MAT388E:Data Analysis in Fundamental Sciences

Fall23-Lecture 14: Clustering

Gül İnan

İstanbul Technical University

Learning Objectives

- Clustering
  - K-Means Clustering

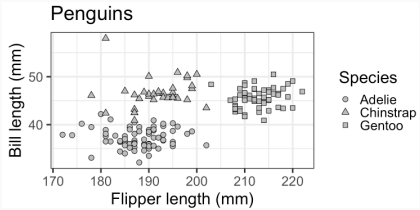
Clustering

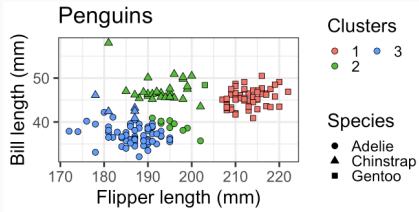
Clustering

- **Clustering** refers to a **very broad set of techniques** where the aim is to group data into several **homogeneous subgroups** or **clusters**.

Example: Clustering penguins

- Consider the penguins data set.
- In this data set, we can be interested in identifying **subgroups of penguins** which are similar in terms of their bill length and flipper length.





Clustering

- When we cluster the observations of a data set, we seek to partition them into distinct groups such that:
  - **Observations within each group** are **quite similar** to each other and
  - **Observations in different groups** are quite **different** from each other.
- Since grouping data points into clusters is done with **no observed labels**, **clustering** is an **unsupervised learning** technique.

Distance as a measure of dissimilarity

- Let $\{\mathbf{x}_i\}_{i=1}^n$ be the n observed data of p-dimensional feature vector, where $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T$.
- We expect to see that the **data within each cluster is similar**.
- In order to **quantify the similarity**, we define a **dissimilarity measure** for the jth feature of ith observation and jth feature of i^*th observation:

$$d_j(x_{ij}, x_{i^*j}),$$

- where $j = 1, \dots, p$ and $i = 1, \dots, n$.

Distance functions

- For **continuous variables**, the most common choice for distance function $d(.)$ is the **squared Euclidean distance**:

$$d_j(x_{ij}, x_{i^*j}) = \left\| x_{ij} - x_{i^*j} \right\|^2$$

- where $j = 1, \ldots, p$ and $i = 1, \ldots, n$.
- Since clustering is a distance-based algorithm, scaling the numerical features are suggested.

Distance functions

- For **discrete variables**, the most common choice for distance function $d(.)$ is the **Hamming distance**:

$$d_j(x_{ij}, x_{i^*j}) = I(x_{ij} \neq x_{i^*j})$$

- where $j = 1, \ldots, p$ and $i = 1, \ldots, n$.

Overall dissimilarity

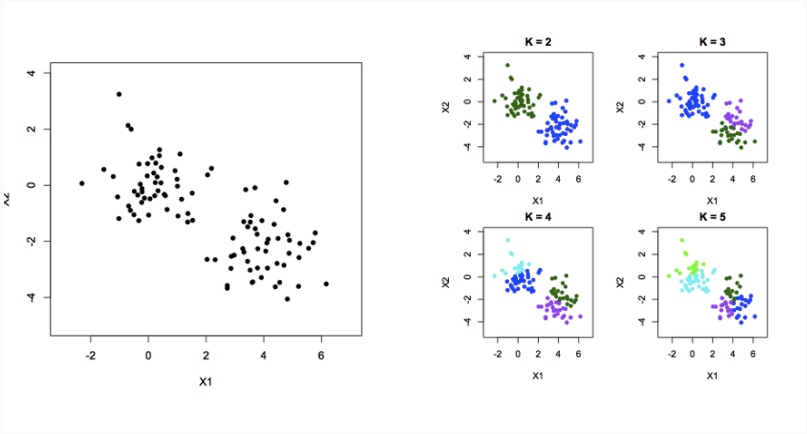
- Then, the **overall dissimilarity** between any two p-dimensional points $\mathbf{x}_i = (x_{i1}, \dots, x_{ij}, \dots, x_{ip})^T$ and $\mathbf{x}_{i^*} = (x_{i^*1}, \dots, x_{i^*j}, \dots, x_{i^*p})^T$ can be calculated as:

$$D(\mathbf{x}_i, \mathbf{x}_{i^*}) = \sum_{j=1}^{p} d_j(x_{ij}, x_{i^*j}).$$

Clustering

- Suppose that we want to **separate the data into** K clusters.
- Let $C_1, ..., C_k, ..., C_K$ denote sets containing the indices of the observations in each cluster.
- For example, if the ith observation is in the kth cluster, then $i \in C_k$.
- These sets satisfy two properties:
  - $C = C_1 \cup C_2 \cup ... \cup C_K = \{1, ..., n\}$. In other words, **each observation belongs to at least one of the** K clusters.
  - $C_k \cap C_{k^*} = \emptyset$ for all $k \neq k^*$. In other words, the clusters are non-overlapping: **no observation belongs to more than one cluster**.

Cluster Analysis



Within-Cluster Distance for kth cluster

- Let $W(C_k)$ is the **within-cluster distance for the cluster** C_k which is the sum of all pairwise distances within the kth cluster, divided by the total number of observations in the cluster as follows:

$$\begin{aligned} W(C_k) &= \frac{1}{2|C_k|} \sum_{i \in C_k} \sum_{i^* \in C_k} D(\mathbf{x}_i, \mathbf{x}_{i^*}), \\ &= \frac{1}{2|C_k|} \sum_{i, i^* \in C_k} D(\mathbf{x}_i, \mathbf{x}_{i^*}). \end{aligned}$$

- Here, **one half** is used because $D(\mathbf{x}_i, \mathbf{x}_{i^*})$ and $D(\mathbf{x}_{i^*}, \mathbf{x}_i)$ are both counted in the expression above.

Total Within-Cluster Distance

- $W(C_k)$ measures the amount by which the observations within a cluster differ from each other.
- Then, we define the overall (total) **within-cluster distance** as the sum over all clusters, which is given by:

$$\begin{aligned} W(C) &= \sum_{k=1}^{K} W(C_k) \\ &= \sum_{k=1}^{K} \frac{1}{2|C_k|} \sum_{i,i^* \in C_k} D(\mathbf{x}_i, \mathbf{x}_{i^*}). \end{aligned}$$

- Smaller $W(C)$ is better.

Minimizing Total Within-Cluster Distance

- The idea behind **clustering** is that a **good clustering** is one for which the **total within-cluster distance** is **as small as possible**.
- Hence, we want to solve the following optimization problem:

$$\underset{C_1,\ldots,C_K}{minimize} \quad W(C).$$

- In words, this means that we want to partition the data points into clusters such that the **total within-cluster variation** summed over all K clusters is **as small as** possible.

HW: Stirling Numbers Second Kind

- All possible assignments of n data points into K different groups can be calculated through:

$$A(n, K) = \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \binom{K}{k} k^n.$$

- See: See page 91, Jain and Dubes (1998), "Algorithms for Clustering Data".
- Note that $A(10, 4) = 34,105$, and $A(25, 4) \approx 5 \times 1013$ ... huge
- So we need a **much simple way**.

K-Means Clustering

K-Means Clustering

- **K-means** clustering, also referred as the *Llyod algorithm* is a simple but popular clustering algorithm, especially, when all features are **quantitative**.
- K-Means clustering aims to find **cluster centers** and **cluster memberships** to minimize the sum of squared Euclidean distances of data points $\mathbf{x}_i$ to their assigned cluster centers.

HW: An Identity

- When the distance $d(.)$ is the squared Euclidean distance, we have the following the identity:

$$W(C_k) = \frac{1}{2|C_k|} \sum_{i,i^* \in C_k} D(\mathbf{x}_i, \mathbf{x}_{i^*})$$
$$= \frac{1}{2|C_k|} \sum_{i,i^* \in C_k} ||\mathbf{x}_i - \mathbf{x}_{i^*}||_2^2 = \sum_{i \in C_k} ||\mathbf{x}_i - \boldsymbol{\mu}_k||_2^2,$$

- where $\mathbf{x}_i = (x_{i1}, ..., x_{ip})^T$ is the data point in the kth cluster C_k and $\boldsymbol{\mu}_k = (\mu_{k1}, ..., \mu_{kp})^T$ is the **centroid** of the kth cluster C_k.
- Informal proof available here: https://stats.stackexchange.com/questions/554052/identity-for-k-means-clustering.

Total Within-Cluster Euclidean Distance

- which implies that total within-cluster Euclidean distance is:

$$W(C) = \sum_{k=1}^{K} W(C_k) = \sum_{k=1}^{K} \sum_{i \in C_k} ||\mathbf{x}_i - \boldsymbol{\mu}_k||_2^2.$$

Minimizing Total Within-Cluster Euclidean Distance

- Then, K-means clustering aims to solve the following optimization problem:

$$(C_1^{opt}, \dots, C_K^{opt}, \mu_1^*, \dots, \mu_K^*) = \underset{C_1,\dots,C_K,\mu_1,\dots,\mu_K}{argmin} \sum_{k=1}^{K} \sum_{i \in C_k} ||\mathbf{x}_i - \boldsymbol{\mu}_k||_2^2.$$

- Namely, find the **cluster memberships** and **cluster centroids** that minimize the sum of the distance of each data point to the centroid of its cluster.
- Finding an optimal solution when minimizing **jointly** over the parameters (cluster centroids and cluster memberships) is an **NP-hard** problem.

Alternating Minimization

- But if we **fix one parameter and minimize over the other parameter**, then the optimization problem becomes easy.
- In other words, we can **alternate** between the cluster memberships and the centers of the clusters.

K-Means Clustering Algorithm

1. Randomly assign a number, from 1 to K, to each of the observations to form the initial cluster memberships $C_1, \ldots, C_K$.
2. **Given the clustering results** $\{C_k\}_{k=1}^K$, find cluster centroid for each C_k through:

$$\frac{\partial}{\boldsymbol{\mu}_k} \sum_{i \in C_k} ||\mathbf{x}_i - \boldsymbol{\mu}_k||_2^2 = 0,$$

- where the optimum solution is:

$$\hat{\boldsymbol{\mu}}_k = (\hat{\mu}_{k1}, ..., \hat{\mu}_{kp})^T = \frac{1}{|C_k|} \sum_{i \in C_k} \mathbf{x}_i = (\bar{x}_{k1}, \ldots, \bar{x}_{kp})^T,$$

- where $k = 1, \ldots, K$ and $\hat{\boldsymbol{\mu}}_k$ is the vector of the p feature means

K-Means Clustering Algorithm

- for the observations in the kth cluster and $\bar{x}_{kj}$ is the average for feature j ($j = 1, ..., p$) in cluster C_k.

3. **Given the estimated cluster centroids** $\{\hat{\boldsymbol{\mu}}_k\}_{k=1}^K$, the optimal clustering membership for ith observation is the one that is closest to its centroid:

$$C_k* = \underset{k \in \{1,...,k*,...,K\}}{argmin} \quad ||\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k||^2.$$

4. Repeat steps 2 and 3 until the clustering results do not change.
5. Try multiple initial values, pick the solution with the best objective value (that's total within-cluster euclidean distance).

An illustration

- Please watch this video for illustration of K-means algorithm:

http://tech.nitoyon.com/en/blog/2013/11/07/k-means/.

Some Practical Issues with Clustering

- K-means algorithm reduces the objective function, **total within-cluster distance**, at each iteration.
- If the cluster membership do not change after a while, we have converged to a **local minimum**.
- Since the algorithm reaches a local optimum and not a global optimum, the results obtained will **depend on the initial** (random) cluster assignment of each observation (step 1).
- Due to this, it's crucial to **run the algorithm many times** and from multiple (random) starting points.
- One should select the best solution, namely, the one where the objective function is the smallest.
- No worries, sckit-learn's KMeans's class does this for us.

Choosing K via the Elbow method

- The **elbow method** is a visual procedure for choosing a correct value for K.
- The idea is to run K-means clustering for a range of K (let's say from k=1 to 10) and calculate the sum of squared distances from each point to its assigned center. This quantity is called as **inertia** in scikit-learn.
- When we plot the number of clusters vs the sum of squared distances, we can choose k_{opt} that defines the **inflection point**: the elbow (separating the forearm from the arm).
- The reason to choose this value of k_{opt} is that for small k values, the sum of squared distances decreases quickly and then starting from some value, the sum of squared distances describes a **plateau**.

Choosing K via the Elbow method

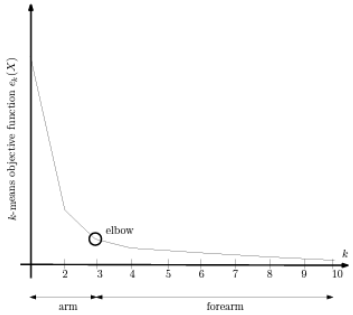


Figure 7.8 Choosing k with the elbow method: the elbow defines the value of k that separates the area of high decrease (the arm) to the plateau area (the forearm).

References

- Fan, J., Li, R., Zhang, C.H., and Zou, H. (2020). Statistical Foundations of Data Science. Chapman and Hall/CRC.
- James, G., Witten, D., Hastie, T., and Tibshirani, R. (2021). An Introduction to Statistical Learning: With Applications in R. New York: Springer.
- https://homepages.inf.ed.ac.uk/rbf/BOOKS/JAIN/Clustering_Jain_Dubes.pdf
- https://amfarahmand.github.io/IntroML-Fall2022/lectures/lec10.pdf
- http://syskall.com/kmeans.js/
- https://colab.research.google.com/github/lcharlin/80-629/blob/master/week7-Unsupervised/Unsupervised_questions.ipynb#scrollTo=rAIs748KVz8B

References

- http://www2.stat.duke.edu/~rcs46/lectures_2015/04-clus1/04-clus1.pdf
- https://astrostatistics.psu.edu/RLectures/stat_learning.pdf
- https://franknielsen.github.io/Clustering/BookChapter-kmeans.pdf
- https://stats.stackexchange.com/questions/554052/identity-for-k-means-clustering