification or clustering on one of the data sets.