COMPSCI 589
Lecture 16: K-Means Clustering

Benjamin M. Marlin

College of Information and Computer Sciences
University of Massachusetts Amherst

Slides by Benjamin M. Marlin (marlin@cs.umass.edu).
Created with support from National Science Foundation Award# IIS-1350522.

The Clustering Task

Definition: The Clustering Task

Given a collection of data cases $\mathbf{x}_i \in \mathbb{R}^D$, partition the data cases into groups such that the data cases within each partition are more similar to each other than they are to data cases in other partitions.

Defining a Clustering

- Suppose we have N data cases $\mathcal{D} = \{\mathbf{x}_i\}_{i=1:N}$.
- A clustering of the N cases into K clusters is a partitioning of $\mathcal{D}$ into K mutually disjoint subsets $\mathcal{C} = \{C_1, ..., C_K\}$ such that $C_1 \cup ... \cup C_K = \mathcal{D}$.

The Hierarchical Agglomerative Clustering Algorithm

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.

The K-Means Algorithm

- The K-Means algorithm is an iterative optimization algorithm for clustering that alternates between two steps.
- The algorithm maintains a set of K cluster centroids or prototypes μ_k that represent the average (mean) feature vector of the data cases in each cluster.
- In the first step, the distance between each data case and each prototype is computed, and each data case is assigned to the nearest prototype.
- In the second step, the prototypes are updated to the mean of the data cases assigned to them.

The K-Means Algorithm

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 kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.

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

The K-Means Algorithm

Suppose we let z_i indicate which cluster $\mathbf{x}_i$ belongs to and $\mu_k \in \mathbb{R}^D$ be the cluster centroid/prototype for cluster k. The two main steps of the algorithm can then be expressed as follows:

1. $z_i = \arg\min_k ||\mu_k - \mathbf{x}_i||_2^2$
2. $\mu_k = \frac{\sum_{i=1}^N [z_i = k]\mathbf{x}_i}{\sum_{i=1}^N [z_i = k]}$

The K-Means Objective

- The K-Means algorithm attempts to minimize the sum of the within-cluster variation over all clusters (also called the within-cluster sum of squares):

$$\mathcal{C}^* = \arg\min_{\mathcal{C}} \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{\mathbf{x}_i, \mathbf{x}_j \in C_k} ||\mathbf{x}_i - \mathbf{x}_j||_2^2$$

- Note that this objective function has many, many local optima in general, each corresponding to a different clustering of the data.
- K-Means produces a non-increasing sequence of objective function values and is guaranteed to converge to some local optima. Finding the global optimum is not computationally tractable.

Initialization

- Because K-Means finds a local optimum, it can be highly sensitive to initialization.
- It is common to perform multiple random re-starts of the algorithm and take the clustering with the minimal total variation.
- Common initializations include setting the initial centers to be randomly selected data cases, setting the initial partition to a random partition, and selecting centers using a "furthest first"-style heuristic (more formally known as K-Means++).
- It often helps to initially to run with $K\log(K)$ clusters, then merge clusters to get down to K and run the algorithm from that initialization.

Issues

- Only works with Euclidean distance. An alternate version based on Manhattan distance exists and is called the K-medians algorithm.
- Pre-processing like re-scaling/normalizing features can completely change the results.
- We need some way to determine the "right" number of clusters to focus on. We want to cluster on salient differences between data cases, not noise.
- The run time is $O(NKT)$ where T is the number of iterations to convergence of the total variation. T is often small (like 20), but examples can be constructed that require an exponential number of steps to converge.
- Results in a hard assignment of data cases to clusters, which may be a problem if there are outliers.