Run the Clustering Algorithm

In machine learning, you sometimes encounter datasets that can have millions of examples. ML algorithms must scale efficiently to these large datasets. However, many clustering algorithms do not scale because they need to compute the similarity between all pairs of points. This means their runtimes increase as the square of the number of points, denoted as $O(n^2)$. For example, agglomerative or divisive hierarchical clustering algorithms look at all pairs of points and have complexities of $O(n^2 log(n))$ and $O(n^2)$, respectively.

This course focuses on k-means because it scales as $O(nk)$, where k is the number of clusters. k-means groups points into k clusters by minimizing the distances between points and their cluster's centroid (as seen in Figure 1 below). The **centroid** (/machine-learning/glossary#centroid) of a cluster is the mean of all the points in the cluster.

As shown, k-means finds roughly circular clusters. Conceptually, this means k-means effectively treats data as composed of a number of roughly circular distributions, and tries to find clusters corresponding to these distributions. In reality, data contains outliers and might not fit such a model.

Before running k-means, you must choose the number of clusters, k. Initially, start with a guess for k. Later, we'll discuss how to refine this number.

k-means Clustering Algorithm

To cluster data into k clusters, k-means follows the steps below:

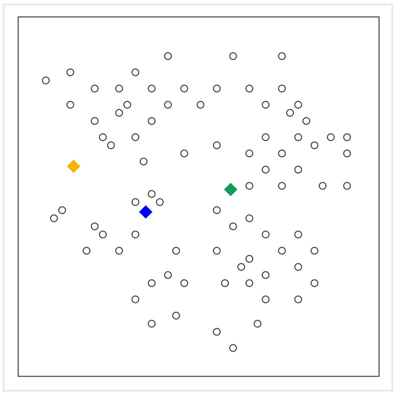

Figure 1: k-means at initialization.

Step One

The algorithm randomly chooses a centroid for each cluster. In our example, we choose a k of 3, and therefore the algorithm randomly picks 3 centroids.

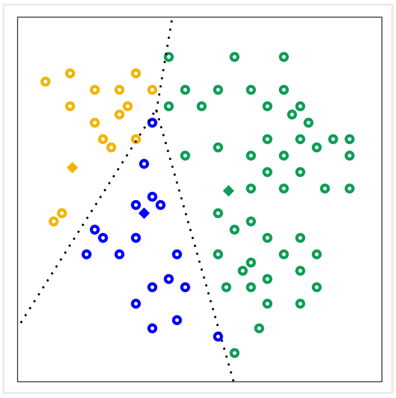

Figure 2: Initial clusters.

Step Two

The algorithm assigns each point to the closest centroid to get k initial clusters.

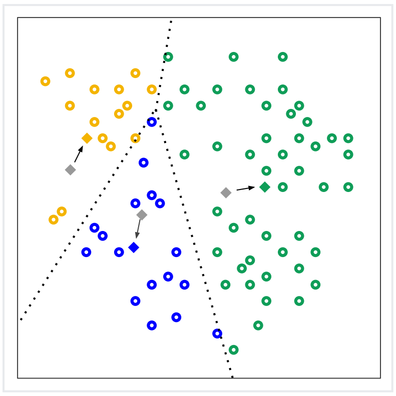

Figure 3: Recomputation of centroids.

Step Three

For every cluster, the algorithm recomputes the centroid by taking the average of all points in the cluster. The changes in centroids are shown in Figure 3 by arrows. Since the centroids change, the algorithm then re-assigns the points to the closest centroid. Figure 4 shows the new clusters after re-assignment.

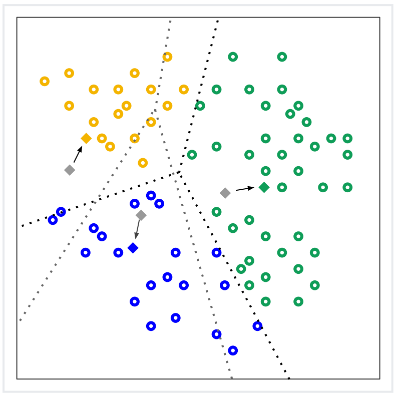

Figure 4: Clusters after reassignment.

Step Four

The algorithm repeats the calculation of centroids and assignment of points until points stop changing clusters. When clustering large datasets, you stop the algorithm before reaching convergence, using other criteria instead.

You do not need to understand the math behind k-means for this course. However, if you are curious, see below for the mathematical proof.

Click the plus icon for the mathematical proof

Given n examples assigned to k clusters, minimize the sum of distances of examples to their centroids. Where:

- $A_{nk} = 1$ when the nth example is assigned to the kth cluster, and 0 otherwise
- θ_k is the centroid of cluster k

We want to minimize the following expression:

$$\min_{A,\theta} \sum_{n=1}^{N} \sum_{k=1}^{K} A_{nk} ||\theta_k - x_n||^2$$

subject to:

$$A_{nk} \in \{0,1\} \forall n,k$$

and

$$\sum_{k=1}^{K} A_{nk} = 1 \forall n$$

To minimize the expression with respect to the cluster centroids θ_k, take the derivative with respect to θ_k and equate it to 0.

$$f(\theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} A_{nk} ||\theta_k - x_n||^2$$

$$\frac{\partial f}{\partial \theta_k} = 2 \sum_{n=1}^{N} A_{nk} (\theta_k - x_n) = 0$$

$$\implies \sum_{n=1}^{N} A_{nk}\theta_k = \sum_{n=1}^{N} A_{nk}x_n$$

$$\theta_k \sum_{n=1}^{N} A_{nk} = \sum_{n=1}^{N} A_{nk}x_n$$

$$\theta_k = \frac{\sum_{n=1}^{N} A_{nk}x_n}{\sum_{n=1}^{N} A_{nk}}$$

The numerator is the sum of all example-centroid distances in the cluster. The denominator is the number of examples in the cluster. Thus, the cluster centroid θ_k is the average of example-centroid distances in the cluster. Hence proved.

Because the centroid positions are initially chosen at random, k-means can return significantly different results on successive runs. To solve this problem, run k-means multiple times and choose the result with the best quality metrics. (We'll describe quality metrics later in this course.) You'll need an advanced version of k-means to choose better initial centroid positions.

ment: Using this k-means simulator (http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.htr
ꜱtanford, try running k-means multiple times and see if you get different results.

ɛrms:

centroid (/machine-learning/glossary#centroid)

Previous
← Summary (/machine-learning/clustering/similarity/similarity-summary)

Next
Interpret Results (/machine-learning/clustering/interpret) →

Last updated 2020-02-10 UTC.