4.2 A clustering objective

In this section we formalize the idea of clustering, and introduce a natural measure of the quality of a given clustering.

Specifying the cluster assignments. We specify a clustering of the vectors by saying which cluster or group each vector belongs to. We label the groups $1, \ldots, k$, and specify a clustering or assignment of the N given vectors to groups using an N-vector c, where c_i is the group (number) that the vector x_i is assigned to. As a simple example with $N = 5$ vectors and $k = 3$ groups, $c = (3, 1, 1, 1, 2)$ means that x_1 is assigned to group 3, x_2, x_3, and x_4 are assigned to group 1, and x_5 is assigned to group 2. We will also describe the clustering by the sets of indices for each group. We let G_j be the set of indices corresponding to group j. For our simple example above, we have

$$G_1 = \{2, 3, 4\}, \qquad G_2 = \{5\}, \qquad G_3 = \{1\}.$$

(Here we are using the notation of sets; see appendix A.) Formally, we can express these index sets in terms of the group assignment vector c as

$$G_j = \{i \mid c_i = j\},$$

which means that G_j is the set of all indices i for which $c_i = j$.

Group representatives. With each of the groups we associate a *group representative* n-vector, which we denote $z_1, \ldots, z_k$. These representatives can be any n-vectors; they do not need to be one of the given vectors. We want each representative to be close to the vectors in its associated group, *i.e.*, we want the quantities

$$\|x_i - z_{c_i}\|$$

to be small. (Note that x_i is in group $j = c_i$, so z_{c_i} is the representative vector associated with data vector x_i.)

A clustering objective. We can now give a single number that we use to judge a choice of clustering, along with a choice of the group representatives. We define

$$J^{\text{clust}} = \left(\|x_1 - z_{c_1}\|^2 + \cdots + \|x_N - z_{c_N}\|^2\right)/N, \tag{4.1}$$

which is the mean square distance from the vectors to their associated representatives. Note that J^{clust} depends on the cluster assignments (*i.e.*, c), as well as the choice of the group representatives $z_1, \ldots, z_k$. The smaller J^{clust} is, the better the clustering. An extreme case is $J^{\text{clust}} = 0$, which means that the distance between every original vector and its assigned representative is zero. This happens only when the original collection of vectors only takes k different values, and each vector is assigned to the representative it is equal to. (This extreme case would probably not occur in practice.)

Our choice of clustering objective J^{clust} makes sense, since it encourages all points to be near their associated representative, but there are other reasonable

choices. For example, it is possible to use an objective that encourages more balanced groupings. But we will stick with this basic (and very common) choice of clustering objective.

Optimal and suboptimal clustering. We seek a clustering, *i.e.*, a choice of group assignments $c_1, \ldots, c_N$ and a choice of representatives $z_1, \ldots, z_k$, that minimize the objective J^{clust}. We call such a clustering *optimal*. Unfortunately, for all but the very smallest problems, it is practically impossible to find an optimal clustering. (It can be done in principle, but the amount of computation needed grows extremely rapidly with N.) The good news is that the k-means algorithm described in the next section requires far less computation (and indeed, can be run for problems with N measured in billions), and often finds a very good, if not the absolute best, clustering. (Here, 'very good' means a clustering and choice of representatives that achieves a value of J^{clust} near its smallest possible value.) We say that the clustering choices found by the k-means algorithm are *suboptimal*, which means that they might not give the lowest possible value of J^{clust}.

Even though it is a hard problem to choose the best clustering and the best representatives, it turns out that we *can* find the best clustering, if the representatives are fixed, and we can find the best representatives, if the clustering is fixed. We address these two topics now.

Partitioning the vectors with the representatives fixed. Suppose that the group representatives $z_1, \ldots, z_k$ are fixed, and we seek the group assignments $c_1, \ldots, c_N$ that achieve the smallest possible value of J^{clust}. It turns out that this problem can be solved exactly.

The objective J^{clust} is a sum of N terms. The choice of c_i (*i.e.*, the group to which we assign the vector x_i) only affects the ith term in J^{clust}, which is $(1/N)\|x_i - z_{c_i}\|^2$. We can choose c_i to minimize just this term, since c_i does not affect the other $N-1$ terms in J^{clust}. How do we choose c_i to minimize this term? This is easy: We simply choose c_i to be the value of j that minimizes $\|x_i - z_j\|$ over j. In other words, we should assign each data vector x_i to its nearest neighbor among the representatives. This choice of assignment is very natural, and easily carried out.

So when the group representatives are fixed, we can readily find the best group assignment (*i.e.*, the one that minimizes J^{clust}), by assigning each vector to its nearest representative. With this choice of group assignment, we have (by the way the assignment is made)

$$\|x_i - z_{c_i}\| = \min_{j=1,\ldots,k} \|x_i - z_j\|,$$

so the value of J^{clust} is given by

$$\left(\min_{j=1,\ldots,k} \|x_1 - z_j\|^2 + \cdots + \min_{j=1,\ldots,k} \|x_N - z_j\|^2 \right) /N.$$

This has a simple interpretation: It is the mean of the squared distance from the data vectors to their closest representative.

Optimizing the group representatives with the assignment fixed. Now we turn to the problem of choosing the group representatives, with the clustering (group assignments) fixed, in order to minimize our objective J^{clust}. It turns out that this problem also has a simple and natural solution.

We start by re-arranging the sum of N terms into k sums, each associated with one group. We write

$$J^{\text{clust}} = J_1 + \cdots + J_k,$$

where

$$J_j = (1/N) \sum_{i \in G_j} \|x_i - z_j\|^2$$

is the contribution to the objective J^{clust} from the vectors in group j. (The sum here means that we should add up all terms of the form $\|x_i - z_j\|^2$, for any $i \in G_j$, *i.e.*, for any vector x_i in group j; see appendix A.)

The choice of group representative z_j only affects the term J_j; it has no effect on the other terms in J^{clust}. So we can choose each z_j to minimize J_j. Thus we should choose the vector z_j so as to minimize the mean square distance to the vectors in group j. This problem has a very simple solution: We should choose z_j to be the average (or mean or centroid) of the vectors x_i in its group:

$$z_j = (1/|G_j|) \sum_{i \in G_j} x_i,$$

where $|G_j|$ is standard mathematical notation for the number of elements in the set G_j, *i.e.*, the size of group j. (See exercise 4.1.)

So if we fix the group assignments, we minimize J^{clust} by choosing each group representative to be the average or centroid of the vectors assigned to its group. (This is sometimes called the *group centroid* or *cluster centroid*.)

4.3 The k-means algorithm

It might seem that we can now solve the problem of choosing the group assignments and the group representatives to minimize J^{clust}, since we know how to do this when one or the other choice is fixed. But the two choices are circular, *i.e.*, each depends on the other. Instead we rely on a very old idea in computation: We simply *iterate* between the two choices. This means that we repeatedly alternate between updating the group assignments, and then updating the representatives, using the methods developed above. In each step the objective J^{clust} gets better (*i.e.*, goes down) unless the step does not change the choice. Iterating between choosing the group representatives and choosing the group assignments is the celebrated *k-means algorithm* for clustering a collection of vectors.

The k-means algorithm was first proposed in 1957 by Stuart Lloyd, and independently by Hugo Steinhaus. It is sometimes called the Lloyd algorithm. The name 'k-means' has been used since the 1960s.