Machine Learning, Lecture 5

S. Nõmm

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

05.03.2015

Influence of the hyper parameters

- Distance function and number of clusters.
- Distance between two sets.
- Density and neighbourhood defining parameters.

An open question: how to validate clustering results?

Different approaches to clustering

- Representative-Based Algorithms
  - The k-Means Algorithm.
  - The Kernel k-Means Algorithm
  - The k-Medians Algorithm
  - The k-Medoids Algorithm
- Hierarchical Clustering Algorithms
  - Bottom-Up Agglomerative Methods
  - Top-Down Divisive Methods
- Density and grid based techniques
  - Grid based clustering
  - Density based clustering
- Probabilistic clustering

Cluster Validation

- Internal Cluster Validation
  - Sum of square distances to centroids;
  - Intracluster to intercluster distance ratio;
  - Silhouette coefficient;
  - Probabilistic measure;
- External Cluster Validation, used when ground truth information is available.
  - Confusion matrix;
  - Cluster purity;
  - Gini index;

Bottom-Up Agglomerative Methods

Hyper parameters: distance between two clusters

- **Step 1:** Consider each point of the data set as the cluster
- **Step 2:** Compute $n \times n$ matrix representing distances between each pair of clusters.
- **Step 3:** Select two closest clusters and merge them
- **Step 3:** If convergence criterion not satisfied return to Step 2

Group-Based Statistics

- Best (single) linkage
- Worst (complete) linkage
- Group-average linkage
- Closest centroid
- Variance based criterion
- Ward's method

Grid - based methods

Hyper parameters: range r defines the grid, τ defines the liminal density

- **Step 1:** Discretize each dimension of the dataset into the r ranges
- **Step 2:** Find the cells with the density level higher or equal to τ
- **Step 3:** Define clusters as the sets of adjacent cells

Density - based methods

Definition

Data point d is defined as a **core point***, if for each density τ there exists positive ε_τ such that ε_τ-neighborhood of d contains at least τ data points.*

Definition

A data point d is said to be a **border point***, if for each density τ there exists positive ε_τ such that ε_τ-neighborhood of d contains at least two data points whereas one of them is core point.*

Definition

A data point that is neither a core point nor a border point is defined as a **noise point***.*

DBSCAN

- Determine core, border and noise points of $\mathcal{D}$ at level (ε, τ);
- Create graph in which core points are connected if they are within Eps of one another;
- Determine connected components in graph;
- Assign each border point to connected component with which it is best connected;
- Return points in each connected component as a cluster;

Cluster Purity

- 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:

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

Mixture models

Let $z_i = \{1, \ldots, K\}$, - discrete latent states.

$$\begin{aligned} p(z_i) &= \mathsf{Cat}(\pi) \\ \mathcal{L}(x_i \mid z_i = k) &= p_k(x_i) \end{aligned}$$

Overall model is known as *Mixture model* (we are mixing together K base distributions)

$$p(x_i \mid \theta) = \sum_{k=1}^{K} \pi_k p_k(x_i \mid \theta)$$

where mixed weights π_k satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$

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.

Parameter estimation for Gaussian Mixture Models

- The goal is to estimate parameters:
  $\boldsymbol{\pi}, \boldsymbol{\mu_k}, \Sigma_k, \quad k = 1, \ldots, K$
- The log-likelihood function of GMM is

$$\log p\big(\boldsymbol{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\big) = \sum_{i=1}^{n} \log\Big(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x_i} \mid \boldsymbol{\mu_k}, \Sigma_k)\Big)$$

- Possible problems:
  - Unidentifiability: K-component mixture has $K!$ possible labeling therefore there is no unique maximal likelihood estimate and in turn no unique maximum a posterior estimate.
  - Summation inside the logarithm ... .

Observe the following

- The knowledge of component parameters and mixing proportions allows to compute the probability that the component k responsible [1] for the i-th point $p(z_i = k \mid \boldsymbol{x_i}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$.
- The knowledge of the responsibilities allows to compute the estimates for the mixing coefficients π_k.
- The knowledge of responsibilities and mixing coefficients allows to compute the estimates for component means μ_k and variances Σ_k

This leads the idea of two step iterative algorithm:

- **Step E:** Inferring the missing values given the parameters.
- **Step M:** Optimization of the parameters given the "filled data".

[1] Responsibility of the cluster k for point i is the posterior probability that point i belongs to cluster k, $p(z_i = k \mid \boldsymbol{x_i}, \boldsymbol{\theta})$

Expectation - Maximization

Expectation - Maximization (EM):

- Let x_i denote the visible observed values in case i, and z_i - hidden or missing variables. The goal is to maximize the $\log$ likelihood of the observed data:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \log p(x_i \mid \theta) = \sum_{i=1}^{N} \log \Big[\sum_{z_i} p(x_i, z_i \mid \theta) \Big]$$

- Way around the problem with the sum under the log. Define the complete data log likelihood as is follows

$$\mathcal{L}_c(\theta) = \sum_{i=1}^{N} \log p(x_i, z_i \mid \theta)$$

Note, that this could not be computed due to the fact that z_i are unknown.

EM

- Define expected complete data log likelihood:

  $$Q(\theta, \theta^{t-1}) = \mathbb{E}[l_c(\theta) \mid \mathcal{D}, \theta^{t-1}].$$

  here t is the iteration number. Q will be referred as *auxiliary function*.
- **E** step computes the latent values needed to compute $Q(\theta \mid \theta^{t-1})$.
- **M** step optimizes Q with respect to θ.

  $$\theta^t = \arg\max_\theta Q(\theta, \theta^{t-1})$$

EM -algorithm

- Auxiliary function:

$$Q(\theta, \theta^{t-1}) = \sum_i \sum_k r_{i,k} \log \pi_k + \sum_i \sum_k r_{i,k} \log p(\boldsymbol{x}_{\boldsymbol{i}} \mid \theta_k).$$

- **E step:** compute the responsibilities $r_{i,k}$ for each i and k:

$$r_{i,k} = \frac{\pi_k p(\boldsymbol{x}_{\boldsymbol{i}} \mid \theta_k^{t-1})}{\sum_{k'} \pi_{k'} p(\boldsymbol{x}_{\boldsymbol{i}} \mid \boldsymbol{\theta}_{k'}^{t-1})}.$$

EM -algorithm

- Optimize Q with respect to $\boldsymbol{\pi}, \boldsymbol{\mu}_k, \Sigma_k$.
- $$\pi_k = \frac{1}{N}\sum_i r_{i,k} = \frac{r_k}{N}$$

  where $r_k = \sum_i r_{i,k}$
- Derive **M step** for the μ_k and Σ_k

$$\mathcal{L}(\mu_k, \Sigma_k) = -\frac{1}{2}\sum_i r_{i,k}[\log \mid \Sigma_k \mid + (x_i - \mu_k)^T \sigma_k^{-1}(x_i - \mu k)]$$

$$\begin{aligned} \mu_k &= \frac{\sum_i r_{i,k} x_i}{r_k} \\ \Sigma_k &= \frac{\sum_i r_{i,k} x_i x_i^t}{r_k} - \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T \end{aligned}$$