CS 760: Machine Learning

Spring 2024

Topic 13: K-Means Clustering

INSTRUCTOR: DANIEL L. PIMENTEL-ALARCÓN

© Copyright 2024

DO NOT POLLUTE! AVOID PRINTING, OR PRINT 2-SIDED MULTIPAGE.

13.1 Introduction

Recall that classification can be summarized as assigning a label (class) $y \in \{1, 2, ..., C\} =: [C]$ to a data point $\mathbf{x} \in \mathbb{R}^{D}$ based on a collection of training data points $\mathbf{x}_{1}, \mathbf{x}_{2}, ..., \mathbf{x}_{N} \in \mathbb{R}^{D}$ whose corresponding classes $y_{1}, y_{2}, ..., y_{N} \in [C]$ are already known. To this end we can use nearest neighbors and other *supervised* learning algorithms.

However, in data science the labels $\{y_i\}_{i=1}^N$ of the training data points $\{x_i\}_{i=1}^N$ are often unavailable:

- Given a collection of vectorized images $\{\mathbf{x}_i\}_{i=1}^N$, we want to determine which correspond to the same individuals (but we don't know their names).
- Given a collection of vectors $\{\mathbf{x}_i\}_{i=1}^N$ containing information about people's movies ratings (e.g., Netflix or Amazon), we want to determine which people have similar preferences.
- Given a collection of vectors $\{\mathbf{x}_i\}_{i=1}^N$ containing genomic sequences from different organisms in a human gut microbiome sample, determine which sequences correspond to the same species.

Unsupervised learning refers to the tasks when labels $\{y_i\}_{i=1}^N$ are unavailable. Clustering is one of such tasks.

13.2 Clustering

The task of clustering can be summarized as splitting a collection of data points into groups such that the points in each group are *similar*. More precisely, given a collection of data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \in \mathbb{R}^D$, we want to identify a partition $\{C_1, C_2, \ldots, C_K\}$ of [N] (called clusters) such that if $i, j \in C_k$, then \mathbf{x}_i and \mathbf{x}_j are *close* to each other (recall that there are several ways to define how *close* two points are, e.g., $\|\mathbf{x}_i - \mathbf{x}_j\|_2$, $\|\mathbf{x}_i - \mathbf{x}_j\|_1$, or $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$).

Recall that norms satisfy the so-called triangle inequality: $\|\mathbf{x}\| + \|\mathbf{y}\| \ge \|\mathbf{x} + \mathbf{y}\|$, which implies that if \mathbf{x} is close to \mathbf{y} , and \mathbf{y} is close to \mathbf{z} , then \mathbf{x} is also close to \mathbf{z} . Using this insight, we can rephrase/adapt our clustering goal in terms of *centers* as follows: given a collection of data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \in \mathbb{R}^D$, we want to identify *centers* $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^D$ that minimize the within-cluster distances:

$$\sum_{k=1}^{K} \sum_{i \in C_k} \|\boldsymbol{\mu}_k - \mathbf{x}_i\|,$$



Figure 13.1: Initial and final step of Lloyd's algorithm.

where $i \in C_k$ if $\|\boldsymbol{\mu}_k - \mathbf{x}_i\| \leq \|\boldsymbol{\mu}_{\ell} - \mathbf{x}\|$ for every $\ell \in [K]$. Notice that this is a kind of chicken and egg problem: you need to know the clusters $\{C_k\}$ to find the centers $\{\boldsymbol{\mu}_k\}$, and you need to know the centers in order to determine the clusters. This observation is the main insight behind Lloyd's algorithm.

13.3 Lloyd's Algorithm

Lloyd's algorithm, aka the K-means clustering algorithm, is perhaps the simplest unsupervised clustering method, which uses an *alternating* strategy that is very common in machine learning. The main idea is to (i) *pretend* that we know the centers and determine the clusters, (ii) *pretend* that we know the clusters and compute the centers, and then alternate between steps (i) and (ii) until convergence. More precisely, we start with some initial estimates $\hat{\mu}_1, \hat{\mu}_2, \ldots, \hat{\mu}_K \in \mathbb{R}^D$, and then

(i) Assign each datapoint to its closest center to produce a clustering:

$$\hat{C}_k = \left\{ i \in [N] : \| \hat{\boldsymbol{\mu}}_k - \mathbf{x}_i \| \le \| \hat{\boldsymbol{\mu}}_\ell - \mathbf{x} \| \forall \ell \in [K] \right\}.$$

(ii) Compute the center of each cluster:

$$\hat{\boldsymbol{\mu}}_{\mathrm{k}} = rac{1}{|\hat{C}_{\mathrm{k}}|} \sum_{\mathrm{i}\in\hat{C}_{\mathrm{k}}} \mathbf{x}_{\mathrm{i}}.$$

Finally, we alternate between steps (i) and (ii) until convergence. See Figure 13.1 to build some intuition.

13.4 Initialization

As with most *alternating* algorithms, Lloyd's algorithm depends heavily on initialization, that is, the choice of initial centers $\{\hat{\boldsymbol{\mu}}_k\}_{k=1}^{K}$. There are several popular options:

- Random samples. This option simply selects K data points randomly, that is, $\hat{\mu}_k = x_i$ for some randomly chosen i. This tends to spread out initial centers.
- Random partition. This option first partitions data randomly into K clusters, and then computes the initial centers as the mean of each cluster. This tends to place all initial centers close to the center of the entire dataset.

• K-means++. This option aims to spread initial centers according to the data distribution. To this end, K-means++ selects one random data point \mathbf{x}_i as the first center $\hat{\boldsymbol{\mu}}_1$, and then for every $2 \le k \le K$, it chooses the kth center from the remaining data with probability proportional to its closest existing center. That is, if \mathbf{x}_j is none of the first k-1 centers, then \mathbf{x}_j is chosen as the kth center with probability proportional to

$$\min_{\ell \in [k-1]} \| \hat{\boldsymbol{\mu}}_{\ell} - \mathbf{x}_j \|^2.$$

With this initialization, Lloyd's algorithm is guaranteed to find a solution that is close (within a log K factor) to the optimal solution. This is remarkable, because the K-means problem is non-convex, NP-hard, so it is not evident that *any* algorithm should work.