Classification and clustering

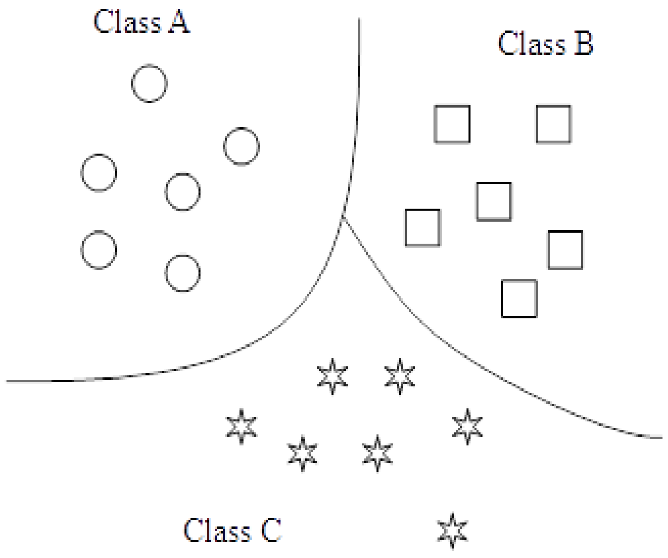
1 Classification

Classification – assignment of documents to predefined classes:

- building a model describing predefined classes,
- using built model for new documents

Applications

- recommendation systems,
- spam filtering,
- personalized news, advertisements
- …



1.1 Rocchio classifier

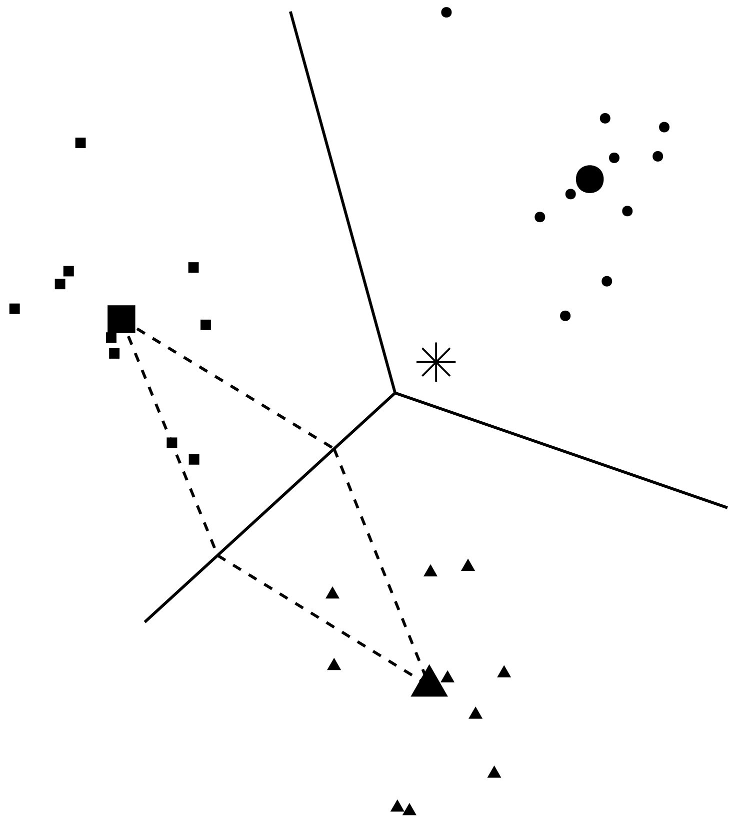
Roccion classifier determines the boundaries between classes based on the centroid of each class:

$$\mu_C = \frac{1}{|D_C|}\sum v\,(d)$$

Boundary between classes is defines as a hyperplane: the set of points in the equal distance from the centroids of the classes.

New document is assigned to the class with the centroid most similar to this document.

Example



In the picture above there are 3 classes (squares, circles and triangles). Big shapes determine the centroids of elements in these classes. Lines determine the border between classes (equal distance to the nearest class centroids). New (unclassified) element – star will be classified as circle (the nearest class centroid is circle).

1.2 Nearest neighbor algorithm

Nearest neighbor algorithm (k-nn): class of new example is determined by the class, which dominates for k training examples nearest to the new one.

Characteristics:

- **Non-parametric algorithm** – does not make assumptions about data distribution,
- **Lazy/instance-based classification** – no additional computation needed for model creation (no exact training phase).

The most often k = 3 or 5

Modification: weighted k-NN – weights of examples taken into consideration are given by the similarity (or distance) to this example.

1.3 Naïve Bayes algorithm

Naïve Bayes is a probabilistic algorithm based on Bayes theorem. It tries to answer question which class is the most probable class based on given training data.
We classify the document D to class C_i for which $P(C_i \mid D)$ is maximal.

$$P(C|D) = \frac{P(D|C) * P(C)}{P(D)}$$

P(C) – probability of class C estimated as ratio of training examples in class C to all training examples,
P(D) – probability of document, the same for all documents, so it does not influence the final class ranking.
How to calculate P(D|C)?
We assume the independence of attributes – in reality attributes are not independent, this is why the algorithm is called **Naïve** Bayes.
For independent attributes the following equation is correct:

$$P(D|C) = \prod_{j=1}^{n} P\left(t_j|C\right)$$

$P(t_j|C)$ – fraction of documents from class C, which contain term t_j.

If the document representation is binary we can use standard Naive Bayes approach. For representation taking into consideration the frequency of term the following modification (called add-one or Laplace smoothing) is applied for omitting situation where $P\left(t_j \vee C_i\right) = 0$:
$P\left(t_j|C\right) = \frac{T_{ct}+1}{\sum(T_{ct'}+1)} = \frac{T_{ct}+1}{(\sum T_{ct'})+|V|}$,
where:
T_{ct} is the number of occurrences of term t in documents from class C (including the repeating occurrences of this term in documents)
|V| is the number of terms in the dictionary.

Example:

Doc ID	Content	Decision (c = China?)
1	Chinese Beijing Chinese	Yes
2	Chinese Chinese Shanghai	Yes
3	Chinese Macao	Yes
4	Tokyo Japan Chinese	No
5	**Chinese Chinese Chinese Tokyo Japan**	**?**

$P(c) = \frac{3}{4}$ (3 out of 4 training documents are in class c = Yes)

$$P(\sim c) = \frac{1}{4}$$

In the new (unclassified) document where are 3 distinct terms: Chinese, Tokyo, Japan.

$P(Chinese \mid c) = \frac{5+1}{8+6} = \frac{3}{7}$(there are 5 occurrences of word "Chinese" in documents with **c = Yes**, 8 terms at all in these documents and 6 words in the whole dictionary)

$P(Chinese \vee\sim c) = \frac{1+1}{3+6} = \frac{2}{9}$(there is 1 occurrence of word "Chinese" in document with **c = No**, 3 terms at all in this document and 6 words in the whole dictionary)

$$P(Tokyo \mid c) = \frac{(0+1)}{(8+6)} = \frac{1}{14}$$
$$P(Japan \mid c) = \frac{(0+1)}{(8+6)} = \frac{1}{14}$$
$$P(c \mid d_5) = \frac{3}{4} * \left(\frac{3}{7}\right)^3 * \frac{1}{14} * \frac{1}{14} \approx 0.0003$$

$$P(Tokyo \vee\sim c) = \frac{(1+1)}{(3+6)} = \frac{2}{9}$$
$$P(Japan \mid \sim c) = \frac{(1+1)}{(3+6)} = \frac{2}{9}$$
$$P(\sim c \mid d_5) = \frac{1}{4} * \left(\frac{2}{9}\right)^3 * \frac{2}{9} * \frac{2}{9} \approx 0.0001$$

Decision: c=Yes

1.4 Collaborative filtering – Social learning

Recommendation of not rated objects based on historical ratings of other users based on similarities between users or items.

User-based collaborative filtering
- Calculate similarity between current user and all other users.
- Use k-NN algorithm to find the most similar users.
- Calculate the predicted rate for given movie.

Item-based collaborative filtering
- Find similarities between items based on user rates.
- Prediction of unknown rate based on the historical rates of given user for the similar items.

Pearson correlation
Correlation between user/item U and J:

$$r_{UJ} = \frac{\sum(U-\bar{U}) * (J-\bar{J})}{\sqrt{\sum(U-\bar{U})^2 * \sum(J-\bar{J})^2}}$$

Where $\bar{U}$ and $\bar{J}$ are the average rates of users U and J. Value 1 means the total similarity, -1 the opposite preferences.
It is good for rating at the scale at least 1-5.

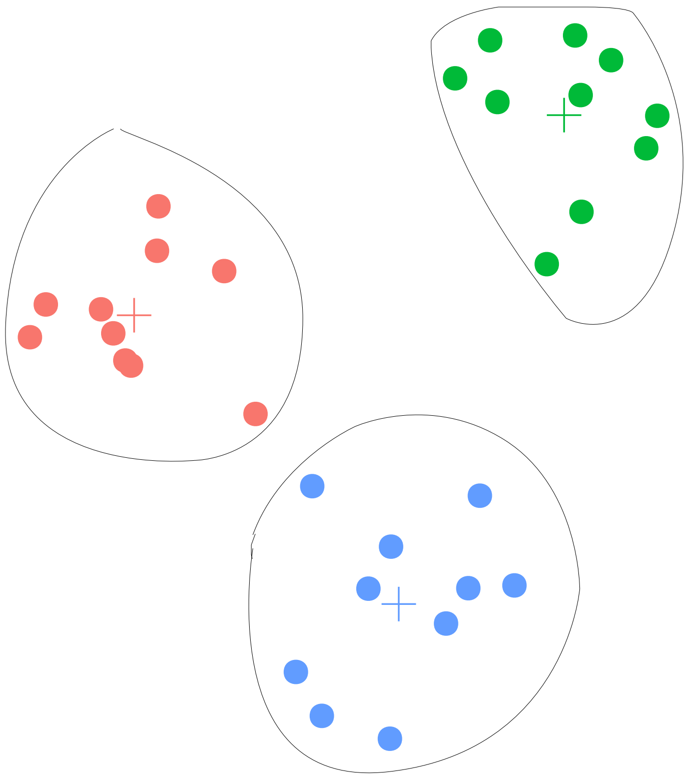
Predicted rate computing
In the simplest approach the predicted rate is the average of the rates of k nearest neighbors. In reality we use the weighted average (weights – similarity values) of neighbors' rates or weighted average of deviations from the average rate of nearest neighbors.

$$r(a,i) = \frac{\sum_{u=1}^{k} r_{u,i} * sim(a,u)}{\sum_{u=1}^{k} sim(a,u)}$$

$$r(a,i) = \bar{r_a} + \frac{\sum_{u=1}^{k}(r_{u,i} - \bar{r_u}) * sim(a,u)}{\sum_{u=1}^{k} sim(a,u)}$$

2 Clustering

Clustering is the natural grouping of objects with similar features into clusters. Each cluster should differ significantly from other clusters.
Cluster – set of objects which are "similar" (ex. group of documents which contain information about the same topic, group of users with similar interests or similar behavior).
Hierarchical methods – Generate nested sequences of division of documents during the clustering process.
Iterative optimization methods - generate only one division of sets of documents in every moment of the clustering process.



Applications:

- For documents searching – adding to search results other documents from cluster for better recall measure,
- grouping search results for displaying them in better organized form,
- discovering profiles of users, behavior or content.

2.1 K-means

Number of groups is known from the beginning of the algorithm (k).
Reallocation method – start with some start division and modify it with iterational method to achieve better division.

k-means:

1. Choose k documents which would be centroids of groups (usually at random)
2. Assign other documents to group with the nearest centroid
3. For each of created groups compute new centroid
4. Return to step 2 until the assignments change or the iteration limit is achieved.

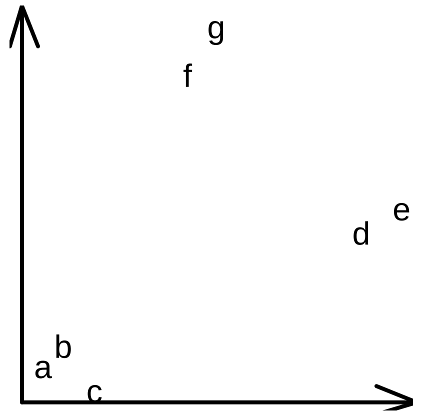
Remarks:

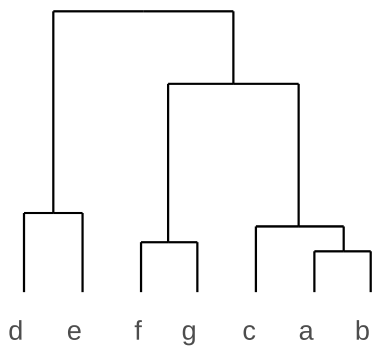
- Relatively fast – O(tkn), where t – number of iterations, n – number of objects, k – number of clusters
- Often ends in the local optimal division
- The result is strongly dependent on the starting division
- Number of clusters must be set at the beginning of the process
- Sensitive to noise and outliers in input data
- It can be used only if mean of documents can be computed

As a modification centroids can be replaced with **medoids** – the most central points in clusters.

2.2 Hierarchical agglomerative clustering

- At the beginning every object is the different group
- In every step 2 closest (the most similar) objects (or groups) are merged
- Algorithm is stopped if the given number of groups is achieved or the similarity between objects (groups) is lower than the given threshold.





Distance between groups of documents **S1** and **S2:**

- Distance between centroids of groups
- Maximal similarity between documents in groups (nearest neighbors clustering)
- Minimal similarity between documents in groups (furthest neighbors clustering)
- Average similarity between pairs of documents

3 Exercices

3.1 K-Nearest Neighbors

Given is collection of 10 documents with assigned classes A or B and their similarity to the new document X with unknown class. Similarity of documents D9 and D10 to X are also unknown. Using data in tables below find the missing Jaccard similarities of determine to which class X should be assigned according to 3-NN simple and weighted algorithms.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	X
Class	B	A	B	B	A	B	A	A	A	B	?
Similarity to X	0.35	0.22	0.89	0.17	0.42	0.09	0.28	0.27	?	?	

	X	D9	D10
Number of terms in document	10	14	9
Number of terms common with X		7	4

Similarity D9 to X = $\frac{1}{2}$ = 0,5
Similarity D10 to X = $\frac{4}{9}$ = 0,44
Documents which are used to determine class for X:

Decision 3-NN (simple algorithm): B

Decision 3-NN (weighted algorithm): B

3.2 Naive Bayes classifier

Given is the binary representation of 10 documents (D1-D10). To which class (A or B) should be assigned document Y, if we use Naive Bayes as classifier. If any of partial probabilities is equal to 0, then assume it is equal to 0.01.

	T1	T2	T3	C
D1	1	1	1	B
D2	0	1	0	B
D3	1	0	0	B
D4	0	1	0	B
D5	0	0	0	A
D6	0	1	1	A

	T1	T2	T3	C
D7	1	0	1	A
D8	0	1	0	A
D9	1	0	1	A
D10	0	0	1	A
Y	1	1	0	?

For class A:
P(A) = $\frac{6}{10} = \frac{3}{5}$
P(T1 | A) = $\frac{2}{15}$
P(T2 | A) = $\frac{2}{15}$
P(T3 | A) = $\frac{4}{15}$
P(A | Y) = 0,025

For class B:
P(B) = $\frac{2}{5}$
P(T1 | B) = $\frac{1}{2}$
P(T2 | B) = $\frac{3}{4}$
P(P3 | B) = $\frac{1}{4}$
P(B | Y) = 0,00041

A>B

3.3 Collaborative filtering

	Star Wars	Jur. Park	Terminator	Ind. Day
Sally	7	6	3	7
Bob	7	4	4	6
Chris	3	7	7	2
Lynn	4	4	6	2
Karen	7	4	3	?

Avg	Pearson
5,75	? 0
5.00	0.97
5.67	-0.97
4.67	-0.69
4	1.0

User-based collaborative filtering.
Given historical rates for 4 movies of 4 users, and Karen's rate of 3 movies (she hasn't seen Independence Day yes). Determine if we should recommend this movie to her. Pearson correlation values of Bob, Chris and Lynn to Karen is given in the table above. Taking into consideration k-NN with k =1,2 predict the Karen's rank of Independence Day.

K = 1: 4
K = 2: 2 | 5,5

Pearson(Sally, Karen) = 0

3.4 K-Means

Given is TF-IDF document representation and documents similarity matrix. Show the first iteration of 2-means algorithm. As starting centroids use documents with the **least similarity**. Find centroids for the second iteration of algorithm.

TF-IDF matrix:

	D1	D2	D3	D4	D5
T1	0.2	0.4	0.8	0.0	0.2
T2	0.0	0.4	0.2	0.8	0.2
T3	0.8	0.0	0.2	0.4	0.4

Similarity matrix:

	D1	D2	D3	D4	D5
D1	1.0	0.17	0.46	0.40	0.89
D2	0.17	1.0	0.83	0.60	0.58
D3	0.46	0.83	1.0	**0.30**	0.67
D4	0.40	0.60	**0.30**	1.0	0.73
D5	0.89	0.58	0.67	0.73	1.0

Step I:

Group I: D1, D5

Group II: D2, D3, D4

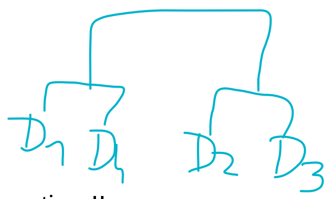
New centroids:

	T1	T2	T3
Group I	0,2	0, 1	0, 6
Group II	0,4	0,46	0,2

3.5 Hierarchical clustering

Show clustering process for documents D1-D4 using hierarchical agglomerative clustering. As similarity measure use the **minimal similarity** between documents in one group. In each step show matrix of similarities between documents (clusters). Name clusters with indexes of documents in it (eg. D12). Draw dendrogram.

	D1	D2	D3	D4
D1	1.0	0.8	0.6	0.5
D2	0.8	1.0	0.4	0.7
D3	0.6	0.4	1.0	0.6
D4	0.5	0.7	0.6	1.0



Iteration I:

	D2	D3	D4
D2	1.0	0,4	0,7
D3	0,4	1,0	0,6
D14	0,7	0,6	1,0

Iteration II:

	D14	D23
D14	1,0	0,9
D3	0,9	1,0

In iteration III all documents were merged into one group.

Dendrogram:

D1	D2	D3	D4

Programming assignment

Download files **Lab12 – MachineLearning.ipynb** and **movies.csv** from here. Follow instructions in the file.