EECS 553: Machine Learning (ECE) **University of Michigan**

k-means

Winter 2023 **Clayton Scott**

1 Clustering

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$. Clustering is the following problem: Partition $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ into disjoint subsets called *clusters* such that points in the same cluster are more similar to each other than to points in other clusters. A partition of the data points, i.e., a "clustering," can be represented by a function

$$C : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, k\} \tag{1}$$

where k is the number of clusters.

Clustering is unlike supervised learning in that the goal is not to generalize to new data points, but rather to draw conclusions about the training data.

The definition of clustering above hinges on a notion of similarity. There is no one agreed-upon notion of similarity, with different algorithms using different similarity measures. Since similarity is intrinsic to the definition of clustering, this means there is not necessarily one "right" or "optimal" way to cluster a dataset. Clustering is often performed in the context of a larger data analysis pipeline, in which case the efficacy of different clustering algorithms can be evaluated based on how well they enable the completion of some larger task.

2 k-means Criterion

The k-means criterion is to choose C to minimize

$$W(C) := \sum_{\ell=1}^{k} \sum_{i:C(i)=\ell} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_\ell\|^2$$

where

$$\bar{\boldsymbol{x}}_\ell = \frac{1}{n_\ell} \sum_{j:C(j)=\ell} \boldsymbol{x}_j$$

$$n_\ell = \#\{i : C(i) = \ell\}$$

This criterion aims to find C to minimize the sum of squared distances from each data point to the "centroid" or average of all points assigned to the same cluster. This smaller this quantity, the better the cluster map partitions the data into compact or homogeneous clusters. Note that k is assumed fixed and known.

Remark 1. $W(C)$ is sometimes called the *within class scatter*, because it can be shown that

$$W(C) = \frac{1}{2} \sum_{\ell=1}^{k} \sum_{i:C(i)=\ell} \left[\frac{1}{n_\ell} \sum_{j:C(j)=\ell} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \right]. \tag{2}$$

Thus $W(C)$ can also be viewed as the sum over all datapoints $\boldsymbol{x}_i$ of the average squared distance of $\boldsymbol{x}_i$ to every other data point in the same cluster. This reveals that the similarity measure underlying k-means is the squared Euclidean distance.

Establishing Eqn. (2) is an exercise in algebra. First observe that for any ℓ

$$\begin{aligned}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 &= \|(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_\ell) - (\boldsymbol{x}_j - \bar{\boldsymbol{x}}_\ell)\|^2 \\ &= \langle (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_\ell) - (\boldsymbol{x}_j - \bar{\boldsymbol{x}}_\ell), (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_\ell) - (\boldsymbol{x}_j - \bar{\boldsymbol{x}}_\ell) \rangle \\ &= \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_\ell\|^2 - 2\langle \boldsymbol{x}_i - \bar{\boldsymbol{x}}_\ell, \boldsymbol{x}_j - \bar{\boldsymbol{x}}_\ell \rangle + \|\boldsymbol{x}_j - \bar{\boldsymbol{x}}_\ell\|^2.\end{aligned}$$

Then

$$\begin{aligned}\frac{1}{2}\sum_{\ell=1}^{k}\sum_{i:C(i)=\ell}\left[\frac{1}{n_\ell}\sum_{j:C(j)=\ell}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right] &= \frac{1}{2}\sum_{\ell=1}^{k}\frac{1}{n_\ell}\Bigg[\sum_{i:C(i)=\ell}\sum_{j:C(j)=\ell}\|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_\ell\|^2 \\ &\qquad - 2\sum_{i:C(i)=\ell}\sum_{j:C(j)=\ell}(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_\ell)^T(\boldsymbol{x}_j - \bar{\boldsymbol{x}}_\ell) + \sum_{i:C(i)=\ell}\sum_{j:C(j)=\ell}\|\boldsymbol{x}_j - \bar{\boldsymbol{x}}_\ell\|^2\Bigg] \\ &= \frac{1}{2}\sum_{\ell=1}^{k}\frac{1}{n_\ell}\Bigg[\sum_{i:C(i)=\ell}\sum_{j:C(j)=\ell}\|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_\ell\|^2 + \sum_{i:C(i)=\ell}\sum_{j:C(j)=\ell}\|\boldsymbol{x}_j - \bar{\boldsymbol{x}}_\ell\|^2\Bigg] \\ &= W(C).\end{aligned}$$

3 k-means Algorithm

Minimizing the k-means criterion is a combinatorial optimization problem. The number of possible cluster maps C is

$$\frac{1}{k!}\sum_{\ell=1}^{k}(-1)^{k-\ell}\binom{k}{\ell}\ell^n$$

which is extremely large ever for moderate values of n and k (Jain and Dubes, 1998). There is no known efficient search strategy for this search space. Therefore we resort to an iterative, suboptimal algorithm known as k-means.

Input: $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, k
Initialize $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_k \in \mathbb{R}^d$
Repeat
  For $i = 1, \ldots, n$
    $C(i) = \arg\min_\ell \|\boldsymbol{x}_i - \boldsymbol{m}_\ell\|$
  End
  For $\ell = 1, \ldots, k$
    $\boldsymbol{m}_\ell = \frac{1}{|\{i:C(i)=\ell\}|}\sum_{i:C(i)=\ell}\boldsymbol{x}_i$
  End
Until termination criterion is satisfied
Output: C

This algorithm is also known as Lloyd's algorithm or the Lloyd-Max algorithm. The same algorithm is used in the problem of vector quantization.

Remark 2. k-means can be viewed as an alternating algorithm[1] to minimize $W(C)$. To see this, observe that for fixed C,

$$\bar{\boldsymbol{x}}_\ell = \arg\min_{\boldsymbol{m}}\frac{1}{n_\ell}\sum_{i:C(i)=\ell}\|\boldsymbol{x}_i - \boldsymbol{m}\|^2. \tag{3}$$

[1]Let the variables of an objective function be partitioned into $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. An alternating algorithm is one that alternates between optimizing over $\boldsymbol{\alpha}$ with $\boldsymbol{\beta}$ held fixed, and optimizing over $\boldsymbol{\beta}$ with $\boldsymbol{\alpha}$ held fixed.

This can be seen by taking the gradient with respect to $\boldsymbol{m}$, equating to zero, and solving for $\boldsymbol{m}$, and is left as an exercise. Now consider the expanded optimization problem

$$\min_{C, \boldsymbol{m}_1, \ldots, \boldsymbol{m}_k} \sum_{\ell=1}^{k} \sum_{i:C(i)=\ell} \|\boldsymbol{x}_i - \boldsymbol{m}_\ell\|^2.$$

This problem can be expressed

$$\min_{C} \left[\min_{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_k} \sum_{\ell=1}^{k} \sum_{i:C(i)=\ell} \|\boldsymbol{x}_i - \boldsymbol{m}_\ell\|^2 \right] = \min_{C} \sum_{\ell=1}^{k} \left[\min_{\boldsymbol{m}_\ell} \sum_{i:C(i)=\ell} \|\boldsymbol{x}_i - \boldsymbol{m}_\ell\|^2 \right].$$

From (3), the term in brackets is $\sum_{i:C(i)=\ell} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_\ell\|^2$, and therefore the expanded problem and the original problem (of minimizing $W(C)$) have the same optimal value of C.

The k-means algorithm is now clearly an alternating algorithm for solving the expanded problem. The algorithm alternates between optimizing for C with $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_k$ fixed, and optimizing for $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_k$ with C fixed. Each step can be solved efficiently and exactly. From this perspective, we can also see that the k-means algorithm produces a sequence of cluster maps $W(C_1), W(C_2), \ldots$ such that $W(C_1) \geq W(C_2) \geq \ldots$. This is because every step of the alternating algorithm does not increase the objective function.

4 Initialization

The k-means algorithm is highly dependent on initialization. One common strategy is to initialize $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_k$ to be randomly chosen data points. It is also common to run the algorithm several time with different initializations, and take the run with smallest $W(C)$. Unfortunately, random initialization has some problems.

- The number of iterations can be quite large in the worst case
- The converged value of $W(C)$ can be quite far from the optimal one.

A better idea is to choose the initial $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_k$ to be far apart. A particular implementation of this idea, with guaranteed performance relative to the optimal solution, is called k-means++:

1. Choose the first cluster center $\boldsymbol{m}_1$ at random from among $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$.
2. For each $\boldsymbol{x} \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, compute $D(\boldsymbol{x})$, the distance from $\boldsymbol{x}$ to the nearest cluster center that has been selected
3. Choose one new data point $\boldsymbol{x}$ at random as a new cluster center, with probability proportional to $D(\boldsymbol{x})^2$
4. Repeat steps 2-3 until k centers have been selected

For more on k-means++, see the original paper by Arthur and Vassilvitskii (2007). They show that k-means++ leads to an objective value within a guaranteed tolerance of the global optimum.

4.1 Termination

There are several options for terminating k-means:

- Fixed number of iterations
- When the decrease in $W(C)$ is sufficiently small
- When the clusters haven't changed

5 Cluster Geometry

The clusters produced by k-means are "nearest neighbor" regions, also known as Voronoi cells, defined with respect to the cluster centers. Therefore the cluster boundaries are piecewise linear, and the clusters are convex sets. See Figure 1.

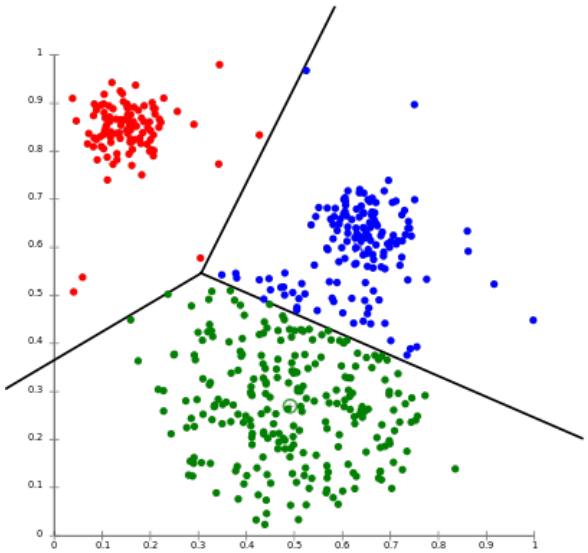

Figure 1: k-means produces convex clusters.

k-means will fail to identify the true clusters if at least one of them is nonconvex. From the figure we also see that k-means struggles when clusters have different sizes. k-means can be kernelized to accommodate clusters of different shapes, including nonconvex clusters.

6 Model Selection

How should k be chosen? Let $\hat{C}_k$ denote the output of k-means. One simple heuristic is to plot $\mathrm{W}(\hat{C}_k)$ as a function of k.

The basic idea is that if k^* is the ideal cluster number, then

- If $k < k^*$, $W(\hat{C}_k) - W(\hat{C}_{k+1})$ will be relatively large (adding clusters leads to a substantial reduction in the k-means criterion).

- If $k \geq k^*$, $W(\hat{C}_k) - W(\hat{C}_{k+1})$ will be relatively small (adding clusters provide a small decrease in the k-means criterion).

This suggests choosing k near the "knee" of the curve. See Figure 2.

A more systematic method was developed by Kulis and Jordan in the paper "Revisiting k-means" (2007). They suggest optimizing the following objective with respect to both C and k:

$$\sum_{\ell=1}^{k} \sum_{i:C(i)=\ell} \|\boldsymbol{x}_i - \boldsymbol{x}_\ell\|^2 + \lambda k$$

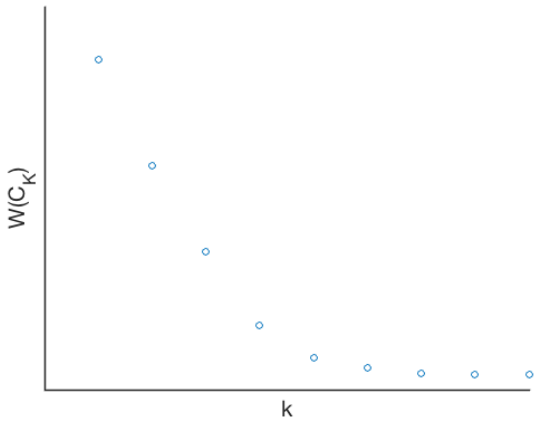

Figure 2: Hueristic for selecting k.

where λ is a trade off parameter. This criterion is derived from a nonparametric Bayesian perspective, and can be optimized (suboptimally) by a variant of the k-means algorithm.

7 Uses of k-means

k-means makes such strong assumptions about the data that is rarely used as a clustering algorithm on unprocessed data. It is sometimes used for clustering after the data have been suitably processed so that the assumptions of the algorithm are satisfied. We will see an example of this when we study spectral clustering.

k-means is also used as an algorithm for vector quantization to speed up other machine learning algorithms. As an example, consider the nearest neighbor classifier. This classifier is computationally expensive to evaluate, requiring a comparison with all n training data points. A more efficient alternative is to first run k-means on the training data, and associate each of the k centroids with a class, by taking the majority vote over all training data points in that cluster. Then, a test point may be classified by assigning it the label of the nearest centroid, which only requires k comparisons. The resulting classifier is an approximation of the full nearest neighbor classifier, but it can be much faster computationally. When used in this way, k does not correspond to the number of cluster/classes. Vector quantization is just a convenient way to compress multidimensional data, and k is a tuning parameter that trades off between speed and accuracy.

Exercises

1. (★★) Kernelize k-means. Be sure to address initialization.
2. (★★) Prove (3). *Hint*: One approach is the gradient-based approach suggested in the text. Another approach, that avoids gradients, is to add and subtract $\bar{\boldsymbol{x}}_\ell$ within the term $\boldsymbol{x}_i - \boldsymbol{m}$, expand, and reason directly about the minimizer.
3. (★★) Use the alternating minimization perspective to argue that k-means is a *descent* algorithm, meaning that at every iteration, the value of $W(C)$ decreases. In other words, k-means produces a sequence of cluster maps $C_1, C_2, \ldots$ such that $W(C_1) \geq W(C_2) \geq \cdots$.