CLUSTERING (Ch 10)

11/05/21

Aim of clustering: divide the dataset into subgroups / clusters
↓
homogeneous
↓
- similarity within groups
- dissimilarity between groups

Unsupervised learning (unknown groups)

Many applications:

1. Genomics → Find unknown subtypes of breast cancer
   → Find groups of genes / proteins with similar functions
2. Marketing (Segmentation) → Find clusters of shoppers in terms of their shopping history
3. Communities in social networks

MANY METHODS

1. K-means clustering
2. Hierarchical clustering

) most popular

3. Mixture modelling : assume an underlying probability distribution

4. Kernel density clustering : clusters identified by a high density of points separated by areas of low density

K-MEANS CLUSTERING

↑ number of clusters fixed

Objective Split the dataset D into K clusters, $C_1, \dots, C_K$, that form a partition ↑ with meaningful clusters

$$C_1 \cup \dots \cup C_K = D$$
$$C_i \cap C_j = \emptyset \quad \forall i \neq j$$

So we need to define a within-cluster similarity (ideally associated to between-cluster dissimilarity)

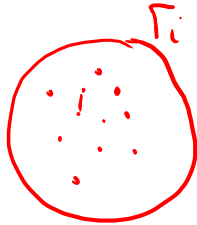
WITHIN-CLUSTER VARIATION

$$(C_1, \dots, C_K) = \underset{\substack{\Gamma_1, \dots, \Gamma_K \\ \text{partitions} \\ \text{of } D}}{\operatorname{argmin}} \sum_{i=1}^{K} W(\Gamma_i)$$

with

$$W(\Gamma_i) = \frac{1}{2|\Gamma_i|} \sum_{\substack{\underline{x} \in \Gamma_i \\ \underline{y} \in \Gamma_i}} \sum_{\ell=1}^{p} (x_\ell - y_\ell)^2$$

$\|\underline{x} - \underline{y}\|^2$ (Euclidean distance)



with $\underline{x} = (x_1, \dots, x_p)$, $\underline{y} = (y_1, \dots, y_p)$

Why variation?

We can show that

$$W(\Gamma_i) = \sum_{x \in \Gamma_i} \sum_{e=1}^{p} (x_e - \bar{x}_{ie})^2 \quad \text{with} \quad \bar{x}_{ie} = \frac{\sum_{x \in \Gamma_i} x_e}{|\Gamma_i|}$$

$\bar{x}_{ie}$: sample mean of X_e in cluster Γ_i

i.e. within-cluster variance.

Proof: ($p = 1$)

$$\frac{1}{|\Gamma_i|} \sum_{x \in \Gamma_i} \sum_{y \in \Gamma_i} (x-y)^2 = \frac{1}{|\Gamma_i|} \sum_{x \in \Gamma_i} \sum_{y \in \Gamma_i} (x^2 - 2xy + y^2)$$

$$= \frac{1}{|\Gamma_i|} \sum_{x \in \Gamma_i} \left[|\Gamma_i| x^2 - 2x \underbrace{\sum_{y \in \Gamma_i} y}_{|\Gamma_i| \bar{x}_i} + \sum_{y \in \Gamma_i} y^2 \right] =$$

$$= \frac{1}{|\Gamma_i|} \left[|\Gamma_i| \sum_{x \in \Gamma_i} x^2 - 2 |\Gamma_i| \bar{x}_i \underbrace{\sum_{x \in \Gamma_i} x}_{|\Gamma_i| \bar{x}_i} + |\Gamma_i| \sum_{y \in \Gamma_i} y^2 \right]$$

$$= 2 \sum_{x \in \Gamma_i} x^2 - 2 |\Gamma_i| \bar{x}_i^2$$

$$= 2 \left(\sum_{x \in \Gamma_i} x^2 - |\Gamma_i| \bar{x}_i^2 \right) = 2 \left(\sum_{x \in \Gamma_i} (x - \bar{x}_i)^2 \right) \quad \blacksquare$$

Minimizing within-cluster variance $\Leftrightarrow$ Maximising between-cluster variation

In order to see this:

① Since the overall sum of squared distances in the data is fixed, minimizing within-cluster distances means maximising between-cluster distances

② This is connected to the variance decomposition formula:

$$\mathrm{Var}(X) = \underbrace{E(\mathrm{Var}(X|C))}_{\text{average of variances within clustering}} + \underbrace{\mathrm{Var}(E(X|C))}_{\text{variance of averages within groups} \rightarrow \text{distance between centroids}}$$

with $C = \{1, \dots, K\}$ cluster assignment

(unexplained variance) (explained variance)

IMPLEMENTATION

Far too many possible clusterings $\rightarrow \approx K^n$

ALGORITHM (Fixed K)

① **Random Initialization** : randomly assign each observation to one of K clusters

Typically :

① Uniformly for each observation (most common)

② Randomly in space

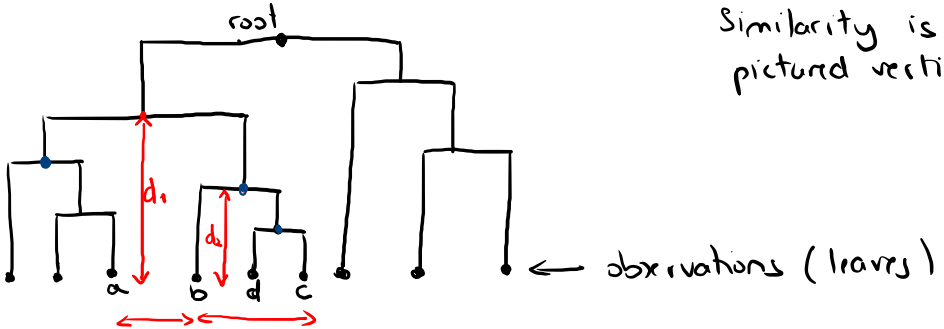
② Iterate until convergence :

a) For each cluster, compute centroid $\bar{x}_i$

b) Assign each observation $1, \ldots, n$ to the cluster whose centroid is closest (in Euclidian distance)

HIERARCHICAL CLUSTERING

- A family of methods
- There is no prior assumption on the number of clusters
- The output consists of a hierarchy of nested clusters, represented by a dendogram
- It can be built bottom-up (agglomerative approach) or top-down (divisive approach)
- Typical dendogram:



Similarity is pictured vertically

Two main ingredients:

① A measure of similarity between points

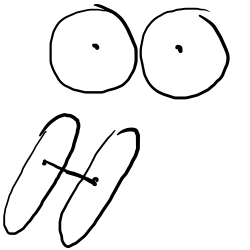
② A linkage criterion: a measure of distance between clusters

The results will depend on these choices, which can be also driven by the data-domain

CHOICE OF METRIC

Different choices:

1. Euclidean distance (L_2 distance): $\sum_{\ell=1}^{p} (x_\ell - y_\ell)^2$
2. Taxicab distance (L_1 distance): $\sum_{\ell=1}^{p} |x_\ell - y_\ell|$
3. Max (or L_∞ distance): $\max_{\ell} |x_\ell - y_\ell|$
4. Mahalanobis distance: $\sqrt{(\underline{x}-\underline{y})^t \Sigma^{-1} (\underline{x}-\underline{y})}$
5. Correlation: $\mathrm{Cor}(\underline{x}, \underline{y})$



Text mining

- Levenshtein distance: # edits to mutate one string into another (between words)

  CUP → CAP

  ↓ one edit

- Distance between documents:

$$d(\underline{x}, \underline{y}) = \cos\left(t(\underline{x}), t(\underline{y})\right) = \frac{t(\underline{x})^t\, t(\underline{y})}{\|t(\underline{x})\|\,\|t(\underline{y})\|}$$

where

$t(\underline{x}) = \left(\underline{tf\text{-}idf}(w, \underline{x}, C)\right)$ across a list of words w

$$tf(w, \underline{x}) = \frac{\#\text{ occurrences of word } w \text{ in a document } \underline{x}}{\#\text{ words in a document}}$$

$$idf(w, C) = \log\left(\frac{\#\text{ documents in corpus } C}{\#\text{ documents in } C \text{ containing } w}\right)$$

$$tf\text{-}idf(w, \underline{x}, C) = tf(w, \underline{x}) \cdot idf(w, C)$$

CHOICE OF LINKAGE

Common choices:

① Single linkage

chosen metric between observations

$$d_{SL}(C_1, C_2) = \min_{\substack{x \in C_1 \\ y \in C_2}} d(x, y)$$

→ Tends to produce long thin clusters (chain)

② COMPLETE LINKAGE

$$d_{CO}(C_1, C_2) = \max_{\substack{x \in C_1 \\ y \in C_2}} d(x, y)$$

→ Compact clusters but very sensitive to outliers

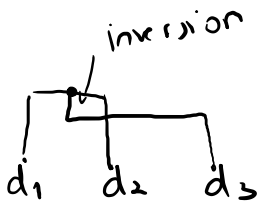
③ Average Linkage

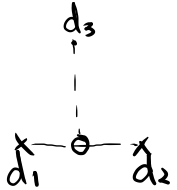
$$d_{AL}(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{\substack{x \in C_1 \\ y \in C_2}} d(x, y)$$

④ Centroid Linkage

$$d_{CE}(C_1, C_2) = d(\bar{x}_1, \bar{x}_2) \quad \text{with } \bar{x}_i \text{ centroid of cluster } C_i$$

→ Non monotone





IMPLEMENTATION

AGGLOMERATIVE

- Start with each observation in one cluster
- At each step join observations / clusters that are closer to each other
- Stop after n steps

DIVISIVE

- Start with everything in one cluster C
- Pick observation most dissimilar from the rest and put it into a new cluster $\tilde{C}$
- Move any observation in C that is closest to $\tilde{C}$ to $\tilde{C}$ $\rightarrow$ Two clusters
- Repeat until you have one observation in each cluster.

HOW MANY CLUSTERS?

Several choices:

1. "Lazy": Cut tree at pre-defined level of cluster similarity
2. "Lazy-K": fix number K of clusters
3. LARGEST GAP: cut the tree before the joining that decreases the quality of the clustering the most (a bit like a scree plot)

IN PRACTICE

1. Careful with scaling
2. Careful with outliers
3. Only choose clustering as a pre-processing / exploratory analysis step