Clustering

Chapter 10 in “Introduction to statistical learning”
Clustering

- Clustering is the art of finding groups in data (Kaufman and Rousseeuw, 1990).

- What is a cluster?
  - Group of objects separated from other clusters
Digression: means and medians

The mean is the minimizer of

$$\arg\min_y \sum_{x \in D} ||x - y||^2$$

Using

$$\arg\min_y \sum_{x \in D} ||x - y||$$

Gives rise to the geometric median, which is more robust to outliers.

Issue: no closed form solution.
Means, medians, medoids

It may be useful to restrict exemplars to be one of the given data points.

Medoid: representative object of a data set or a cluster whose average distance to all the objects in the cluster is minimal

How would we compute the medoid for a set of points?
Clustering

A clustering is a partition of the elements in your dataset into $K$ subsets such that the following holds:

Each observation belongs to a cluster:

$$C_1 \cup C_2 \cup \ldots \cup C_K = \{1, \ldots, n\}$$

The clusters are non-overlapping:

$$C_k \cap C_{k'} = \emptyset \text{ for all } k \neq k'$$
A plausible objective

We would like a clustering to minimize the within-cluster variation.

Let’s assume it is measured via a function $W(C_k)$. So the overall objective is:

$$\text{minimize} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$$

A possible definition of $W(C_k)$:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,j \in C_k} ||x_i - x_j||^2$$
A plausible objective

We would like a clustering to minimize the within-cluster variation. Let’s assume it is measured via a function \( W(C_k) \). So the overall objective is:

\[
\text{minimize}_{\mathcal{C}_1, \ldots, \mathcal{C}_K} \left\{ \sum_{k=1}^{K} W(C_k) \right\}
\]

A possible definition of \( W(C_k) \):

\[
W(C_k) = \frac{1}{|C_k|} \sum_{i,j \in C_k} \|x_i - x_j\|^2
\]

where \( \mu_k \) is the centroid for cluster \( k \)

\[
= 2 \sum_{i \in C_k} \|x_i - \mu_k\|^2
\]
The k-means objective function

Putting it all together:

\[ \min_{C_1, \ldots, C_K} \sum_{k=1}^{K} \sum_{i \in C_k} ||x_i - \mu_k||^2 \]

A problem with this problem: NP-complete.
The k-means algorithm

A heuristic algorithm for solving the problem:

### Algorithm 10.1 K-Means Clustering

1. Randomly assign a number, from 1 to $K$, to each of the observations. These serve as initial cluster assignments for the observations.

2. Iterate until the cluster assignments stop changing:
   
   (a) For each of the $K$ clusters, compute the cluster centroid. The $k$th cluster centroid is the vector of the $p$ feature means for the observations in the $k$th cluster.

   (b) Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).
Example

FIGURE 10.6. The progress of the K-means algorithm on the example of Figure 10.5 with $K = 3$. Top left: the observations are shown. Top center: in Step 1 of the algorithm, each observation is randomly assigned to a cluster. Top right: in Step 2(a), the cluster centroids are computed. These are shown as large colored disks. Initially the centroids are almost completely overlapping because the initial cluster assignments were chosen at random. Bottom left: in Step 2(b), each observation is assigned to the nearest centroid. Bottom center: Step 2(a) is once again performed, leading to new cluster centroids. Bottom right: the results obtained after ten iterations.

As we have seen, to perform K-means clustering, we must decide how many clusters we expect in the data. The problem of selecting $K$ is far from simple. This issue, along with other practical considerations that arise in performing K-means clustering, is addressed in Section 10.3.3.
Local minima

FIGURE 10.7. K-means clustering performed six times on the data from Figure 10.5 with \( K = 3 \), each time with a different random assignment of the observations in Step 1 of the K-means algorithm. Above each plot is the value of the objective (10.11). Three different local optima were obtained, one of which resulted in a smaller value of the objective and provides better separation between the clusters. Those labeled in red all achieved the same best solution, with an objective value of 235.8.

10.3.2 Hierarchical Clustering

One potential disadvantage of K-means clustering is that it requires us to pre-specify the number of clusters \( K \). Hierarchical clustering is an alternative approach which does not require that we commit to a particular choice of \( K \). Hierarchical clustering has an added advantage over K-means clustering in that it results in an attractive tree-based representation of the observations, called a dendrogram. In this section, we describe bottom-up or agglomerative clustering. This is the most common type of hierarchical clustering, and refers to the fact that a dendrogram (generally depicted as an upside-down tree; see...
Running time?

What is the running time per iteration?

**Algorithm 10.1 K-Means Clustering**

1. Randomly assign a number, from 1 to $K$, to each of the observations. These serve as initial cluster assignments for the observations.

2. Iterate until the cluster assignments stop changing:
   
   (a) For each of the $K$ clusters, compute the cluster centroid. The $k$th cluster centroid is the vector of the $p$ feature means for the observations in the $k$th cluster.

   (b) Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).
What is the running time per iteration?

**Algorithm 10.1  \(K\)-Means Clustering**

1. Randomly assign a number, from 1 to \(K\), to each of the observations. These serve as initial cluster assignments for the observations.

2. Iterate until the cluster assignments stop changing:
   
   (a) For each of the \(K\) clusters, compute the cluster *centroid*. The \(k\)th cluster centroid is the vector of the \(p\) feature means for the observations in the \(k\)th cluster.
   
   (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

Typically, converges very quickly (and in fact, guaranteed to converge in a finite number of iterations)
Running time?

Algorithm 10.1 $K$-Means Clustering

1. Randomly assign a number, from 1 to $K$, to each of the observations. These serve as initial cluster assignments for the observations.

2. Iterate until the cluster assignments stop changing:

   (a) For each of the $K$ clusters, compute the cluster centroid. The $k$th cluster centroid is the vector of the $p$ feature means for the observations in the $k$th cluster.

   (b) Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).

Can easily be kernelized.
Dealing with local minima

Run the algorithm multiple times with different initializations.
Initialization

A good initialization can lead to faster convergence to a better optimal solution.
The standard choice: k random data points
More sophisticated approaches:

- Create a collection of subsamples of the data. Cluster the resulting cluster centers using K-means and use for initialization.


- Kmeans++: Choose the first center randomly; subsequent centers are chosen with probability proportional to their distance to the closest center. The default in scikit-learn

Related algorithms

Related algorithms:

K-medoids
Partitioning around medoids (PAM)
Sensitivity to scaling

(Left) On this data 2-means detects the right clusters. (Right) After rescaling the $y$-axis, this configuration has a higher between-cluster scatter than the intended one.
Assumptions behind the model

K-means assumes spherical clusters.

There are probabilistic extensions that address this to some extent.

Probably the most widely used clustering algorithm because of its simplicity, speed, and easy implementation.
Silhouettes

How do we know we have a good clustering?

The Silhouette coefficient is defined for each example

\[ s(x) = \frac{b(x) - a(x)}{\max(b(x), a(x))} \]

a: the mean distance between the example and all other points in the same cluster.

b: the mean distance between the example and all other points in the next nearest cluster.

Silhouettes

The Silhouette coefficient is defined for each example

\[ s(x) = \frac{b(x) - a(x)}{\max(b(x), a(x))} \]

Sort \( s(x) \) and group by cluster:

Figures generated using the scikit-learn silhouette method
Silhouettes

The Silhouette coefficient is defined for each example

\[ s(x) = \frac{b(x) - a(x)}{\max(b(x), a(x))} \]

Sort \( s(x) \) and group by cluster:
Dendrograms

Definition: Given a dataset D, a dendrogram is a binary tree with the elements of D at its leaves. An internal node of the tree represents the subset of elements in the leaves of the subtree rooted at that node.
Hierarchical clustering

Algorithm outline:

Start with each data point in a separate cluster

At each step merge the closest pair of clusters
Hierarchical clustering

Algorithm outline:

Start with each data point in a separate cluster

At each step merge the closest pair of clusters

Need to define a measure of distance between clusters:

A \textit{linkage function} $L : 2^X \times 2^X \rightarrow \mathbb{R}$ calculates the distance between arbitrary subsets of the instance space, given a distance metric $\text{Dis} : X \times X \rightarrow \mathbb{R}$. 

Linkage functions

- **Single linkage**
  - Smallest pairwise distance between elements from each cluster

- **Complete linkage**
  - Largest distance between elements from each cluster

- **Average linkage**
  - The average distance between elements from each cluster

- **Centroid linkage**
  - Distance between cluster means
Dendrograms revisited

Interpretation of the vertical dimension: The distance between the clusters when they were merged (the level associated with the cluster). The leaves have level 0.
Dendrograms revisited

Interpretation of the vertical dimension: The distance between the clusters when they were merged (the level associated with the cluster). The leaves have level 0.
Hierarchical clustering

**Algorithm 10.2 Hierarchical Clustering**

1. Begin with $n$ observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.

2. For $i = n, n-1, \ldots, 2$:
   
   (a) Examine all pairwise inter-cluster dissimilarities among the $i$ clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.

   (b) Compute the new pairwise inter-cluster dissimilarities among the $i - 1$ remaining clusters.

---

**Linkage Description**

- **Complete**
  - Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the largest of these dissimilarities.

- **Single**
  - Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the smallest of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.

- **Average**
  - Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the average of these dissimilarities.

- **Centroid**
  - Dissimilarity between the centroid for cluster A (a mean vector of length $p$) and the centroid for cluster B. Centroid linkage can result in undesirable inversions.

TABLE 10.2. As a summary of the most commonly used types of linkage are generally preferred over single linkage, as they tend to yield more balanced dendrograms. Centroid linkage is often used in genomics, but suffers from a major drawback in that an inversion can occur, whereby two clusters are fused at a height below either of the individual clusters in the dendrogram. This can lead to difficulties in visualization as well as in interpretation of the dendrogram. The dissimilarities computed in Step 2(b) of the hierarchical clustering algorithm will depend on the type of linkage used, as well as on the choice of dissimilarity measure. Hence, the resulting
Linkage matters

Average Linkage

Complete Linkage

Single Linkage

FIGURE 10.12. Average, complete, and single linkage applied to an example data set. Average and complete linkage tend to yield more balanced clusters.

An unusual use of correlation, which is normally computed between variables; here it is computed between the observation profiles for each pair of observations. Figure 10.13 illustrates the difference between Euclidean and correlation-based distance. Correlation-based distance focuses on the shapes of observation profiles rather than their magnitudes.

The choice of dissimilarity measure is very important, as it has a strong effect on the resulting dendrogram. In general, careful attention should be paid to the type of data being clustered and the scientific question at hand. These considerations should determine what type of dissimilarity measure is used for hierarchical clustering.

For instance, consider an online retailer interested in clustering shoppers based on their past shopping histories. The goal is to identify subgroups of similar shoppers, so that shoppers within each subgroup can be shown items and advertisements that are particularly likely to interest them. Suppose the data takes the form of a matrix where the rows are the shoppers and the columns are the items available for purchase; the elements of the data matrix indicate the number of times a given shopper has purchased a given item (i.e. a 0 if the shopper has never purchased this item, a 1 if the shopper has purchased it once, etc.) What type of dissimilarity measure should be used to cluster the shoppers? If Euclidean distance is used, then shoppers who have bought very few items overall (i.e. infrequent users of the online shopping site) will be clustered together. This may not be desirable. On the other hand, if correlation-based distance is used, then shoppers with similar preferences (e.g. shoppers who have bought items A and B but
Hierarchical clustering can be performed with respect to a distance measure or a similarity measure.

Correlation is often a better choice:
Summary

Clustering depends on the choice of similarity/distance and preprocessing.

Different methods will give different results.

Clustering algorithms will find as many clusters as you ask for: need methods for deciding the number of clusters.

Clustering is sensitive to noise.

Hard choices to make - there is no teaching signal as we had in supervised learning.