Chapter 4

Clustering

In this chapter we consider the task of clustering a collection of vectors into groups or clusters of vectors that are close to each other, as measured by the distance between pairs of them. We describe a famous clustering method, called the *k-means algorithm*, and give some typical applications.

The material in this chapter will not be used in the sequel. But the ideas, and the k-means algorithm in particular, are widely used in practical applications, and rely only on the ideas developed in the previous three chapters. So this chapter can be considered an interlude that covers useful material that builds on the ideas developed so far.

4.1 Clustering

Suppose we have N n-vectors, $x_1, \ldots, x_N$. The goal of *clustering* is to group or partition the vectors (if possible) into k groups or clusters, with the vectors in each group close to each other. Clustering is very widely used in many application areas, typically (but not always) when the vectors represent features of objects.

Normally we have k much smaller than N, *i.e.*, there are many more vectors than groups. Typical applications use values of k that range from a handful to a few hundred or more, with values of N that range from hundreds to billions. Part of the task of clustering a collection of vectors is to determine whether or not the vectors can be divided into k groups, with vectors in each group near each other. Of course this depends on k, the number of clusters, and the particular data, *i.e.*, the vectors $x_1, \ldots, x_N$.

Figure 4.1 shows a simple example, with $N = 300$ 2-vectors, shown as small circles. We can easily see that this collection of vectors can be divided into $k = 3$ clusters, shown on the right with the colors representing the different clusters. We could partition these data into other numbers of clusters, but we can see that $k = 3$ is a good value.

This example is not typical in several ways. First, the vectors have dimension $n = 2$. Clustering any set of 2-vectors is easy: We simply scatter plot the values

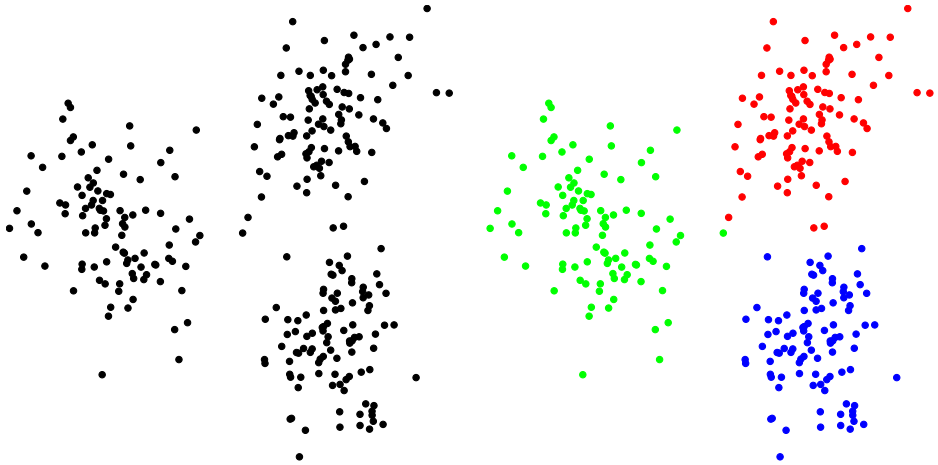

Figure 4.1 300 points in a plane. The points can be clustered in the three groups shown on the right.

and check visually if the data are clustered, and if so, how many clusters there are. In almost all applications n is larger than 2 (and typically, much larger than 2), in which case this simple visual method cannot be used. The second way in which it is not typical is that the points are very well clustered. In most applications, the data are not as cleanly clustered as in this simple example; there are several or even many points that lie in between clusters. Finally, in this example, it is clear that the best choice of k is $k = 3$. In real examples, it can be less clear what the best value of k is. But even when the clustering is not as clean as in this example, and the best value of k is not clear, clustering can be very useful in practice.

Examples. Before we delve more deeply into the details of clustering and clustering algorithms, we list some common applications where clustering is used.

- *Topic discovery.* Suppose x_i are word histograms associated with N documents. A clustering algorithm partitions the documents into k groups, which typically can be interpreted as groups of documents with the same or similar topics, genre, or author. Since the clustering algorithm runs automatically and without any understanding of what the words in the dictionary mean, this is sometimes called *automatic topic discovery*.

- *Patient clustering.* If x_i are feature vectors associated with N patients admitted to a hospital, a clustering algorithm clusters the patients into k groups of similar patients (at least in terms of their feature vectors).

- *Customer market segmentation.* Suppose the vector x_i gives the quantities (or dollar values) of n items purchased by customer i over some period of time. A clustering algorithm will group the customers into k market segments, which are groups of customers with similar purchasing patterns.

- *ZIP code clustering.* Suppose that x_i is a vector giving n quantities or statistics for the residents of ZIP code i, such as numbers of residents in various age groups, household size, education statistics, and income statistics. (In this example N is around 40000.) A clustering algorithm might be used to cluster the 40000 ZIP codes into, say, $k = 100$ groups of ZIP codes with similar statistics.

- *Student clustering.* Suppose the vector x_i gives the detailed grading record of student i in a course, *i.e.*, her grades on each question in the quizzes, homework assignments, and exams. A clustering algorithm might be used to cluster the students into $k = 10$ groups of students who performed similarly.

- *Survey response clustering.* A group of N people respond to a survey with n questions. Each question contains a statement, such as 'The movie was too long', followed by some ordered options such as

  Strongly Disagree, Disagree, Neutral, Agree, Strongly Agree.

  (This is called a *Likert scale*, named after the psychologist Rensis Likert.) Suppose the n-vector x_i encodes the selections of respondent i on the n questions, using the numerical coding -2, -1, 0, $+1$, $+2$ for the responses above. A clustering algorithm can be used to cluster the respondents into k groups, each with similar responses to the survey.

- *Weather zones.* For each of N counties we have a 24-vector x_i that gives the average monthly temperature in the first 12 entries and the average monthly rainfall in the last 12 entries. (We can standardize all the temperatures, and all the rainfall data, so they have a typical range between -1 and $+1$.) The vector x_i summarizes the annual weather pattern in county i. A clustering algorithm can be used to cluster the counties into k groups that have similar weather patterns, called *weather zones*. This clustering can be shown on a map, and used to recommend landscape plantings depending on zone.

- *Daily energy use patterns.* The 24-vectors x_i give the average (electric) energy use for N customers over some period (say, a month) for each hour of the day. A clustering algorithm partitions customers into groups, each with similar patterns of daily energy consumption. We might expect a clustering algorithm to 'discover' which customers have a swimming pool, an electric water heater, or solar panels.

- *Financial sectors.* For each of N companies we have an n-vector whose components are financial and business attributes such as total capitalization, quarterly returns and risks, trading volume, profit and loss, or dividends paid. (These quantities would typically be scaled so as to have similar ranges of values.) A clustering algorithm would group companies into *sectors*, *i.e.*, groups of companies with similar attributes.

In each of these examples, it would be quite informative to know that the vectors can be well clustered into, say, $k = 5$ or $k = 37$ groups. This can be used to develop insight into the data. By examining the clusters we can often understand them, and assign labels or descriptions to them.

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