High-level structure in texts as sets of words - I

Anton Alekseev,
Steklov Mathematical Institute in St Petersburg

NRU ITMO, St Petersburg, 2019

anton.m.alexeyev+itmo@gmail.com

Motivation

Suppose we have a large amount of unannotated texts,
based on which we are to, e.g.

- create a useful content-based recommendation service
  (“you love reading on battle rap, want some more articles on the same topic?”)

- make conclusions on topics represented in textual data --
  or other data structure features
  (annals analysis, blogosphere trends, etc.)

- find duplicate content
  (“the same piece of news, can be skipped”, “plagiarism!”)

And so on

Plan

1. Clustering
    a. The task
    b. Clustering quality evaluation
    c. Clustering methods types
        i. Representative-based
        ii. Probabilistic
        iii. Hierarchical
        iv. Density-based
    d. Tools & data

*2. *Finding Similar Items*
3. Topic modeling

Clustering

An unsupervised learning problem: split document collection into **groups (clusters)** so that the documents **in one group were similar** to each other, whereas documents f**rom different groups should be dissimilar**

Apart from the applications we've discussed –

- document summarization,
- cluster ID as a feature for classification/regression task,
- …?

Clustering: the task formulation

(one of the possible ones)

Given

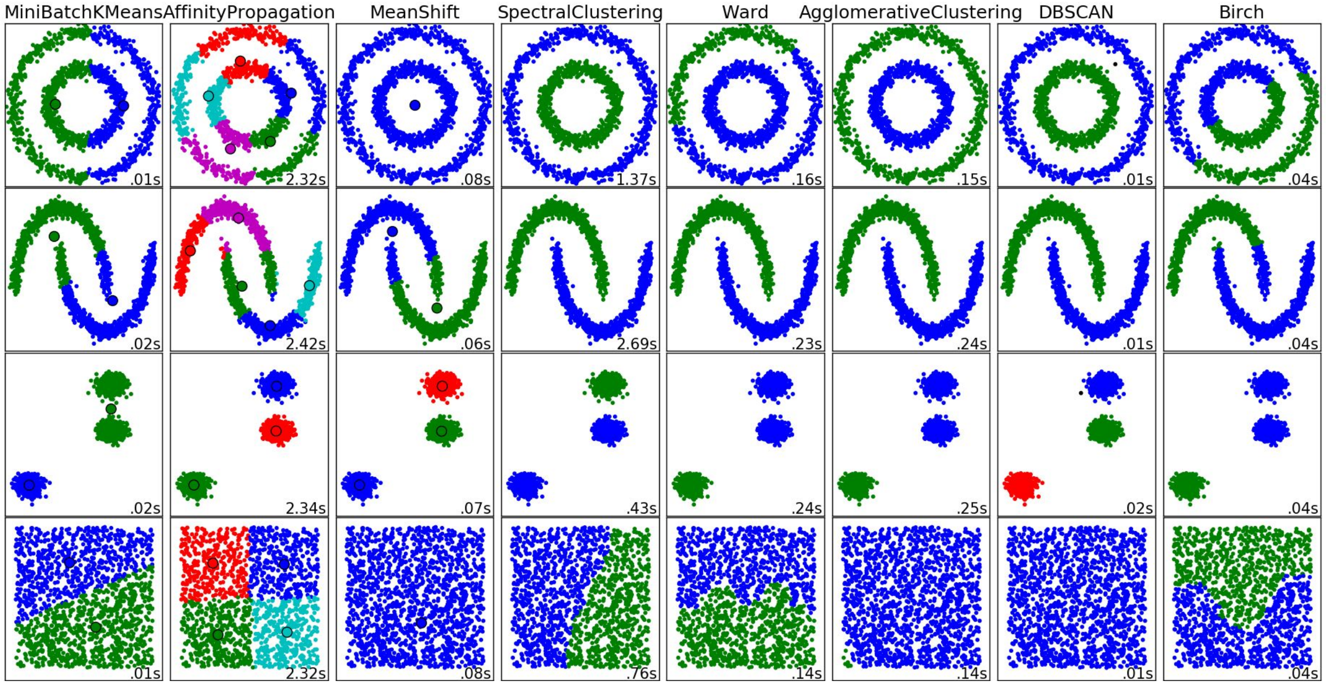
- **D** documents
- a way to measure distances between any pair of documents
- “understanding”, what good clustering is (quality functional)
- number of clusters **k** (optional!)

Do

Build a function **cl: D -> 1..k**, that matches each document with a cluster

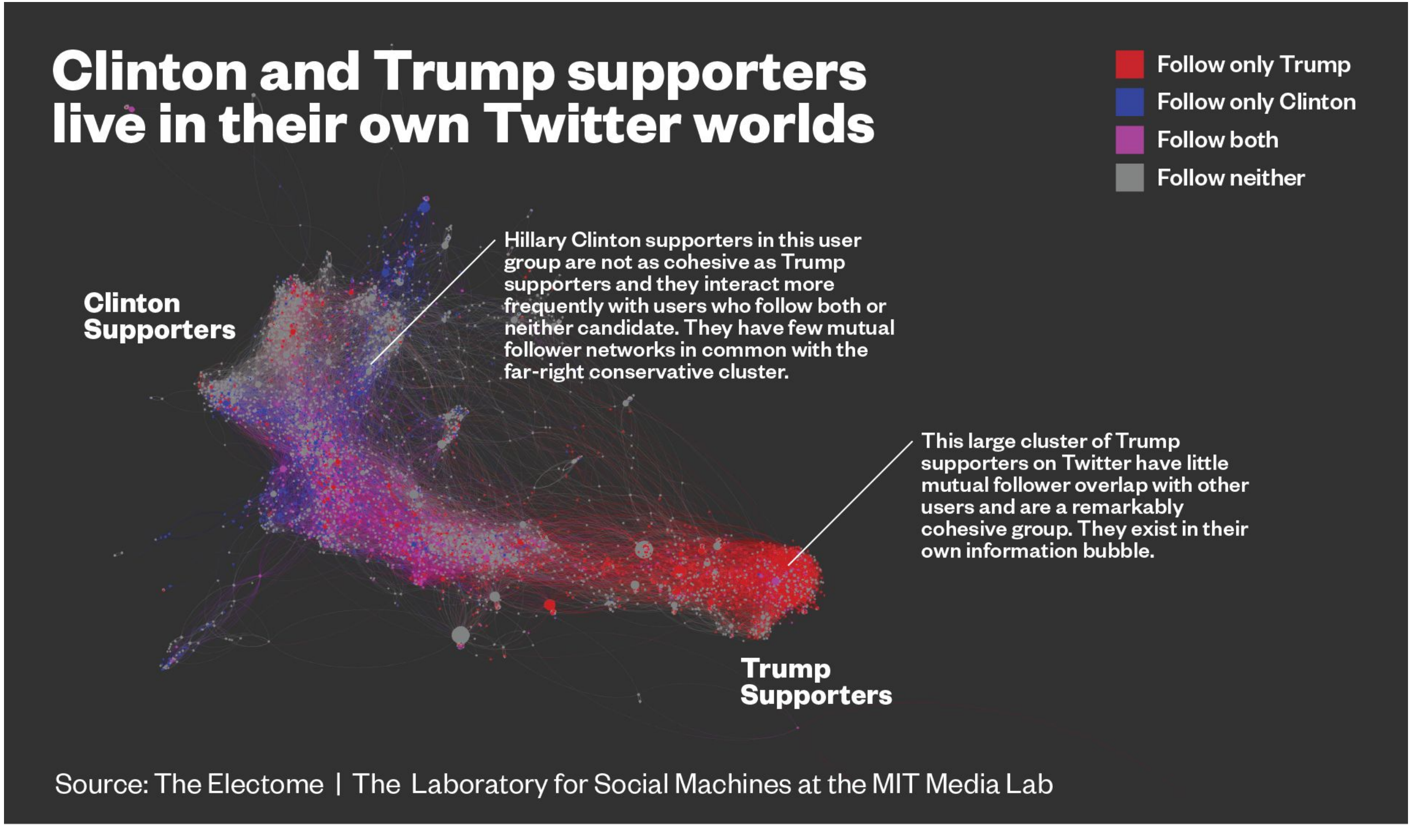
We already know how to represent text as a vector; all methods we will discuss are of course applicable in other domains and for other data types

Example: different methods work for different shapes of clusters



http://scikit-learn.org/0.18/_images/sphx_glr_plot_cluster_comparison_001.png

Example: non-Euclidean case

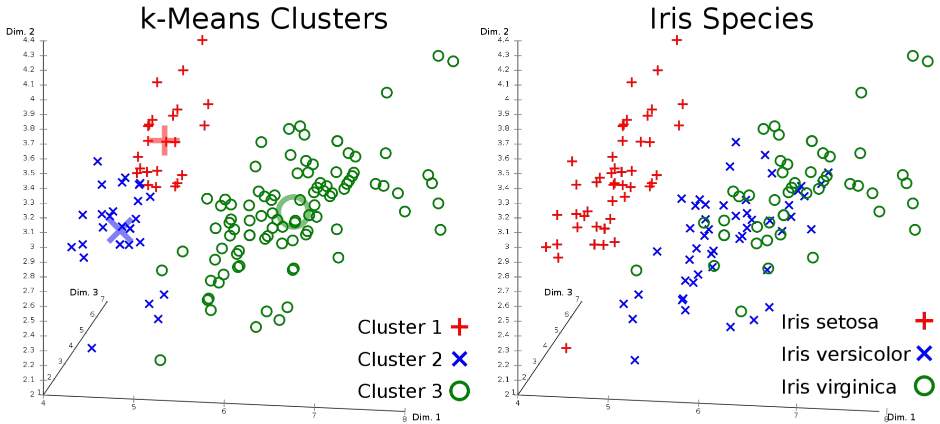


https://news2-images.vice.com//uploads/2016/12/TwitterData1-01.png

Clustering isn't that simple

- When we have two/three dimensions and a small dataset, things may be simple, however large number of dimensions is a different story (e.g. see Curse of dimensionality:Nearest neighbours)

- Quality evaluation: expensive or hard (= expensive)



Plan

1. ~~Clustering~~
    a. ~~The task~~
    b. Clustering quality evaluation
    c. Clustering methods types
        i. Representative-based
        ii. Probabilistic
        iii. Hierarchical
        iv. Density-based
    d. Tools & data

2. **Finding Similar Items*
3. *Topic modeling*

Clustering quality evaluation

As hard as clustering itself :(

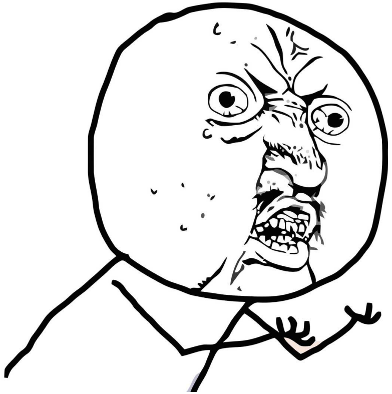
Ideas:

1. annotate and check **by hand**
2. apply to an already **annotated** dataset
3. extrinsic evaluation: estimate the ‘usefulness’ increase for some application
4. intrinsic evaluation: estimate some clustering ‘quality index’

Each of it is **ugly** in its own way!

Clustering quality evaluation

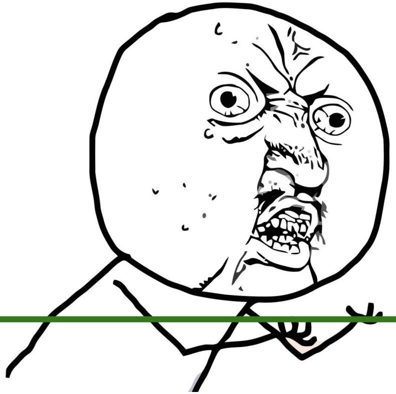
As hard as clustering itself :(
Ideas:



1. annotate and check **by hand**
   - doesn't scale
2. apply to an already **annotated** dataset
   - if we have the markup for training, why would we cluster the data?
3. extrinsic evaluation: estimate the 'usefulness' increase for some application
   - but this way we don't look at **clusters** quality
4. intrinsic evaluation: estimate some clustering 'quality index'
   - we look at one index when we optimize, then we look at a 'better' one… why not use the better one for optimization?

Clustering quality evaluation

As hard as clustering itself :(
Ideas:



1. annotate and check **by hand**
   - doesn't scale
2. apply to an already **annotated** dataset
   - if we have the markup for training, why would we cluster the data?
3. extrinsic evaluation: estimate the 'usefulness' increase for some application
   - but this way we don't look at **clusters** quality
4. intrinsic evaluation: estimate some clustering 'quality index'
   - we look at one index when we optimize, then we look at a 'better' one… why not use the better one for optimization?

Clustering quality evaluation

Suppose we have a test set where each object is matched with some cluster

Evaluation, way 1:

Annotate each pair of objects in the test set with

1 if they are in the same cluster or

0 if they are in different ones;

Then we do the same with our predictions

Thus we can evaluate quality the same way as we can do with classification:

1) we can compute **Accuracy**
   (how many pairs are correctly/incorrectly put into the same cluster)
2) or we can compute **Precision, Recall, F-measure**

Clustering quality evaluation

Evaluation, way 2: purity

‘How pure is each cluster’: max share of some true cluster in each of the predicted ones

$$\frac{1}{N}\sum_{m\in M}\max_{d\in D}|m\cap d|$$

D -- ‘true’ clusters

M -- predicted clusters

Manning, Christopher D.; Raghavan, Prabhakar; Schütze, Hinrich. Introduction to Information Retrieval. Cambridge University Press.

Clustering quality evaluation: problems

Pairs:

n(n-1)/2 pairs **is a lot**, the size of dataset (n) can't be large due to that

Purity:

large number of clusters delivers large purity value!

if every element is a cluster, purity = 1.0

A lot of clustering evaluation indices were invented,
each of them is ugly in its own way :)

For a good start one may take a look at the [Wikipedia article]

Plan

1. ~~Clustering~~
    a. ~~The task~~
    b. ~~Clustering quality evaluation~~
    c. Clustering methods types
        i. Representative-based
        ii. Probabilistic
        iii. Hierarchical
        iv. Density-based
    d. Tools & data

2. **Finding Similar Items*
3. *Topic modeling*

Clustering methods types

We can compare clustering algorithms in terms of:

- computational complexity
- do they build flat or hierarchical clustering?
- can the shape of clustering be arbitrary?
  - if not is it symmetrical, can clusters be of different size?
- can clusters vary in density of contained objects?
- robustness to outliers

http://www.machinelearning.ru/wiki/images/e/ea/13-MMP-Text_mining-Clustering.pdf

Clustering algorithms

1. Representative-based clustering
2. Probabilistic clustering
3. Hierarchical clustering
4. Density-based clistering

NB! The algorithms we are going to discuss have numerous modifications and implementations can differ greatly. Take care when carrying out experiments and training models for production environment!

Plan

1. ~~Clustering~~
    a. ~~The task~~
    b. ~~Clustering quality evaluation~~
    c. ~~Clustering methods types~~
        i. Representative-based
        ii. Probabilistic
        iii. Hierarchical
        iv. Density-based
    d. Tools & data

2. **Finding Similar Items*
3. *Topic modeling*

Representative-based: K-means

Applicability: vectors

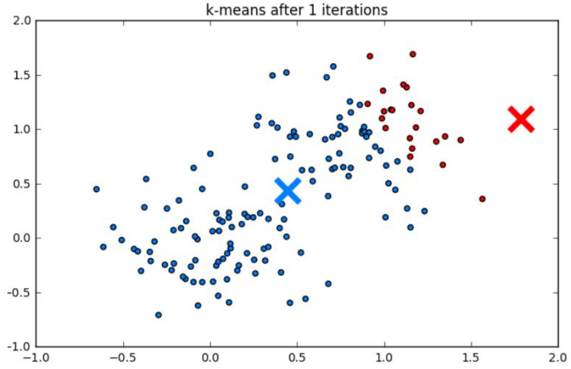
Goal: minimize the sum of squares of distances from centroid of each cluster to each element of the cluster

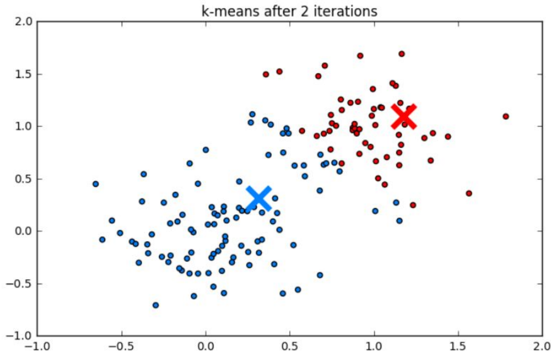
$$RSS = \sum_{k=1}^{K} \sum_{\vec{d} \in \omega} \left\| \vec{d} - \vec{\mu}(\omega) \right\|^2$$

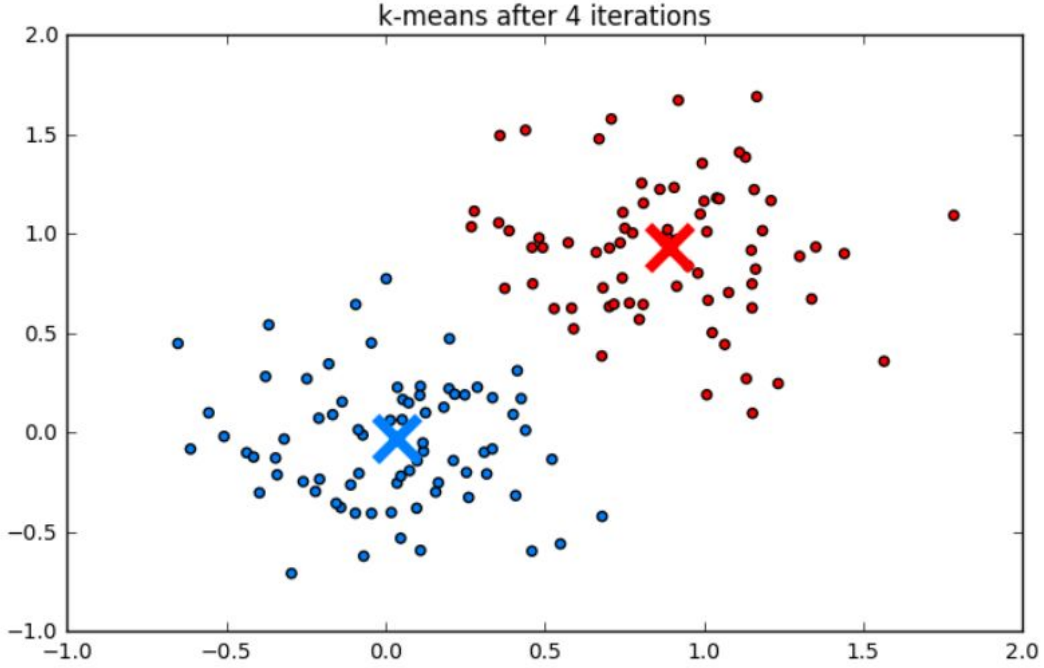
1. Set the number of clusters K.
2. Choose K documents at random -- clusters centroids.
3. Include the remaining documents into the closest cluster.
4. Compute new cluster **centroids** as a mean vector in the cluster.
5. Repeat steps 3-4, until
   a. Centroids stop to change?
   b. The partition of the dataset stops to change?
   c. We're tired? (limited number of iterations)

Steinhaus, H. (1957). "Sur la division des corps matériels en parties". Bull. Acad. Polon. Sci. (in French). 4 (12): 801–804
MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. 1. University of California Press. pp. 281–297.a

KMeans: what it looks like



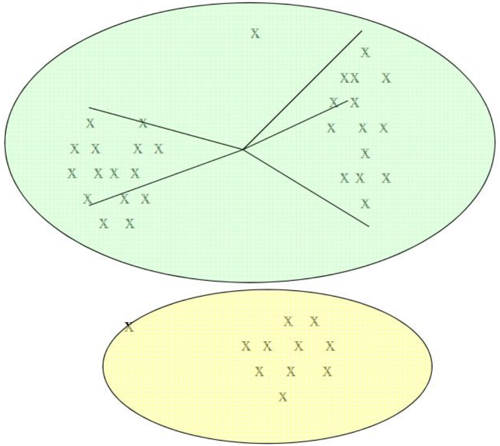




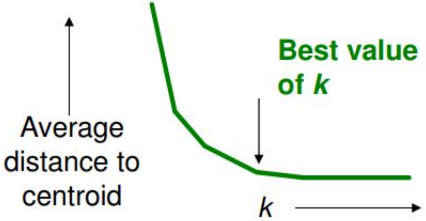
How to choose K?

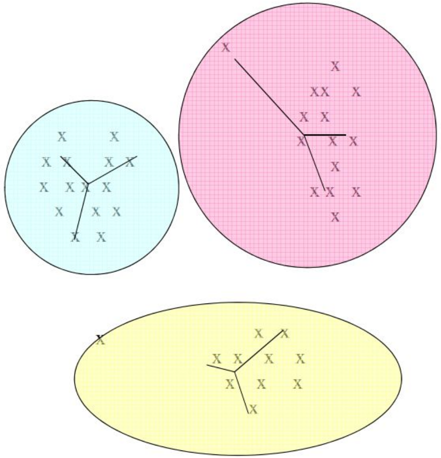
1. Gradually increase *K*
2. Look at the average distance to centroid

At some value of *K* it will stop to drop fast; this is the recommended *K* value



NOT BRO





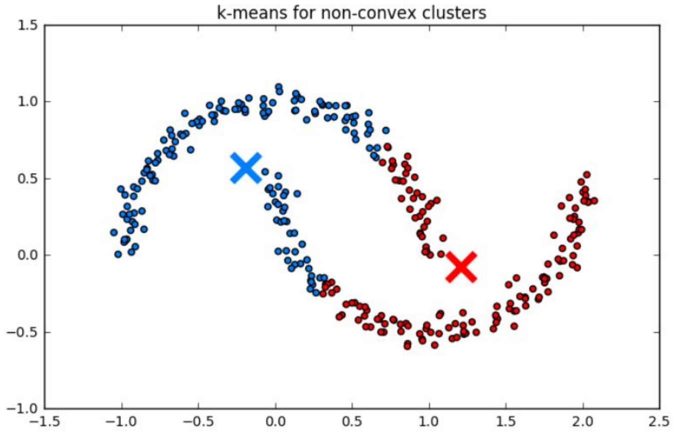
BRO

KMeans: discussion

Approximate solution of an NP-hard problem

Restrictions:

- can’t apply to the domain where there is no such thing as an ‘average object’
- is prone to ball-like clusters detection
- **always** finds K clusters
- sometimes heavily depends on initial centroids candidates choice



However, there are quite a few modifications useful in real life

Plan

1. ~~Clustering~~
    a. ~~The task~~
    b. ~~Clustering quality evaluation~~
    c. ~~Clustering methods types~~
        i. ~~Representative-based~~
        ii. Probabilistic
        iii. Hierarchical
        iv. Density-based
    d. Tools & data

2. **Finding Similar Items*
3. *Topic modeling*

Probabilistic: EM

In KMeans we had two stages

1. “estimation of the expectation” - mean vector in cluster computation
2. “re-assignment” - choosing which cluster should every point belong to

KMean is a special case of the **Expectation-Maximization** approach

- K-means is EM-algorithm when:
  - applied to Gaussians
  - with equal priors
  - with unity covariance matrices
  - with hard clustering

http://www.machinelearning.ru/wiki/images/e/ea/13-MMP-Text_mining-Clustering.pdf

What is Expectation-Maximization?

Iterative approach to estimation the parameters of the probabilistic models depending on latent variables.

Each iteration:

- **E-step (expectation)**: expected value of the likelihood function is computed, latent variables are not modified.
- **M-step (maximization)**: maximum likelihood estimates are computed, which are then used at the next E-step.

Steps are repeated until convergence

Clustering with EM-algorithm

We want to tune latent variables (~**centroids in KMeans!**) so that the probability of **D** generation was maximal

$$\Theta = \text{argmax}_\Theta\, L(D|\Theta) = \text{argmax}_\Theta \log \prod_{n=1}^{N} P(d_n|\Theta) = \text{argmax}_\Theta \sum_{n=1}^{N} \log P(d_n|\Theta)$$

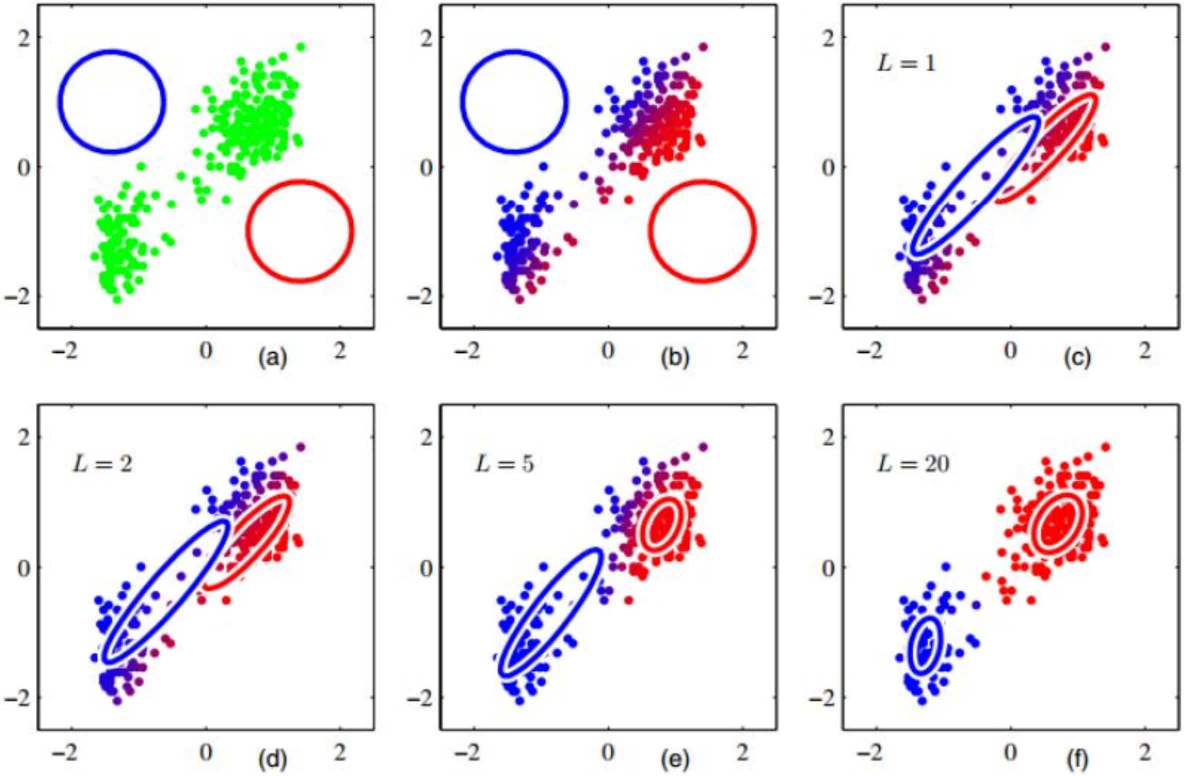
This way we'll have

1. 'fuzzy clustering' (soft clustering): **clusters probabilities** for each document,
2. possibility to restrict/give some hint on the possible **shapes (distribution family) of the cluster**

Please see for better/detailed explanations

Xu L and Jordan MI (1996). On Convergence Properties of the EM Algorithm for Gaussian Mixtures. Neural Computation 2: 129-151

EM-algorithm, visualization



Bishop, Christopher M. Pattern recognition and machine . learning. Vol. 1. New York: springer, 2006.

Plan

1. ~~Clustering~~
    a. ~~The task~~
    b. ~~Clustering quality evaluation~~
    c. ~~Clustering methods types~~
        i. ~~Representative-based~~
        ii. ~~Probabilistic~~
        iii. Hierarchical
        iv. Density-based
    d. Tools & data

2. **Finding Similar Items*
3. *Topic modeling*

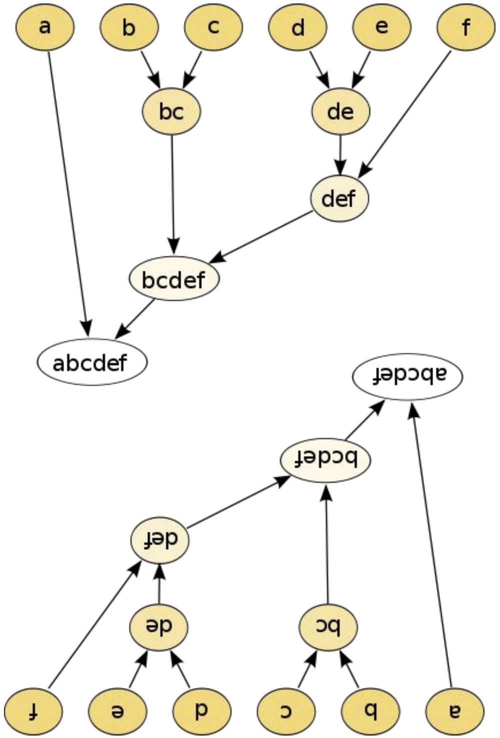
Hierarchical clustering

Two methods types

1. agglomerative
2. divisive

Each hierarchical method builds a **dendrogram** for further pruning = clusters selection

Dendrogram shows measures of closeness between objects and sets of objects



Hierarchical clustering: divisive approach

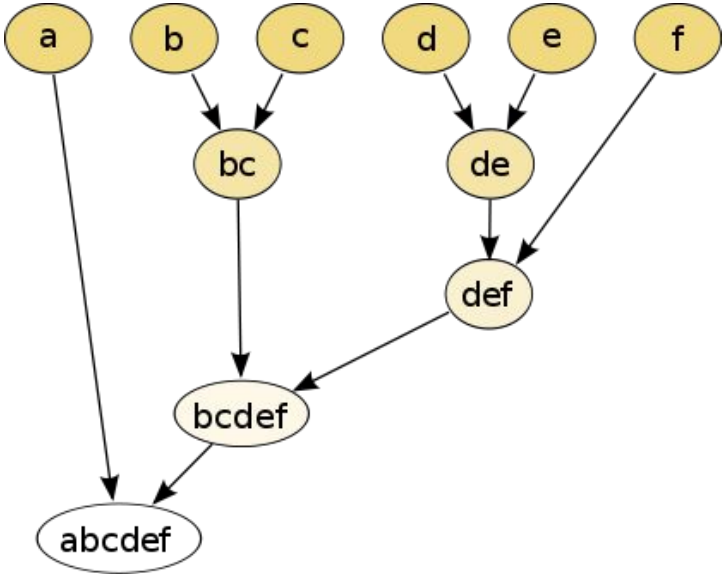
Example: let’s choose some flat clustering method A
(e.g. KMeans)

1. **Initially -- just 1 cluster** containing all elements, root of the tree
2. Apply method A to the leaf of the tree (chosen by some rule).
3. Add resulting clusters as leaves (**x** being their ‘parent’).
4. Repeat 2-3, until the cardinality of each ‘leaf’ is equal to 1.

Hierarchical clustering: agglomerative approach

A more popular and intuitive approach

1. Initially each element is a cluster of size 1
2. Using a certain rule, we choose two closest clusters and merge them into one

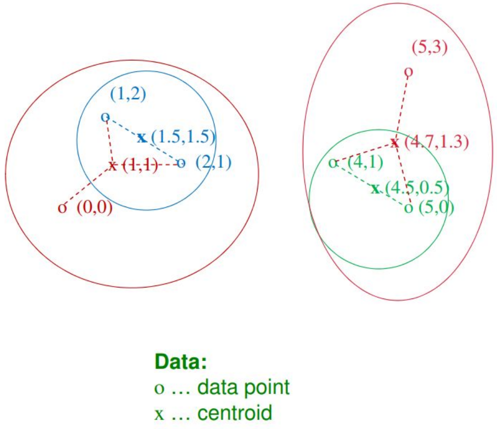


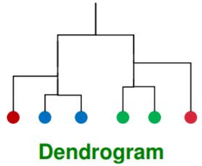
Hierarchical clustering: how to represent clusters?

1. **centroids**: distances between centroids as distances between clusters

2. **medoids:** element distances from which to each of other points in a cluster is ‘minimal’

3. Take all elements of the cluster into account (next slide)





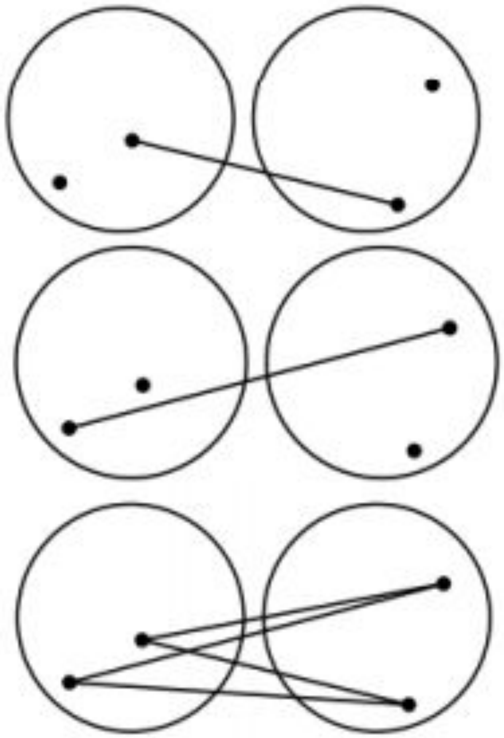
How to compute distance between clusters

1. **Single link**: distance between the two closest points from two clusters
   ‘point chains’ problem

2. **Complete link**: distance between the two farthest
   ’outliers’ problem

3. **Group average**: average distance between all pairs of points from two clusters

4. **Ward linkage**: difference between sum(sqr(distances)) inside the possible **clusters union** and sum(sqr(distances)) inside **each of the two clusters separately**



Ward, J. H., Jr. (1963), "Hierarchical Grouping to Optimize an Objective Function", Journal of the American Statistical Association, 58, 236–244.

Plan

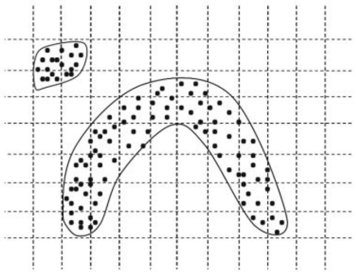
1. ~~Clustering~~
    a. ~~The task~~
    b. ~~Clustering quality evaluation~~
    c. ~~Clustering methods types~~
        i. ~~Representative-based~~
        ii. ~~Probabilistic~~
        iii. ~~Hierarchical~~
        iv. Density-based
    d. Tools & data

2. **Finding Similar Items*
3. *Topic modeling*

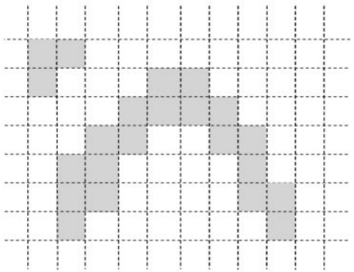
Density-based: grid-based approach

“Grid” approach: we split the space into hypercubes of the same size, we consider cubes neighbors if they have more than **r** common values in vectors (cubes with common corners, edges, nodes, etc.)

1) retain cubes that have > k points in them,
2) build graph: cubes are nodes, edges are between neighbours,
3) finding connected components in the graph.



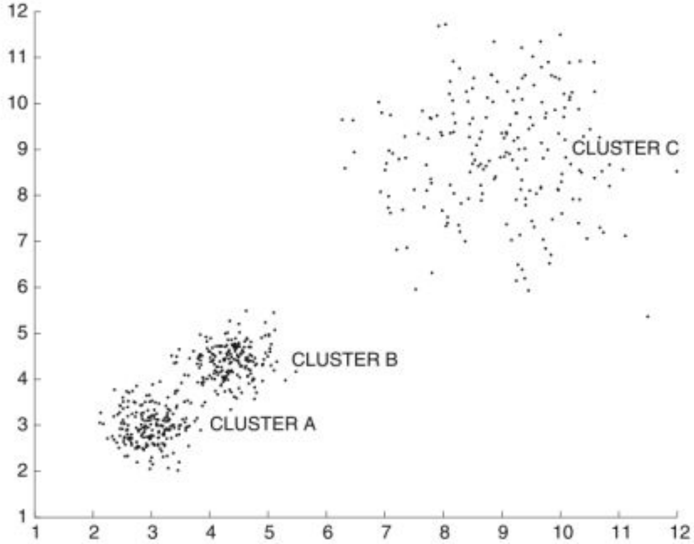
(a) Data points and grid



(b) Agglomerating adjacent grids

Grid-based approach: discussion

+ robust to outliers
+ can work with clusters of any shape
- we can’t tune parameters so that different density clusters were found, even if they look clearly serapable from human’s point of view



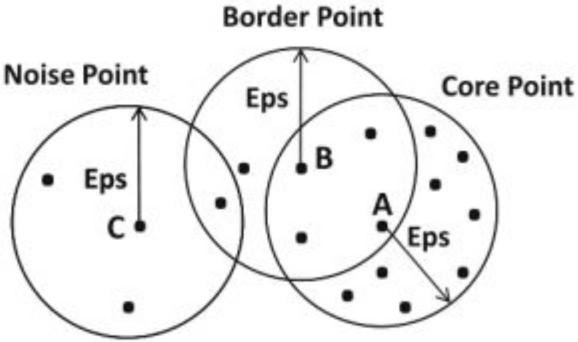
Density-based: DBSCAN

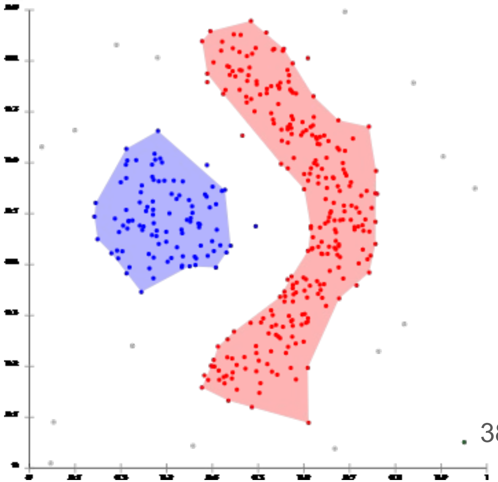
Density-Based Spatial Clustering of Applications with Noise
the most cited clustering algorithm

Set **ε** (distance) and **k** (an integer)

Elements are split into 3 types:

1. Core points: elements having at least **k** other elements in their **ε-neighbourhood**
2. Border points: elements having at least b element in their **ε**-neighbourhood
3. Noise points: other elements

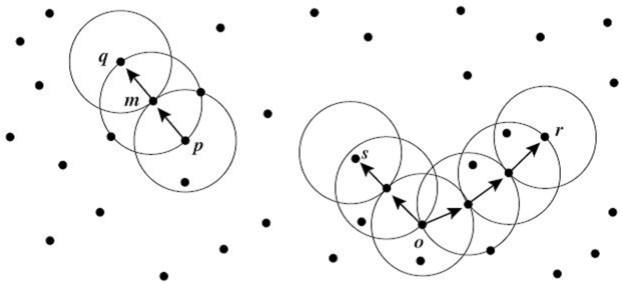




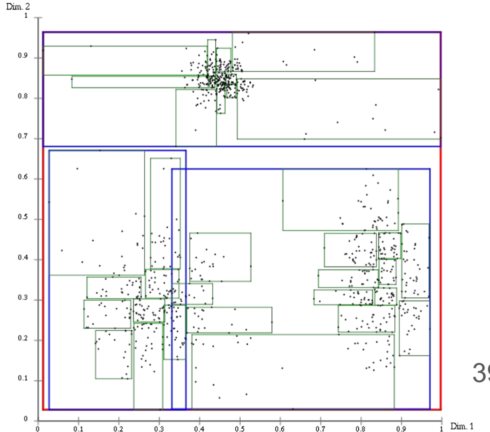
Density-based: DBSCAN

Algorithm

1) Mark elements with those three types
2) Build a graph, connecting core points that are no farther than ε from each other
3) Determine connected components
4) Link every border point to the closest connected component



https://stackoverflow.com/questions/2303510/recommended-anomaly-detection-technique-for-simple-one-dimensional-scenario



DBSCAN: discussion

Graph construction -- actually a ‘single-link’ hierarchical clustering algorithm with ***ε****-cutoff*

- \+ determines the number of clusters
- \+ robust to outliers and noise
- \+ detects clusters of arbitrary shape and form
- \- is slow
- \- fails at determining clusters of different density
- \- tuning parameters may be a challenge

Other famous algorithms worth reading up on

- **CURE**
  (Clustering Using REpresentatives: hybrid of hierarchical and flat clustering; keeping several representative data points for each cluster)

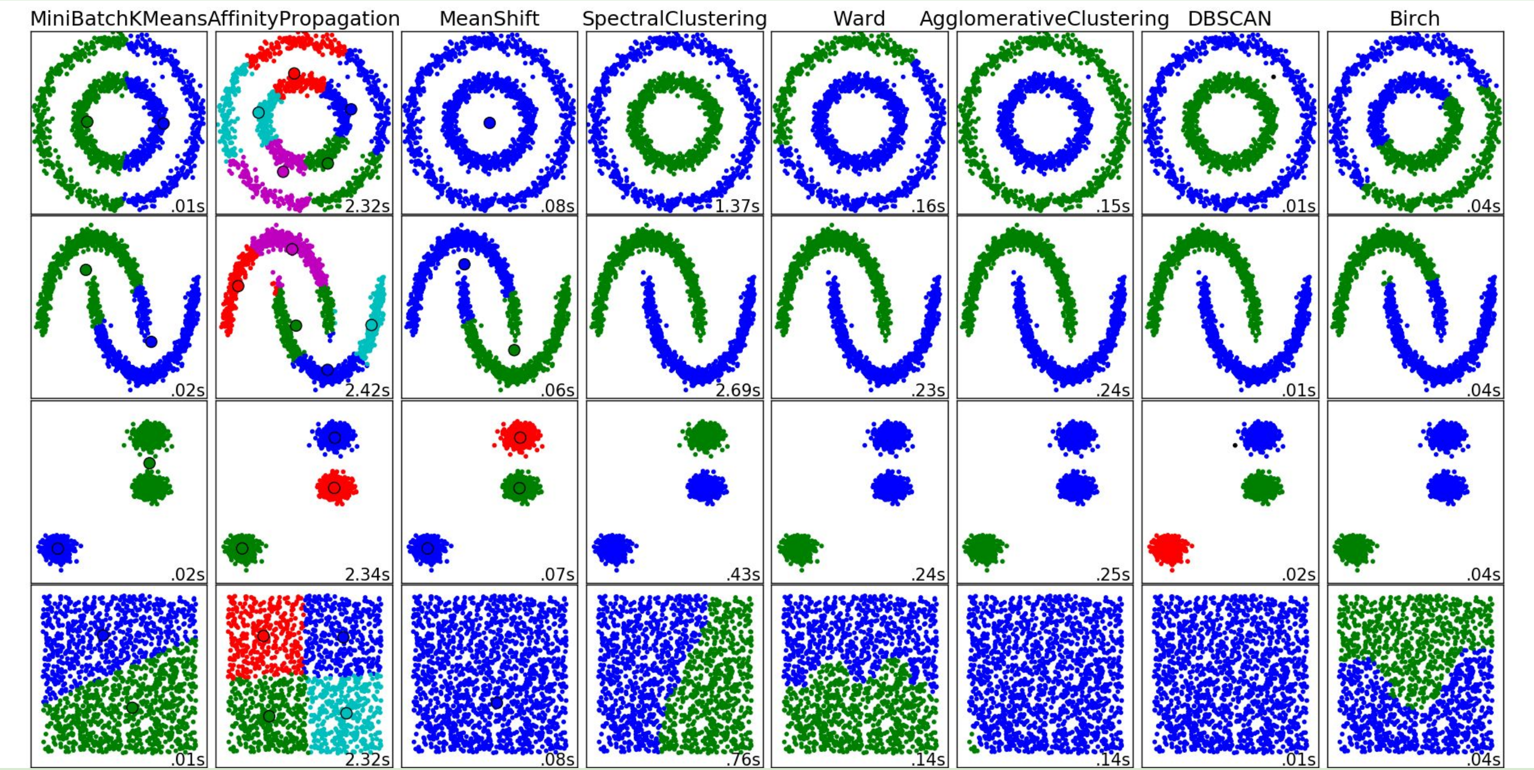
- **BIRCH**
  (hierarchical, designed for large datasets, we build a tree of subclusters, preserving certain constraints, SIGMOD 10y test time award)

- **OPTICS** and other DBSCAN modifications
  (DBSCAN taking density into account)

- "Community detection" **in graphs**

- Word clustering algorithms (mentioned in lectures on vector semantics); most popular one is **Brown clustering**
  Brown, Peter F., et al. "Class-based n-gram models of natural language."Computational linguistics 18.4 (1992): 467-479

Homework: read up on the methods and think why results look like that



Plan

1. ~~Clustering~~
    a. ~~The task~~
    b. ~~Clustering quality evaluation~~
    c. ~~Clustering methods types~~
        i. ~~Representative-based~~
        ii. ~~Probabilistic~~
        iii. ~~Hierarchical~~
        iv. ~~Density-based~~
    d. Tools & data

2. **Finding Similar Items*
3. *Topic modeling*

Tools & Data

Mainstream instruments allowing to try different approaches

- scipy.cluster
- sklearn.cluster
- custom libraries, e.g. pyclustering

Data

- UCI Machine Learning Repository: Clustering + Text

Used/recommended materials

1. CSC 2014 course
2. Mining of Massive Datasets Jure Leskovec, Anand Rajaraman, Jeff Ullman
3. Scikit-learn docs
4. MSU course slides and other materials
5. EM-algorithm @ ml.ru
6. Wikipedia (English)

High-level structure in texts as sets of words - I

Anton Alekseev,
Steklov Mathematical Institute in St Petersburg

NRU ITMO, St Petersburg, 2019

anton.m.alexeyev+itmo@gmail.com

Thanks for helping with the slides go to Daria Maglevanaya