K-means clustering and other clustering methods

Jo Eidsvik

Exam!

- One proper exercise.
- Home exam questions.

Grouping data

- Split data in similar groups (clusters).
- Use these clusters for subsequent classification of other data.
- Distances play a crucial role in both tasks.

Gaussian mixtures

(Sect 4.3 in Steinley)
Density for a Gaussian mixture (with equal weights and covariance):

$$p(\boldsymbol{x}) = \frac{1}{K} \sum_{b=1}^{K} \phi_N(\boldsymbol{x}; \boldsymbol{\mu}^b, \boldsymbol{\Sigma}).$$

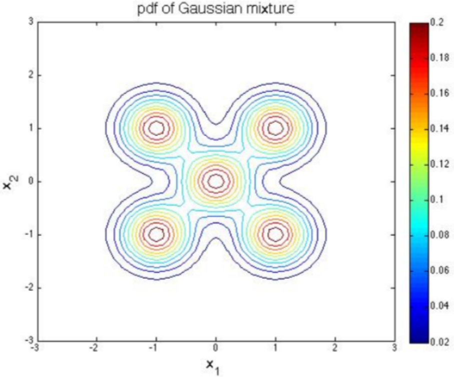
ϕ_N represents the N variate Gaussian density function. $\boldsymbol{x} = (x_1, \ldots, x_N)^t$.
$\boldsymbol{\mu}^b = (\mu_1^b, \ldots, \mu_N^b)^t$ is the mean for the bth component of the mixture.
$\boldsymbol{\Sigma}$ is a $N \times N$ positive definite covariance matrix.

Mixtures

Density for a Gaussian mixture (with equal weights and covariance):

$$p(\boldsymbol{x}) = \frac{1}{K}\sum_{b=1}^{K} \phi_N(\boldsymbol{x}; \boldsymbol{\mu}^b, \boldsymbol{\Sigma}).$$

Illustration $K = 5$ (small covariance):

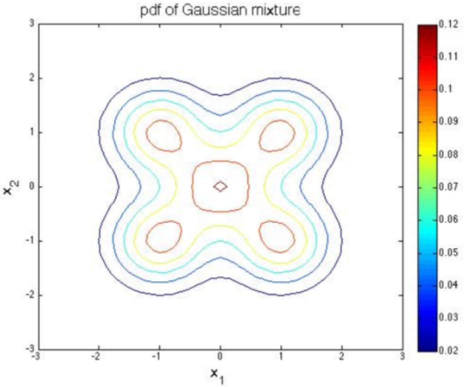


Mixtures

Density for a Gaussian mixture (with equal weights and covariance):

$$p(\boldsymbol{x}) = \frac{1}{K}\sum_{b=1}^{K} \phi_N(\boldsymbol{x}; \boldsymbol{\mu}^b, \boldsymbol{\Sigma}).$$

Illustration $K = 5$ (medium covariance):

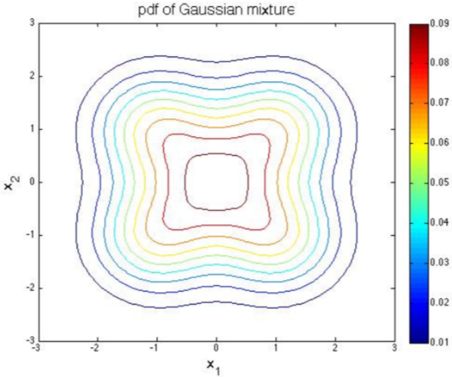


Mixtures

Density for a Gaussian mixture (with equal weights and covariance):

$$p(\boldsymbol{x}) = \frac{1}{K} \sum_{b=1}^{K} \phi_N(\boldsymbol{x}; \boldsymbol{\mu}^b, \boldsymbol{\Sigma}).$$

Illustration $K = 5$ (large covariance):



General type for Gaussian mixture model

Density for a Gaussian mixture:

$$p(\boldsymbol{x}) = \sum_{b=1}^{K} w_b \phi_N(\boldsymbol{x}; \boldsymbol{\mu}^b, \boldsymbol{\Sigma}^b), \quad \sum_{b=1}^{K} w_b = 1.$$

This can be regarded as the basis for discriminant analysis used in classification and clustering.

General type for Gaussian mixture model

Density for a Gaussian mixture:

$$p(\boldsymbol{x}) = \sum_{b=1}^{K} w_b \phi_N(\boldsymbol{x}; \boldsymbol{\mu}^b, \boldsymbol{\Sigma}^b), \quad \sum_{b=1}^{K} w_b = 1.$$

This can be regarded as the basis for discriminant analysis used in classification and clustering.

Classification by Linear discriminant analysis

Assume $w_b = 1/K$, $\boldsymbol{\mu}^b$ and $\boldsymbol{\Sigma}^b = \boldsymbol{\Sigma}$, $b = 1, \ldots, K$ are known from training.
What is the most likely class b for a new data point $\boldsymbol{x}$?
Classify $\boldsymbol{x}$ to the class which has the largest element density $\phi_N(\boldsymbol{x}; \boldsymbol{\mu}^b, \boldsymbol{\Sigma})$:

$$\hat{b} = \operatorname{argmax} \left[\phi_N(\boldsymbol{x}; \boldsymbol{\mu}^1, \boldsymbol{\Sigma}), \ldots, \phi_N(\boldsymbol{x}; \boldsymbol{\mu}^K, \boldsymbol{\Sigma})\right],$$

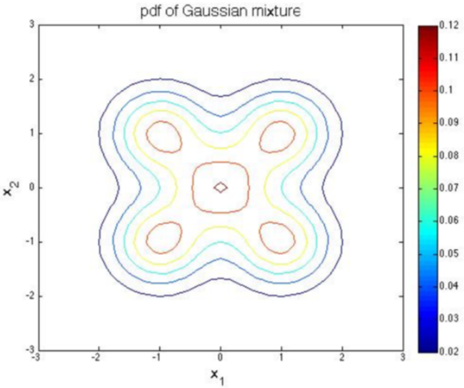
Decision boundary class b and c:

$$2\boldsymbol{x}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}^b - \boldsymbol{\mu}^{b^t}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}^b = 2\boldsymbol{x}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}^c - \boldsymbol{\mu}^{c^t}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}^c.$$

Mixtures and boundaries

$$2\mathbf{x}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}^b - \boldsymbol{\mu}^{b^t}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}^b = 2\mathbf{x}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}^c - \boldsymbol{\mu}^{c^t}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}^c.$$

Boundaries go between modes.
Illustration $K = 5$ (medium covariance):



Classification by Quadratic discriminant analysis

Assume $w_b = 1/K$, $\boldsymbol{\mu}^b$ and $\boldsymbol{\Sigma}^b$, $b = 1, \ldots, K$ are known from training.
What is the most likely class b for a new data point $\boldsymbol{x}$?
Classify $\boldsymbol{x}$ to the class which has the largest element density $\phi_N(\boldsymbol{x}; \boldsymbol{\mu}^b, \boldsymbol{\Sigma})$:

$$\hat{b} = \text{argmax} \left[\phi_N(\boldsymbol{x}; \boldsymbol{\mu}^1, \boldsymbol{\Sigma}^1), \ldots, \phi_N(\boldsymbol{x}; \boldsymbol{\mu}^K, \boldsymbol{\Sigma}^K) \right],$$

Decision boundary class b and c:

$$-\boldsymbol{x}^t \boldsymbol{\Sigma}^{b,-1} \boldsymbol{x} + 2\boldsymbol{x}^t \boldsymbol{\Sigma}^{b,-1} \boldsymbol{\mu}^b - \boldsymbol{\mu}^{b^t} \boldsymbol{\Sigma}^{b,-1} \boldsymbol{\mu}^b = -\boldsymbol{x}^t \boldsymbol{\Sigma}^{c,-1} \boldsymbol{x} + 2\boldsymbol{x}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}^c - \boldsymbol{\mu}^{c^t} \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}^c.$$

Classification in Gaussian mixtures

Assume w_b, $\boldsymbol{\mu}^b$ and $\boldsymbol{\Sigma}^b$, $b = 1, \ldots, K$ are known from training.
What is the most likely class b for a new data point $\boldsymbol{x}$?
Classify $\boldsymbol{x}$ to the class which has the largest element density $\phi_N(\boldsymbol{x}; \boldsymbol{\mu}^b, \boldsymbol{\Sigma})$:

$$\hat{b} = \text{argmax}\left[w_1 \phi_N(\boldsymbol{x}; \boldsymbol{\mu}^1, \boldsymbol{\Sigma}^1), \ldots, w_K \phi_N(\boldsymbol{x}; \boldsymbol{\mu}^K, \boldsymbol{\Sigma}^K)\right],$$

Weights can be interpreted as prior probabilities of classes :
$w_b = P(\boldsymbol{x} \in b)$, $b = 1, \ldots, K$, $\sum_{b=1}^{K} w_b = 1$.

Training in mixtures : supervised learning

From labeled data, one can train the model parameters w_b, $\boldsymbol{\mu}^b$ and $\boldsymbol{\Sigma}^b$, $b = 1, \ldots, K$.
Labeled data means that we know the class for each dataset. The data are then $(\boldsymbol{x}^1, b^1), \ldots, (\boldsymbol{x}^n, b^n)$. Weights are fraction in class, mean and covariance are computed in the usual way from data in the relevant class:

$$\hat{w}^b = \frac{\sum_{i=1}^n I(b^i = b)}{n}$$

$$\hat{\boldsymbol{\mu}}^b = \frac{\sum_{i=1}^n I(b^i = b)\boldsymbol{x}^i}{\sum_{i=1}^n I(b^i = b)}$$

$$\hat{\boldsymbol{\Sigma}}^b = \frac{\sum_{i=1}^n I(b^i = b)(\boldsymbol{x}^i - \hat{\boldsymbol{\mu}}^b)(\boldsymbol{x}^i - \hat{\boldsymbol{\mu}}^b)^t}{\sum_{i=1}^n I(b^i = b)}$$

(Data could be sampled in non-random manner, naturally, or on purpose (stratified sampling to balance fraction in groups).

Training in mixtures : unsupervised learning

From unlabeled data, it is more difficult to train the model parameters w_b, $\boldsymbol{\mu}^b$ and $\boldsymbol{\Sigma}^b$, $b = 1, \ldots, K$.
Unlabeled data means that we **do not** know the class for each dataset.
The data are then $(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^n)$.
Weights, mean and covariance must be specified by more complex optimization methods.

EM algorithm

In statistics the *expectation-maximization* (EM) algorithm, iteratively searches for the likely sets of weights, means and covariances.
The EM algorithm starts by initial parameters values. Then each step of the iterative algorithm consists of

- Expectation: The expected values is taken over the indicators (or the likelihood), given the current parameter values
- Maximization: New estimates of the parameters are obtained from the expression obtained by the expected expression.

Solution is non-unique. It tends to depends a lot on the initial parameters values.
(First defined in a general setting in the statistics literature in Demster et al (1977), although variants for different special cases were provided, such as k-means clustering.)

K-means clustering

From **unlabeled data**, split data $(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^n)$ in K classes or clusters. Elements of K-means clustering:

- Except in special cases, results can be non-unique (as with the EM algorithm). They often depend on the initial values of the algorithm.
- It is using distance measures, and is not tied to a statistical distributions (as the EM algorithm is).
- It is fast to compute and implemented in most software packages. *kmeans* in R and MATLAB. *from sklearn.cluster import KMeans* in Python.
- Not obvious to choose K.

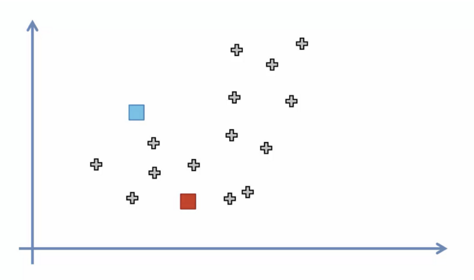
K-means algorithm

Initialize with K points (called centroids). Iterate the following until no further changes in sets.

1. Compute from each point to all K centroids.
2. Allocate each point to a cluster associated with the nearest centroid.
3. Update the centroids as the cluster means.

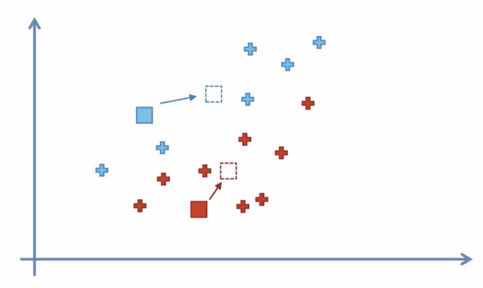
Illus two dimensional data and $K = 2$

Initial:



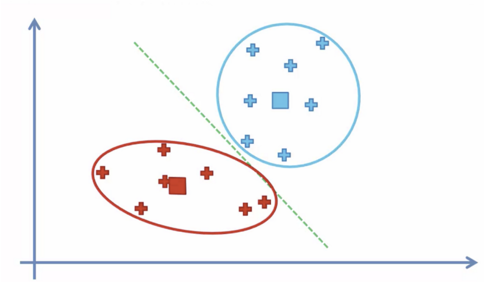
Illus two dimensional data and $K = 2$

Update:

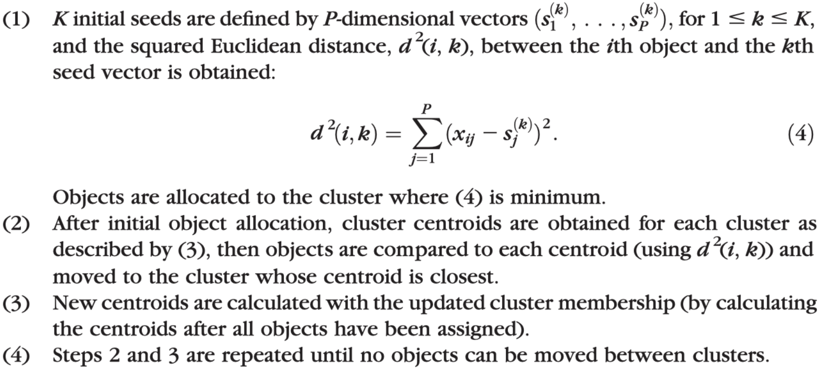


Illus two dimensional data and $K = 2$

Final:



Alg from Steinley paper:



(1) K initial seeds are defined by P-dimensional vectors $(s_1^{(k)}, \ldots, s_P^{(k)})$, for $1 \leq k \leq K$, and the squared Euclidean distance, $d^2(i, k)$, between the ith object and the kth seed vector is obtained:

$$d^2(i,k) = \sum_{j=1}^{P} (x_{ij} - s_j^{(k)})^2. \tag{4}$$

Objects are allocated to the cluster where (4) is minimum.

(2) After initial object allocation, cluster centroids are obtained for each cluster as described by (3), then objects are compared to each centroid (using $d^2(i, k)$) and moved to the cluster whose centroid is closest.

(3) New centroids are calculated with the updated cluster membership (by calculating the centroids after all objects have been assigned).

(4) Steps 2 and 3 are repeated until no objects can be moved between clusters.

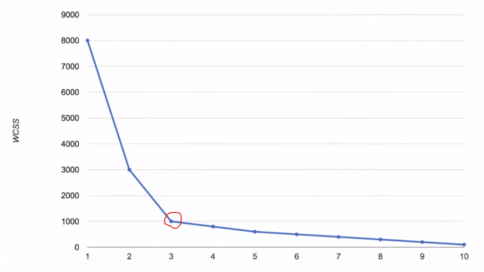
Initial points

The Achilles' heel of the K-means algorithm is the initialization, which leads to a local minimum.

- Try several different initializations and check sensitivity.
- Remove outliers or select influential points in pre-processing steps.
- Use another simple algorithm for the initialization (Ward's method which is sequential starting at n clusters and linking variables that minimize in an analysis of variance procedure.).
- Variable selection in first steps.
- Add constraints to stabilize approach (number of points in each cluster, enforce similar clusters, distance between centroids, etc.) These also speed up the algorithm.

Choosing K

There are diagnostic plots (elbow plot), etc that can indicate which K is suitable to get a reasonable match between cluster size and predictive power to a hold out set.



Akaike's information criterion can be used to select K (likelihood-based). Gap-statistic is using the sum of squares within (SSW) clusters (equivalent the within cluster sum of squares; WCSS), and statistical properties of the sampled data.

Distances

In most cases the Euclidean distance is used for the K-means algorithm, but other measures could be used:

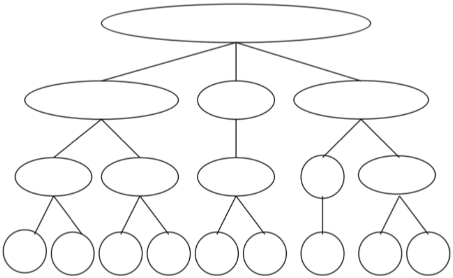
- K-median clustering
- Variable reduction / Projection in MDS space. (In high dimensions distances are often large and similar.)

Other unsupervised clustering methods

- Graph or tree learning.
- Self organizing maps
- (variational) autoencoders.

Tree - graphs clusters

Form a tree based on the 'optimal' split.
Often done sequentially forward according to some selected criterion.



Self-organizing maps

Sometimes called Kohonen maps (Kohonen, 1980).
Builds a best-matching unit for points.
Weights to best matching units are based on distances in a neural network representation.
Clusters are easy to visualize.

Project

- Conduct Dynamic Time Warping on a dataset (you simulate yourself).
- Conduct K-means clustering of data in the 2D MDS space of the data.

DTW

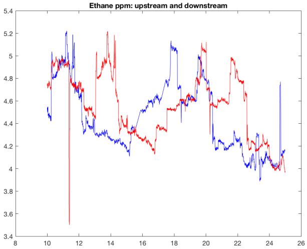


Figure: Gas-pipe ethane measured in Norway, and in Germany.

Align time series.

DTW

1. Simulate a length 500 autoregressive process of order 1; $x(1), \ldots, x(500)$, with mean 0, stationary variance 1 and autocorrelation parameter $\phi = 0.9$.
2. Construct a path such that $j(1) = 51, j(2) = 52, \ldots, j(200) = 250$.
   $j(i) = 250$ for $i = 201, \ldots, 240$,
   $j(241) = 251, j(242) = 252, \ldots j(500) = 510$.
3. Simulate another time series $y(j) = x(i) + \epsilon_j$, $\epsilon_j \sim N(0, 0.15^2)$, $j = 51, \ldots, 510$.
4. Use DTW to extract the most likely path.

Repeat the process for a few replicate simulations, but with the same path. Plot the variability in the extracted paths.

Code for DTW

Use established code in your software of preference:
MATLAB and R: dtw.
dtw-python
These also give the distance matrix to the warping possibilities.

Clustering and MDS

Run two varieties of random walks of length 100 on the line and compare the results.
One model of a random walk (50 first runs) has 0.5 probability of walking left / right.
The other model one (50 last runs) has 0.6 probability of walking to the right, and 0.4 probability of walking left.

Clustering and classification of simulations

Plot the two runs in different colors. Can you see a tendency of a difference?
Conduct MDS and visualize all the 100 datasets in s 2D plot.
Do 2-means clustering in the MDS space.
Count the number of correctly clustered datasets.