Data Mining: Lecture 2

Cluster Analysis

S. Nõmm

[1] Department of Software Science, Tallinn University of Technology

07.09.2021

Introduction

Given a set of data points, partition them into groups with respect to chosen similarity criteria.

- Data summarization.
- Discover the structure of the set.
- Part of preprocessing.

Feature selection

Given a set of data points, partition them into groups with respect to chosen similarity criteria.

- Filter Models
  - Predictive Attribute Dependence
  - Entropy

$$E = -\sum_{i=1}^{m} \left[p_i \log(p_i) + (1 - p_i) \log(1 - p_i) \right]$$

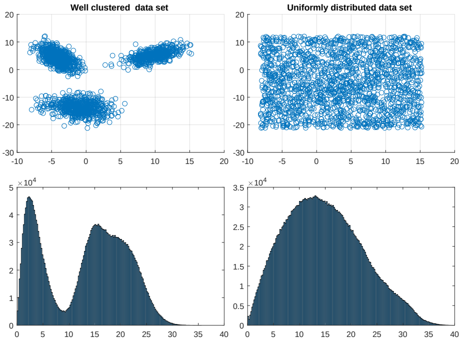
  - Hopkins Statistic

$$H = \frac{\sum_{i=1}^{r} \beta_i}{\sum_{i=1}^{r} (\alpha_i + \beta_i)}.$$

- Wrapper models

Entropy 1

- Underlying idea is that features with uniformly distributed values carry less information compared to those distributed non uniformly.
- Distance distributions of well-clustered sets should be different from those uniformly distributed.



Entropy 2

- Consider $k-$ dimensional subset of the feature set.
- Using ϕ ranges for each dimension discretize the data set. This step results into $m = \phi^k$ regions.
- Observe, that for each evaluated feature subset m is expected to be approximately the same.

$$E = -\sum_{i=1}^{m} \left[p_i \log(p_i) + (1 - p_i) \log(1 - p_i) \right].$$

where p_i is the proportion of the points in the region i, m - total number of regions. Large values of E indicate poor clustering behaviour.

Measures

Hopkins statistics. Let $\mathcal{D}$ be the data set to investigate and $\mathcal{R}$ is a representative sample of $\mathcal{D}$, of power r. $\mathcal{S}$ is a synthetic data set of r data points randomly generated from the same domain. Let $\alpha_1, \dots \alpha_r$ be the distances of each point of $\mathcal{R}$ to the nearest neighbour in $\mathcal{D}$ and $\beta_1, \dots \beta_r$ are the distances of each point of $\mathcal{S}$ to the nearest neighbour in $\mathcal{D}$. The Hopkins statistic is defined as follows:

$$H = \frac{\sum_{i=1}^{r} \beta_i}{\sum_{i=1}^{r} (\alpha_i + \beta_i)}.$$

Higher values of H indicate highly clustered data.

Feature selection

- Filter Methods: Use Entropy or Hopkins Statistics to decide set of features leads best clustering behaviour. Filter methods meay be applied on the stage of preprocessing.
- Warper models: clustering algorithm is used to evaluate the quality of subset of features.

Classification of clustering techniques

Most common clustering techniques may be classified as follows:

- **Representative based techniques:** k-means, k-medians, k-medoids, etc. Each cluster has a representative which is either the element of the data set or an element from the same space as all other elements of the dataset. Shape of the clusters is affected by the choice of distance function. Number of clusters is usually a hyperparameter.
- **Hierarchical clustering techniques:** Agglomerative and Divisive techniques. Not always relies on the distance function. Different levels of clustering granularity provide different provide different application specific insides.
- **Grid and Density based techniques:** Relies on the local density of the data points. Well suited for the clusters of irregular shapes.
- **Probabilistic algorithms:** EM and EM-like algorithms.

Hyperparameter is the parameter which value is not determined during the learning.

As a result of clustering each element is assigned label describing which cluster element belongs. NB! Similarity and distances are synonyms.

K - means

K - means is one of the most popular algorithms belongs to the class of iterative descent methods.

- It is intended for the quantitative variables.
- Squared Euclidean distance as dissimilarity measure.
- The idea is to assign close points to the same cluster. Minimize natural loss ("energy") function.

  $$W(C) = \frac{1}{2} \sum_{k=1}^{K} N_k \sum_{C(i)=k} |x_i - \bar{x}_k|^2.$$

  where $\bar{x}_k$ is the mean vector associated with the kth cluster (*centroid*). $N_k = \sum_{i=1}^{N} I(C(i) = k)$.
- Iterative descent algorithm is used to achieve this goal.

Representative based clustering

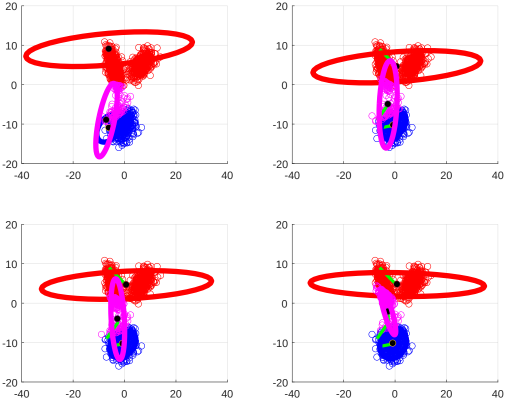
K-means:

- Hyperparameters: K - number of desired clusters, distance function.
- Intialize: generate K random points from the same limits as initial dataset. These points are referred as *centroid*.
- Repeat:
  - For each point assign the label of closest centroid.
  - For each label recompute centroid as the mean of all points with given label.
- Until converge.
- Report labels of each point.

Other representative based techniques differ only by the way representative is find.

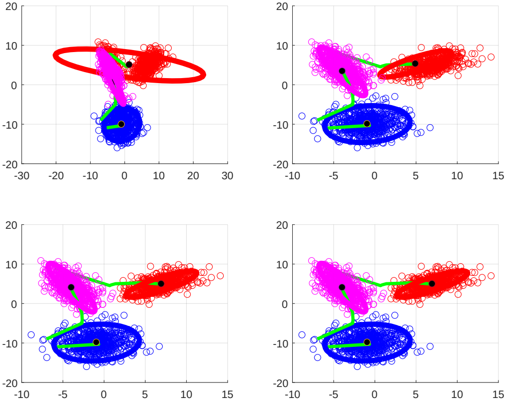
K-means clustering example

Steps 1 - 4



K-means clustering example

Steps 5 - 8



K - means, example discussion

- Convergence criteria?
  - Assignments do not change?
  - Minimum of a loss function?
- Relations to the EM-algorithm? Instead of maximizing likelihood K - means minimizes loss function.
- K - means best perform when clustered dataset composed of spherical or similar subsets.
- How to validate quality of clustering?

Validation

- **Sum of square distances to centroids.** (SSQ) This criteria is suited for K-means since it minimizes the loss function. (With reservations)
- **Intracluster to intercluster distance ratio.** Sample r points from the data set. Let P be the set of pairs that belong to the same cluster and Q the set of remaining pairs.

$$II = \frac{\sum\limits_{(x_i,x_j)\in P} S(x_i,x_j)/|P|}{\sum\limits_{(x_i,x_j)\in Q} S(x_i,x_j)/|Q|}$$

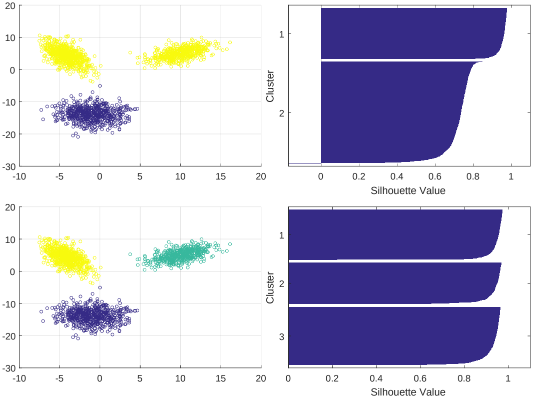
  Small values of the ratio indicate better clustering behaviour.
- **Silhouette coefficient**

$$s(i) = \frac{D^{out}_{min_i} - D^{in}_{avg_i}}{max\{D^{out}_{min_i}, D^{in}_{avg_i}\}}$$

  where $D^{in}_{avg_i}$ is the average distance of point x_i to points within the cluster it belong to. Compute average distance of point x_i to the points of each cluster. Let $D^{out}_{min_i}$ is the minimum of these average distances. $s(i) \in (-1, 1)$. Overall coefficient is the average of the individual points coefficients. Large positive values indicate highly separated clusters.

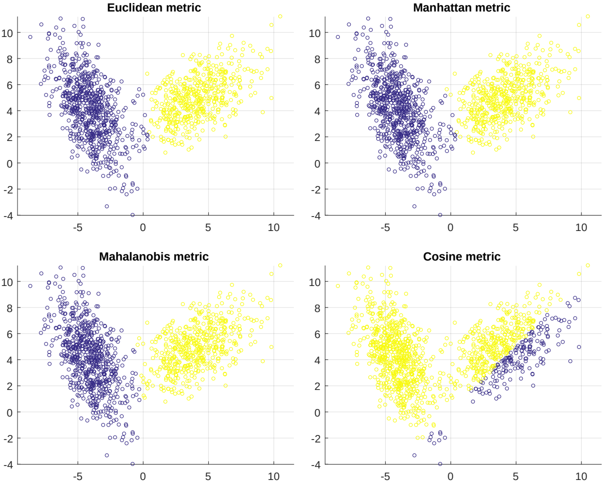
Silhouette coefficient

- Considered to be most popular criteria for clustering validation.
- Silhouette plot is the graphic representation of the silhouette coefficient.
- Overall silhouette coefficient may be used to determine number of clusters.

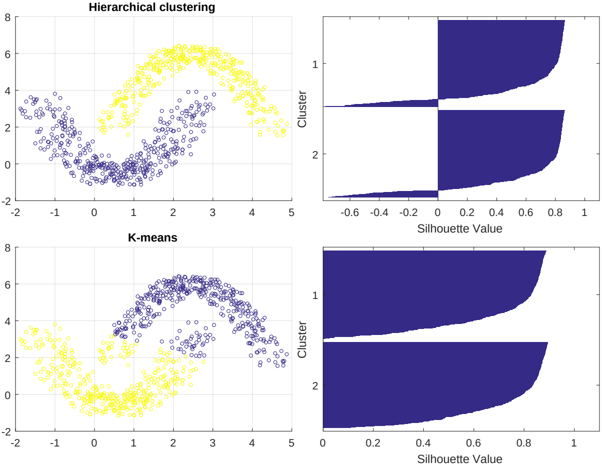


Impact of distance functions

NB! Always observe if distance function is defined for the given dataset and if using it makes sense from the viewpoint of interpretation.



Limitations

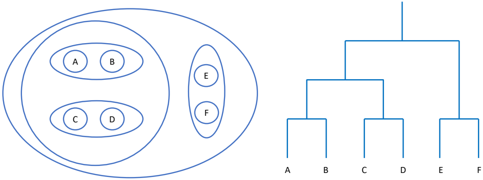


Hierarchical clustering: Agglomerative clustering

Some times referred as *bottom-up*

Algorithm

- Initialize $n \times n$ distance matrix $\mathcal{M}$
- **Repeat**
    - Choose closest pair of clusters (i, j) based on $\mathcal{M}$.
    - Merge clusters i and j and update matrix $\mathcal{M}$.
- **Until** termination criterion.
- Return cluster labels for each point.



Group-based statistics

Also referred as *linkage*.

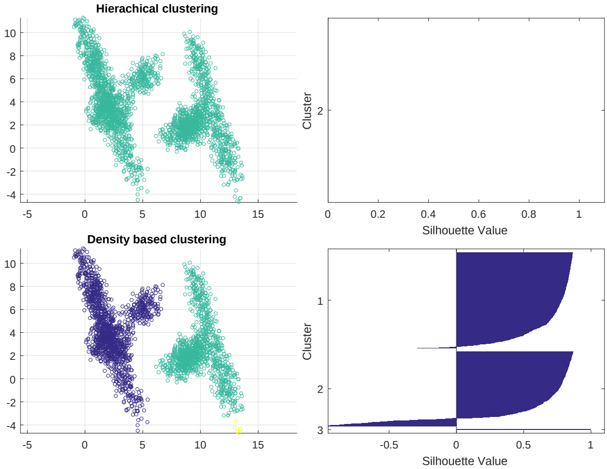
- Best (single) linkage. Distance is equal to the minimum distance between all pairs of elements (from two groups). Suitable to discover clusters of arbitrary shape. Drawback noise points may merge distant clusters.
- Worst (complete) linkage.(Complete linkage method) Distance is equal to the maximum distance between all pairs of elements (from two groups). Attempts to minimize maximual diameter of the cluster.
- Group average linkage. Distance between two groups is equal to the average of the distances between all pairs of elements (from two groups).
- Closest centroid. Clusters with closest centroid are merged.
- Variance based criterion. Minimizes the change in the objective function a result of merging.
- Ward's method, like previous but instead of variance observes changes in some od squared error.

Top-down divisive methods

Algorithm

- Initialize tree $\mathcal{T}$ to root containing dataset $\mathcal{D}$
- **Repeat**
  - Select a lead node $\mathcal{L}$ in $\mathcal{T}$ based on predefined criterion.
  - Use splitting algorithm $\mathcal{A}$ to split $\mathcal{L}$ into $\mathcal{L}_1, \ldots, \mathcal{L}_k$.
  - add $\mathcal{L}_1, \ldots, \mathcal{L}_k$ as children of $\mathcal{L}$ in $\mathcal{T}$.
- **Until** termination criteria.

Limitations



Grid- and density- based methods

One of the major problems with distance-based and probabilistic methods is that the shape of the underlying clusters is already defined implicitly by the underlying distance function or probability distribution. Possible solutions:

- Grid- based methods
- Density- based methods
- Graph- based algorithms
- Nonnegative matrix factorization

Grid- and Density-based clustering

Explores the idea, that clusters are of a different density than space between them. May be see as the sub class of agglomerative methods.

Generic Grid:

Hyperparameters: Ranges and density threshold τ.

- Discretize each dimension into p ranges.
- Determine *dense* grid cells at level τ.
- Create graph in whichdense grids are connected if they are adjacent.
- Determine connected components of the graph.
- Return cluster indexes for each point.

DBSCAN

Let $\mathcal{D}$ denote the data set, τ - density threshold and ϵ - radius of the neighborhood.

Definition

Core point: A data point is defined as the core point, if its ϵ - neighbourhood contains at least τ data points.

Definition

Border point: A data point is defined as the border point, if its ϵ - neighbourhood contains at least one another data point of $\mathcal{D}$ and at least one core point.

Definition

Noise point: Is defined as data point of $\mathcal{D}$ which neither core point nor border point.

DBSCAN

Algorithm:

- Determine Core, border and noise points for given ϵ and τ.
- Create graph in which core points are connected (if they are within ϵ of one another).
- Assign each border point to a connected component.
- Return cluster indexes for each point.

EM-algorithm

Let us consider K-Means from the probabilistic point of view.

- (E-step) Each data point of the set $\mathcal{D}$ has a probability belonging to cluster j, which is proportional to the scaled and exponentiated Euclidean distance to each representative Y_j . In the k-means algorithm, this is done in a "hard" way, by choosing the smallest Euclidean distance to the representative of Y_j.
- (M-step) The center Y_j is the weighted mean over all the data points where the weight is defined by the probability of assignment to cluster j. The hard version of this is used in k-means, where each data point is either assigned to a cluster or not assigned to a cluster (i.e., 0-1 probabilities).

EM-algorithm

Assumption: the data was generated from a mixture of k distributions with probability distributions $\mathcal{G}_1 \ldots \mathcal{G}_k$. Each distribution $\mathcal{G}_i$ represents a cluster and is also referred to as a mixture component.

- (E-Step) Given the current value of the parameters in , estimate the posterior probability $P(\mathcal{G}_i|X_j, \Theta)$ of the component $\mathcal{G}_i$ having been selected in the generative process, given that we have observed data point X_j . The quantity $P(\mathcal{G}_i|X_j, \Theta)$ is also the soft cluster assignment probability that we are trying to estimate. This step is executed for each data point X_j and mixture component G_i.
- (M-Step) Given the current probabilities of assignments of data points to clusters, use the maximum likelihood approach to determine the values of all the parameters in Θ that maximize the log-likelihood fit on the basis of current assignments.

Gaussian

- Gaussian or normal distribution. Its probability density function is given by
$$\mathcal{N}(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big(-\frac{1}{2\sigma^2}(x-\mu)^2\Big)$$
where μ is the mean,σ is the variance and $\sqrt{2\pi\sigma^2}$ is the normalization constant.
- Multivariate Gaussian or Multivariate Normal (MVN). Probability density function is given by.
$$\mathcal{N}(x|\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\Big[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\Big]$$
where μ is the mean vector, Σ is covariance matrix of the data set, d is the dimensionality of the data set.

Geometric interpretation

$$\mathcal{N}(x|\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\Big[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\Big]$$

- Expression under the exponent is Mahalanobis distance between point x and mean.
- Perform an eigendecomposition of Σ.

$$\Sigma^{-1} = U^{-T}\Lambda U^{-1} = U\Lambda^{-1}U^T = \sum_{i=1}^{D}\frac{1}{\lambda_i}u_i u_i^T$$

where u_i is the i'th colum of U (ith eigenvector).

- Rewrite Mahalanobis distance and denote $y_i = u_i^T(x-\mu)$

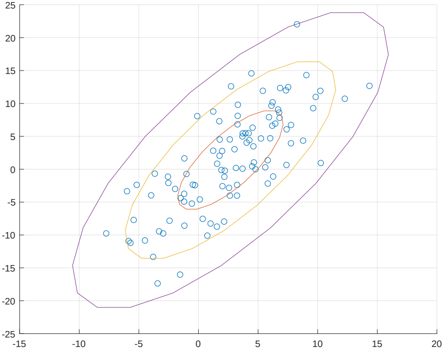
$$(x-\mu)^T\Sigma^{-1}(x-\mu) = (x-\mu)^T\sum_{i=1}^{D}\frac{1}{\lambda_i}u_i u_i^T(x-\mu)$$

$$= \sum_{i=1}^{D}\frac{1}{\lambda_i}(x-\mu)^T u_i u_i^T(x-\mu) = \sum_{i=1}^{D}\frac{y_i^2}{\lambda_i}.$$

Geometric interpretation: example

$$(x-\mu)^T\Sigma^{-1}(x-\mu)=\sum_{i=1}^{D}\frac{y_i^2}{\lambda_i}.$$

Contours of equal probability density of a gaussian lie along ellipses.



Likelihood

- **Likelihood:** Roles of parameters and outcomes distinguish likelihood from probability. Probability describes how possible the outcome before data is available, given the values of parameter. Likelihood describe possibility of parameter values given available data.
  - *Discrete:* Let X be a discrete random variable and p its probability mass function then
    $$\mathcal{L}(\theta|x) = p_\theta(x),$$
    is called likelihood function of θ given the outcome x.
  - *Continuous:* Let X be a continuous random variable and f its density function.
    $$\mathcal{L}(\theta|x) = f_\theta(x).$$
    is called likelihood function of θ given the outcome x.

NB! Note the difference with conditional probabilities.

Prior and posterior

It is presumed that *new* data is expected during the process.

- **Prior** Prior probability is the probability of the event (before collection of a *new* data).
- **Posterior** Posterior probability of the event is the probability of the event (after collection of a *new* data). Easy to memorize: Posterior probability is proportional to likelihood multiplied by prior probability.

Maximal Likelihood Estimate for MVN

Theorem

If one have N samples $x_i \backsim \mathcal{N}(\mu, \Sigma)$ then the maximal likelihood estimate (MLE) for the parameters is given by

$$\begin{aligned} \hat{\mu} &= \frac{1}{N}\sum_{i=1}^{N} x_i \triangleq \bar{x} \\ \hat{\Sigma} &= \frac{1}{N}\Big(\sum_{i=1}^{N} x_i x_i^T\Big) \end{aligned}$$

Gaussian Mixture Model

- **LVM** - latent variable models
- **Mixture of Gaussians**

$$p(x_i|\theta) = \sum_{k=1}^{K} \tau_k \mathcal{N}(x_i|\mu_k, \Sigma_k).$$

  where τ_k are the mixing weights, μ_k are the means and Σ_k are the covariance matrices for each base distribution of the mixture.

- Applications:
  - Black box density model to be used in data compression,outlier detection etc.
  - Clustering. Fit the mixture model and then compute $p(z_i = k|x, \theta)$ - The posterior probability that point i belongs to cluster k.

reminder: Bayes rule

- NB! This is short reminder of Bayes theorem.
- We will return to Bayesian theory in the next chapter.
- Let A and B are two events, whereas $P(B) \neq 0$. Then

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- Computational example.

Mixture models for clustering

- The posterior probability that point i belongs to cluster k is referred as the *responsibility of cluster k for point i*. According to Bayes rule:

$$r_{i,k} = p(z_i = k|x_i, \theta) = \frac{p(z_i = k|\theta)p(x_i|z_i = k, \theta)}{\sum_{k'=1}^{K} p(z_i = k'|\theta)p(x_i|z_i = k', \theta)}$$

- This procedure is referred as *soft clustering*. NB! In the mixture case we never observe variables z_i.
- Link to *hard clustering* using MAP estimate

$$z_i^* = \arg\max_k r_{i,k} = \arg\max_k \log p(x_i|z_i = k, \theta) + \log p(z_i = k|\theta).$$

- Presence of latent variables makes complicated to compute ML estimates. Introduce negative log likelihood function.

$$NLL(\theta) = -\frac{1}{N}\log p(\mathcal{D}|\theta).$$

- Let x be the observed variables and z_i be the hidden or missing variables. The goal is to maximize the log likelihood of the observed data.

$$\ell(\theta) = \sum_{i=1}^{N}\log p(x_i|\theta) = \sum_{i=1}^{N}\log\Big[p(x_i, z_i|\theta)\Big].$$

- Complete data log likelihood could not be computed because z_i is unknown.

$$\ell_C(\theta) = \sum_{i=1}^{N}\log p(x_i, z_i|\theta).$$

- Expected complete data log likelihood

$$\begin{aligned} Q(\theta, \theta^{t-1}) &= \mathbb{E}[\ell_c(\theta)|\mathcal{D}, \theta^{t-1}] \\ &= \sum_i\sum_k r_{i,k}\log\tau_k + \sum_i\sum_k r_{r,k}\log p(x_i|\theta_k). \end{aligned}$$

EM for GMM

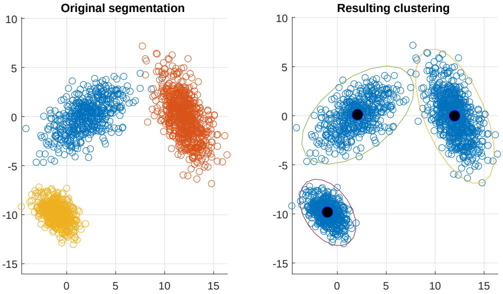
- E step:

$$r_{i,k} = \frac{\tau_k p(x_i|\theta_k^{(t-1)})}{\sum_{k'} \tau_{k'} p\left(x_i|\theta_{k'}^{(t-1)}\right)}$$

- M step: Optimize Q with respect to the θ and τ.

$$\begin{aligned}
\tau_k &= \frac{\sum_i r_{i,k}}{N} \\
\mu_k &= \frac{\sum_i r_{i,k} x_i}{r_k} \\
\Sigma_k &= \frac{\sum_i r_{i,k}(x_i - \mu_k)(x_i - \mu_k)^T}{r_k} = \frac{\Sigma_i r_{i,k} x_i x_i^T}{r_k} - \mu_k \mu_k^T
\end{aligned}$$

Example


Original segmentation
Resulting clustering
10
5
0
-5
-10
-15
0
5
10
15

Clustering overview

- EM estimates the parameters of mixture.
- EM may be referred as parametric method. Model is described by the parameters of clusters.
- How model is described for other clustering techniques? Representative? Hierarchical? Density-based?
- What is clustering model?

Related topics

- Self organizing maps, will be discussed later (together with Neural Networks).
- Outlier analysis.

Exercises for self practice

Please note this is not a mandatory Home Assignment, nevertheless some or all of the exercises may be included into Home Assignments.

Exercises

- Implement EM algorithm.
- Compare performance of your implementation of EM algorithm to the performance of k-means.
- Could you formally verify if given set is gaussian? (question to refresh yor knowledge of probability and statistics)

Cluster Purity. NB! Not unsupervised any more!!!

- Let m_{ij} represent the number of data points from class (ground-truth cluster) i that are mapped to (algorithm determined) cluster j.
- Denote number of data points in true cluster i are by N_i,the number of data points in algorithm-determined cluster j by M_j.

$$N_i = \sum_{j=1}^{k_d} m_{ij}; \qquad M_j = \sum_{i=1}^{k_t} m_{ij};$$

- For a given algorithm-determined cluster j, the number of data points P_j in its dominant class is: $P_j = \max_i m_{ij}$.
- Purity index is defined

$$P_a = \frac{\sum_{j=1}^{k_d} P_j}{\sum_{j=1}^{k_d} M_j}.$$

Gini index

- Gini index for algorithm determined cluster j is defined:

$$G_j = 1 - \sum_{i=1}^{k_t} \left(\frac{m_{ij}}{M_j} \right)^2.$$

- Average Gini index is defined as follows:

$$G = \frac{\displaystyle\sum_{j=1}^{k_d} G_j M_j}{\displaystyle\sum_{j=1}^{k_d} M_j}.$$