MO435 — Probabilistic Machine Learning
INSTITUTO DE COMPUTAÇÃO — UNICAMP

Clustering

Lesson No. 13

Alana de Santana Correia [*]

1 Introduction

Clustering is the process of grouping similar objects. There are several types of clustering algorithms; they are similarity-based clustering, feature-based clustering, flat clustering, and hierarchical clustering.

In similarity-based clustering, the input is a distance matrix D $N \times N$. In feature-based clustering, the input is a matrix of $N \times D$ features. In flat clustering data split occurs in disjoint sets, and hierarchical clustering, the division of data occurs in the form of a tree.

The main advantage of similarity-based clustering is that it is easy to include kernel functions and specific similarity domains. However, feature-based clustering applies successfully to noisy data. Flat clustering methods have speed as an advantage. However, hierarchical methods are more useful, most techniques are deterministic and do not require the specification of count clusters (k), while flat methods are sensitive to initial conditions and require some model selection method for k. The following sections of this summary present some essential information regarding clustering.

2 Measuring (dis)similarity

The most common way to define dissimilarity between objects is in terms of the dissimilarity of their attributes:

$$\Delta(x_i, x_{i'}) = \sum_{j=1}^{D} \Delta_j(x_{ij}, x_{i'}) \tag{1}$$

Some common attribute dissimilarity functions are as follows:

- Squared (Euclidean) distance (uses for ordinal variables):

$$\Delta(x_i, x_{i'j}) = (x_{ij}, x_{i'j})^2 \tag{2}$$

- l_1 distance (uses for ordinal variables):

$$\Delta(x_i, x_{i'j}) = |x_{ij}, x_{i'j}| \tag{3}$$

- hamming distance (uses for categorical variables):

$$\Delta(x_i, x_i) = \sum_{j=1}^{D} \mathbb{I}(x_{ij} \neq x_{i'j}) \tag{4}$$

3 Evaluating the output of clustering methods

Evaluate clustering methods is challenging to assess clustering methods. However, there are some useful metrics: purity, rand index, and mutual information.

[*] RA: 229999 - a229999@dac.unicamp.br

3.1 Purity

Purity varies between 0 (bad) and 1 (good), given as follows:

$$purity = \sum_i \frac{N_i}{N} p_i \tag{5}$$

where, $p_i = max_j p_{ij}$, $N_i = \sum_{j=1}^{C} N_{ij}$, $p_{ij} = \frac{N_{ij}}{N_i}$. N_{ij} is the number of objects in cluster i that belong to class j, N_i is the total number of objects in cluster i, p_{ij} is the empirical distribution over class labels for cluster i.

3.2 Rand index

Rand index can be interpreted as the fraction of clustering decisions that are correct. Let $U = u_1, ..., u_R$ and $V = v_1, ..., V_C$ be two different partitions of the N data points, i.e., two different (flat) clusterings:

$$R = \frac{TP + TN}{TP + FP + FN + TN} \tag{6}$$

where $0 \leq R \leq 1$, *TP* is the number of pairs that are in the same cluster in both U and V (true positives); *TN* is the number of pairs that are in the different clusters in both U and V (true negatives); *FN* is the number of pairs that are in the different clusters in U but the same cluster in V (false negatives); and *FP* is the number of pairs that are in the same cluster in U but different clusters in V (false positives).

3.3 Mutual information

Another way to measure cluster quality is to compute the mutual information between U and V. Then we have:

$$\mathbb{I}(U,V) = \sum_{i=1}^{R} \sum_{j=1}^{C} p_{UV}(i,j) log \frac{p_{UV}(i,j)}{p_U(i) p_V(j)} \tag{7}$$

where $p_{UV}(i,j) = \frac{|u_i \cap v_j|}{N}$ is the probability that a randomly chosen object belongs to cluster u_i in U and v_j in V, $p_U(i) = \frac{|u_i|}{N}$ is the be the probability that a randomly chosen object belongs to cluster u_i in U, define $p_V(i) = \frac{|v_i|}{N}$ similarly.

4 Dirichlet process mixture models

The simplest approach to flat clustering is the finite mixture model. The main problem with finite mixture models is how to choose the number of components K. In this summary, we discuss **infinite mixture models**, in which we do not impose any a *priori* bound on K. To do this, we will use a non-parametric prior based on the Dirichlet process (DP). This allows the number of clusters to grow as the amount of data increases.

4.1 From finite to infinite mixture models

The usual representation is as follows:

$$p(x_i | z_i = k, \theta) = p(x_i | \theta_k) \tag{8}$$

$$p(z_i = k | \pi) = \pi_k \tag{9}$$

$$p(\pi | \alpha) = Dir(\pi | (\alpha / K) 1_K) \tag{10}$$

4.1.1 The Dirichlet process

The Dirichlet process is the generalization of the infinite dimension of the Dirichlet distribution. In the same way that the Dirichlet distribution is the a prior conjugate for categorical distribution, the Dirichlet process is the a prior conjugate for discrete, non-parametric, and infinite distributions. A particularly important application of Dirichlet's processes is as an a prior probability distribution in infinite mixture models.

Specifically, suppose that the generation of values $X_1, X_2 ..., X_N$ can be simulated by the following algorithm:

1. Input: **H** (base distribution), α (escalation parameter).

2. get X_1 from the H distribution.

3. while n ≤ N:

   With probability $\frac{\alpha}{\alpha+n-1}$, get X_n from H.

   With probability $\frac{n_x}{\alpha+n-1}$, configure $X_n = x$, where n_x is the number of previous observations $X_j, j < n$, such that $X_j = x$.

A Dirichlet process is a distribution over probability measures $G : \theta \rightarrow \mathbb{R}^+$, where we require $G(\theta) \geq 0$ and $int_\theta G(\theta) d\theta = 1$. The DP is defined implicitly by the requirement that $(G(T_1), ..., G(T_K))$ has a joint Dirichlet distribution

$$Dir(\alpha H(T_1), ..., \alpha H(T_k)) \tag{11}$$

for any finite partition $(T_1, ..., T_K)$ of θ. If this is the case, we write $G \sim DP(\alpha, H)$.

4.1.2 Applying Dirichlet processes to mixture modeling

The DP is not particularly useful as a model for data directly, since data vectors rarely repeat exactly. We can write the model as follows:

$$\pi \sim GEM(\alpha) \tag{12}$$

$$z_i \sim \pi \tag{13}$$

$$\theta_k \sim H(\lambda) \tag{14}$$

$$x_i \sim F(\theta_{z_i}) \tag{15}$$

where G is now a random draw of an unbounded number of parameters θ_k from the base distribution **H**, each with weight π_k. Each data point x_i is generated by sampling its own "private" parameter θ_i from G. As we get more and more data, it becomes increasingly likely that θ_i will be equal to one of the $\theta'_k s$ we have seen before, and thus xi will be generated close to an existing datapoint.

4.2 Affinity propagation

Affinity propagation is an approach to the problem of local minimums in the *k-medoids* algorithm. The goal is to maximize the following function:

$$S(c) = \sum_{i=1}^{N} s(i, c_i) + \sum_{k=1}^{N} \delta_k(c) \tag{16}$$

The first term measures the similarity of each point to its centroid. The second term is a penalty term:

$$\theta_k(c) = \begin{cases} -\propto & if c_k \neq k \quad but \quad \exists i : c_i = k \\ 0 & otherwise \end{cases} \tag{17}$$

4.3 Spectral Clustering

An alternative view of clustering is in terms of graph cuts. The idea is we create a weighted undirected graph W from the similarity matrix S, typically by using the nearest neighbors of each point:

$$\frac{1}{2}\sum_{k=1}^{K} W(A_k, \overline{A_k}) \tag{18}$$

where $A_1, ..., A_K$ is a minimization criterion.

4.4 Hierarchical Clustering

There are two main approaches to the hierarchical grouping: bottom-up or cluster grouping and top-down or dividing grouping. Both methods take a matrix of dissimilarity between objects as input. In the bottom-up approach, the merge occurs in the most similar groups in each step. In the top-down approach, groups the division occurs using several different criteria.

Agglomerative clustering starts with N groups, each initially containing one object, and then at each step it merges the two most similar groups until there is a single group, containing all the data. The merging process can be represented by a binary tree, called a dendrogram. Divisive clustering starts with all the data in a single cluster, and then recursively divides each cluster into two daughter clusters, in a top-down fashion.

Divisive clustering is less popular than agglomerative clustering, but it has two advantages:

- it can be faster.
- the splitting decisions are made in the context of seeing all the data, whereas bottom-up methods make myopic merge decisions.

5 Conclusion

This summary presents the concepts of the mixture model. It is a probabilistic model for representing the presence of sub-populations within an overall population, without requiring that an observed data set should identify the sub-population to which an individual observation belongs. Some ways of implementing mixture models involve steps that attribute postulated sub-population-identities to individual observations (or weights towards such sub-populations), in which case these can be regarded as types of unsupervised learning or clustering procedures. There are finite and infinite mixture models.

The infinite mixture model has several advantages over its finite counterpart: in many applications, it may be more appropriate not to limit the number of classes; the number of represented classes is automatically determined; the use of MCMC effectively avoids local minimal which plague mixtures trained by optimisation based methods; and it is much simpler to handle the infinite limit than to work with finite models with unknown sizes.