CSCI 4588/5588: Machine Learning II
Chapter #5: Unsupervised Learning
(Ref: Ch-14 of [1])

The previous chapters have been concerned with predicting the values of one or more outputs or response variables $Y = (Y_1, \ldots, Y_m)$ for a given set of input or predictor variables $X^T = (X_1, \ldots, X_p)$. Denote by $x_i^T = (x_{i1}, x_{i2}, ..., x_{ip})$ the inputs for the i^{th} training case, and let y_i be a response measurement. The predictions are based on the training sample $(x_1, y_1), \ldots, (x_N, y_N)$ of previously solved cases, where all the variables' joint values are known. This is called *supervised learning* or "learning with a teacher." Under this metaphor, the "student" presents an answer $\hat{y}_i$ for each x_i in the training sample, and the supervisor or "teacher" provides either the correct answer and/or an error associated with the student's answer.

On the contrary, we address unsupervised learning as "learning without a teacher", basically *unsupervised learning* problem is samples without being told their categories or the outcomes. In this case, one has a set of N observations $(x_1, x_2, \ldots, x_N)$ of a random p-vector X having joint density Pr(X). The goal is to directly infer the properties of this probability density without the help of a supervisor or teacher providing correct answers or degree-of-error for each observation. The dimension of X is sometimes much higher than in supervised learning, and the properties of interest are often more complicated than simple location estimates. These factors are somewhat mitigated by the fact that X represents all of the variables under consideration; one is not required to infer how the properties of Pr(X) change, conditioned on the changing values of another set of variables.

With supervised learning, there is a clear measure of success or lack thereof that can be used to judge adequacy in particular situations and to compare the effectiveness of different methods over various situations. Lack of success is directly measured by expected loss over the joint distribution Pr(X, Y). This can be estimated in a variety of ways, including cross-validation. In the context of unsupervised learning, there is no such direct measure of success. It is difficult to ascertain the validity of inferences drawn from the output of most unsupervised learning algorithms. One must resort to heuristic arguments not only for motivating the algorithms, as is often the case in supervised learning as well but also for judgments as to the quality of the results. This uncomfortable situation has led to the heavy proliferation of proposed methods since effectiveness is a matter of opinion and cannot be verified directly. Here, in this chapter, we will study some

unsupervised learning techniques that are among the most commonly used in practice.

Association Rules **(**14.2**)**

Association rule analysis has emerged as a popular tool for mining commercial databases. The goal is to find joint values of the variables $X = (X_1, X_2, \ldots, X_p)$ that appear most frequently in the database. It is most often applied to binary-valued data $X_j \in \{0, 1\}$, where it is referred to as "market basket" analysis. In this context, the observations are sales transactions, such as those occurring at the checkout counter of a store. The variables represent all of the items sold in the store. For observation i, each variable X_j is assigned one of two values; $x_{ij} = 1$ if the j^{th} item is purchased as part of the transaction, whereas $x_{ij} = 0$ if it was not purchased. Those variables that frequently have joint values of one represent items that are frequently purchased together. This information can be quite useful for stocking shelves, cross-marketing in sales promotions, catalog design, and consumer segmentation based on buying patterns. Next, we will study clustering, which will help use segmentation data for a variety of useful goals.

Cluster Analysis (14.3)

Cluster analysis, also called data segmentation, has a variety of goals. All relate to grouping or segmenting a collection of objects into subsets or "clusters," such that those within each cluster are more closely related to one another than objects assigned to different clusters. An object can be described by a set of measurements or by its relation to other objects. Besides, the goal is sometimes to arrange the clusters into a natural hierarchy. This involves successively grouping the clusters themselves so that at each level of the hierarchy, clusters within the same group are more similar to each other than those in different groups.

Cluster analysis is also used to form descriptive statistics to ascertain whether or not the data consists of a set of distinct subgroups, each group representing objects with substantially different properties. This latter goal requires an assessment of the degree of difference between the objects assigned to the respective clusters.

Central to all of the cluster analysis goals is the notion of the degree of similarity (or dissimilarity) between the individual objects being clustered. A clustering method attempts to group the objects based on the definition of similarity supplied to it. This can only come from subject matter considerations. The situation is somewhat similar to the specification of a

loss or cost function in prediction problems (supervised learning). There the cost associated with an inaccurate prediction depends on considerations outside the data.

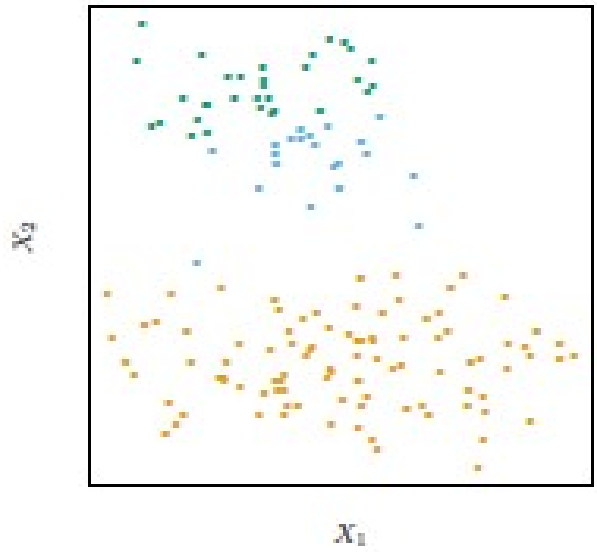


FIGURE 14.4. Simulated data in the plane, clustered into three classes (represented by orange, blue and green) by the *K*-means clustering algorithm.

Figure 14.4 shows some simulated data clustered into three groups via the popular *K*-means algorithm. In this case, two of the clusters are not well separated, so that "segmentation" more accurately describes the part of this process than "clustering." *K*-means clustering starts with guesses for the three cluster centers. Then it alternates the following steps until convergence:

- for each data point, the closest cluster center (in Euclidean distance) is identified;
- each cluster center is replaced by the coordinate-wise average of all data points that are closest to it.

Fundamental to all clustering techniques is the choice of distance or dissimilarity measure between two objects. The goal of cluster analysis is to partition the observations into groups ("clusters") so that the pairwise dissimilarities between those assigned to the same cluster tend to be smaller than those in different clusters.

K-means clustering algorithm (14.3.6)

The *K*-means algorithm is one of the most popular iterative descent clustering methods. It is intended for situations in which all variables are of the quantitative type, and squared Euclidean distance

$$d(x_i, x_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = \| x_i - x_{i'} \|^2 \qquad (i)$$

is chosen as the dissimilarity measure. Note that weighted Euclidean distance can be used by redefining the x_{ij} values.

Algorithm 14.1 *K-mean Clustering*

Inputs: Parameter *K*, dataset D $\{(x_1), \ldots, (x_N)\}$.

1. Initialize cluster centroids $\{m_1, m_2, \ldots, m_K\}$ randomly.

2. For $\forall i$ compute the cluster assignment,

$$C(i) = \arg\min_{1 \le k \le K} \|x_i - m_k\|^2$$

3. For $\forall k$ compute,

$$m_k = \frac{\sum_{i=1}^{N} I\{C(i) = k\}\, x_i}{\sum_{i=1}^{N} I\{C(i) = k\}}$$

where *I* is the indicator function. *Note, for any m_k if $\sum_{i=1}^{N} I\{C(i) = k\} = 0$ then we reset m_k randomly.

4. Goto step 2 to repeat, unless cluster assignments ($C(i)$) do not change.

It is important to note that:

* *While initializing the centroids $\{m_1, m_2, \ldots, m_K\}$ randomly, it is a better idea to pick sample points randomly and use them for initialization.*

* *Also, the algorithm is initialization dependent and can result in the locally optimal solution. Thus it is better to run several times to check globally optimal cluster or centroids are achieved or not.*

* *To pick a better set of centroids, out of different random runs, total cost using equation* (i) *can be used to compute and compare and to pick the one that has minimum cost.*

Figure 14.6 shows some of the K-means iterations for the simulated data of Figure 14.4. The centroids are depicted by "O"s. The straight lines show the partitioning of points, each sector being the set of points closest to each centroid. This partitioning is called the *Voronoi tessellation*. After 20 iterations, the procedure has converged.

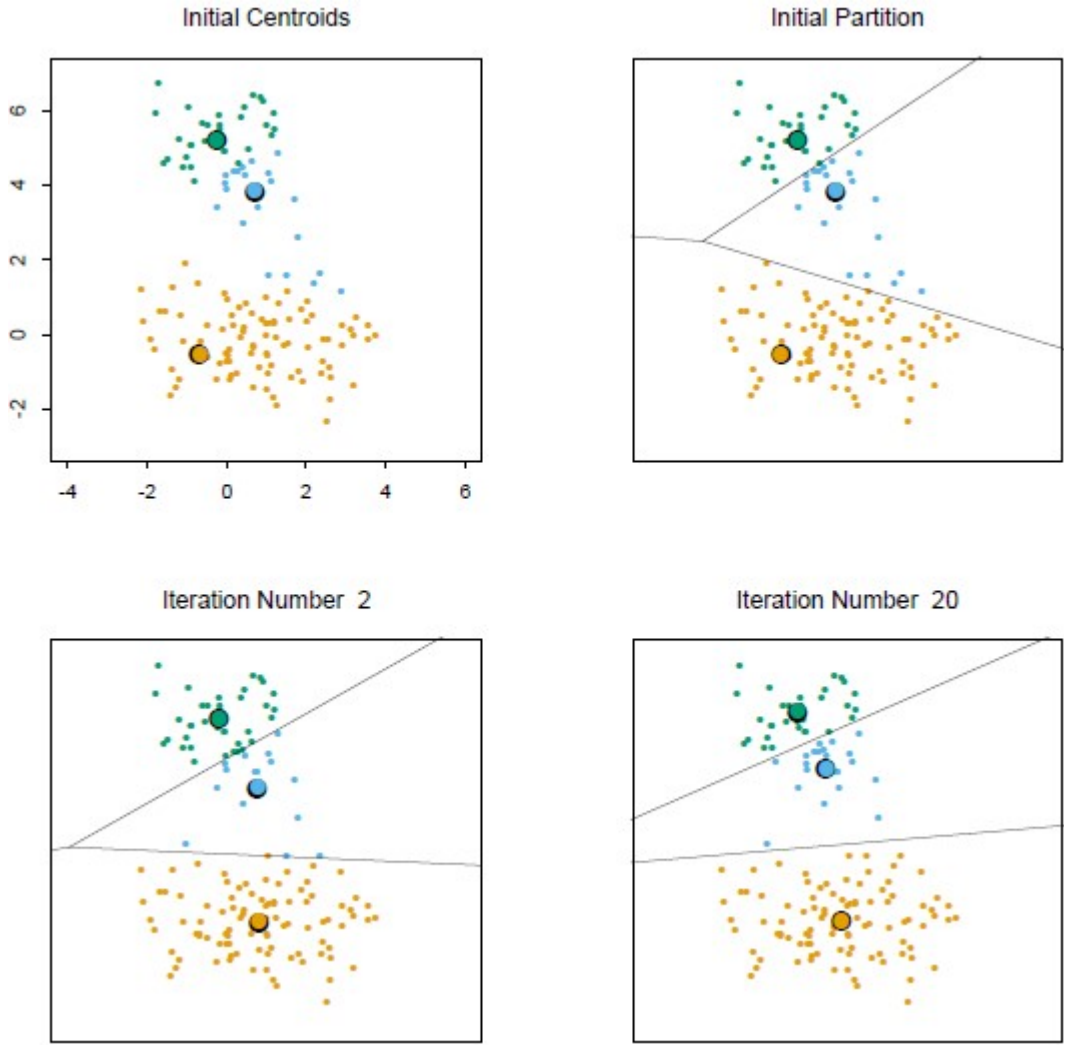


FIGURE 14.6. Successive iterations of the *K*-means clustering algorithm for the simulated data of Figure 14.4.

Practical Issues (14.3.11)

In order to apply K-means, one must select the number of clusters K and an initialization (See above, Algorithm 14.1). A choice for the number of clusters K depends on the goal. For data segmentation, K is usually defined as part of the problem. For example, a company may employ K salespeople, and the goal is to partition a customer database into K segments, one for each salesperson, such that the customers assigned to each one are as similar as possible. Often, however, cluster analysis is used to provide a descriptive statistic for ascertaining the extent to which the observations are comprising the database fall into distinct natural groupings. Here the number of such groups K^* is unknown, and one requires that it, as well as the groupings themselves, be estimated from the data.

Data-based methods for estimating K^* typically examine the within-cluster dissimilarity W_K as a function of the number of clusters K. Separate solutions are obtained for $K \in \{1, 2, \ldots, K_{\max}\}$. The corresponding values $\{W_1, W_2, \ldots, W_{K\max}\}$ generally decrease with increasing values of K.

The intuition underlying the approach is that if there are actually K^* distinct groupings of the observations (as defined by the dissimilarity measure), then for $K < K^*$ the clusters returned by the algorithm will each contain a subset of the true underlying groups. That is, the solution will not assign observations in the same naturally occurring group to different estimated clusters. To the extent that this is the case, the solution criterion value will tend to decrease substantially with each successive increase in the number of specified clusters, $W_{K+1} \ll W_K$, as the natural groups are successively assigned to separate clusters. For $K > K^*$, one of the estimated clusters must partition at least one of the natural groups into two subgroups. This will tend to provide a smaller decrease in the criterion as K is further increased. Splitting a natural group, within which the observations are all quite close to each other, reduces the criterion less than partitioning the union of two well-separated groups into their proper constituents.

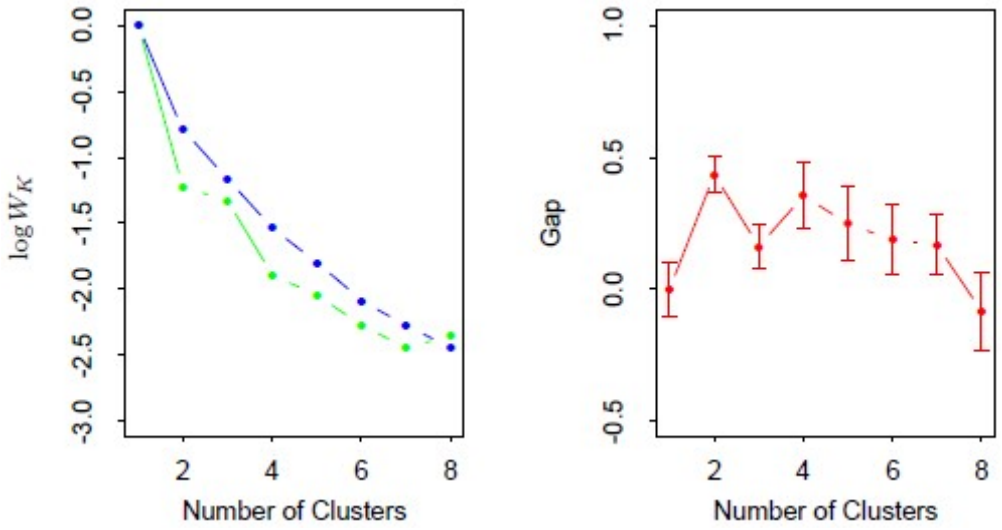


FIGURE 14.11. (Left panel): observed (green) and expected (blue) values of log W_K for the simulated data of Figure 14.4. Both curves have been translated to equal zero at one cluster. (Right panel): Gap curve, equal to the difference between the observed and expected values of log W_K. The Gap estimate K^* is the smallest K producing a gap within one standard deviation of the gap at $K + 1$; here $K^* = 2$.

To the extent this scenario is realized, there will be a sharp decrease in successive differences in criterion value, $W_K - W_{K+1}$, at $K = K^*$. That is, $\{W_K - W_{K+1} \mid K < K^*\} \gg \{W_K - W_{K+1} \mid K \geq K^*\}$. An estimate $\hat{K}^*$ for K^* is then obtained by identifying a "kink" in the plot of W_K as a function of K. As with other aspects of clustering procedures, this approach is somewhat heuristic.

The recently proposed Gap statistic (Tibshirani *et al.*, 2001b) compares the curve log W_K to the curve obtained from data uniformly distributed over a rectangle containing the data. It estimates the optimal number of clusters to be the place where the gap between the two curves is largest. Essentially this is an automatic way of locating the aforementioned "kink."

Figure 14.11 shows the result of the Gap statistic applied to the simulated data of Figure 14.4. The left panel shows log W_K for k = 1, 2, …, 8 clusters (green curve) and the expected value of log W_K over 20 simulations from uniform data (blue curve). The right panel shows the gap curve, which is the expected curve minus the observed curve. This gives $K^* = 2$, which looks reasonable from Figure 14.4.

Hierarchical Clustering (14.3.12)

The results of applying *K*-means clustering algorithms depend on the choice for the number of clusters to be searched and a starting configuration assignment. In contrast, hierarchical clustering methods do not require such specifications. Instead, they require the user to specify a measure of dissimilarity between (disjoint) groups of observations, based on the pairwise dissimilarities among the observations in the two groups. As the name suggests, they produce hierarchical representations in which the clusters at each level of the hierarchy are created by merging clusters at the next lower level. At the lowest level, each cluster contains a single observation. At the highest level, there is only one cluster containing all of the data.

Strategies for hierarchical clustering divide into two basic paradigms: agglomerative (bottom-up) and divisive (top-down). Agglomerative strategies start at the bottom and at each level, recursively merge a selected pair of clusters into a single cluster. This produces a grouping at the next higher level with one less cluster. The pair chosen for merging consists of the two groups with the smallest intergroup dissimilarity. Divisive methods start at the top and at each level recursively split one of the existing clusters at that level into two new clusters. The split is chosen to produce two new groups with the largest between-group dissimilarity. With both paradigms, there are (N – 1) levels in the hierarchy.

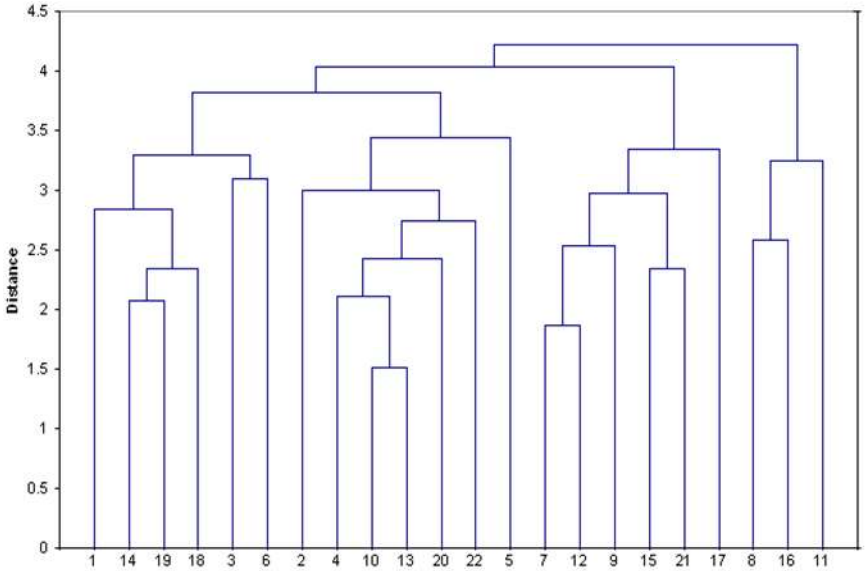
A clustering problem: The Table below gives corporate data on 22 US public utilities. We are interested in forming groups of similar utilities. There are 8 measurements on each utility, as described below.

utility_name	utility	x1	x2	x3	x4	x5	x6	x7	x8
Arizona	1	1.06	9.2	151	54.4	1.6	9077	0	0.628
Boston	2	0.89	10.3	202	57.9	2.2	5088	25.3	1.555
Central	3	1.43	15.4	113	53	3.4	9212	0	1.058
Common	4	1.02	11.2	168	56	0.3	6423	34.3	0.7
Consolid	5	1.49	8.8	192	51.2	1	3300	15.6	2.044
Florida	6	1.32	13.5	111	60	-2.2	11127	22.5	1.241
Hawaiian	7	1.22	12.2	175	67.6	2.2	7642	0	1.652
Idaho	8	1.1	9.2	245	57	3.3	13082	0	0.309
Kentucky	9	1.34	13	168	60.4	7.2	8406	0	0.862
Madison	10	1.12	12.4	197	53	2.7	6455	39.2	0.623
Nevada	11	0.75	7.5	173	51.5	6.5	17441	0	0.768
NewEngla	12	1.13	10.9	178	62	3.7	6154	0	1.897
Northern	13	1.15	12.7	199	53.7	6.4	7179	50.2	0.527
Oklahoma	14	1.09	12	96	49.8	1.4	9673	0	0.588
Pacific	15	0.96	7.6	164	62.2	-0.1	6468	0.9	1.4
Puget	16	1.16	9.9	252	56	9.2	15991	0	0.62
SanDiego	17	0.76	6.4	136	61.9	9	5714	8.3	1.92
Southern	18	1.05	12.6	150	56.7	2.7	10140	0	1.108
Texas	19	1.16	11.7	104	54	-2.1	13507	0	0.636
Wisconsi	20	1.2	11.8	148	59.9	3.5	7287	41.1	0.702
United	21	1.04	8.6	204	61	3.5	6650	0	2.116
Virginia	22	1.07	9.3	174	54.3	5.9	10093	26.6	1.306

X1: Fixed-charge covering ratio (income/debt)
X2: Rate of return on capital
X3: Cost per KW capacity in place
X4: Annual Load Factor
X5: Peak KWH demand growth from 2010 to 2012
X6: Sales (KWH use per year)
X7: Percent Nuclear
X8: Total fuel costs (cents per KWH)

Problem statement: Assume, in a 2D plane we have 22 points, based on the distances given among themselves, we have to develop a dendrogram.

SL	Gr 1	Gr 2	Distance
1	10	13	1.515788
2	7	12	1.872992
3	14	19	2.075641
4	4	10	2.102153
5	15	21	2.345609
6	14	18	2.350194
7	4	20	2.433253
8	7	9	2.530764
9	8	16	2.582103
10	4	22	2.754688
11	1	14	2.847894
12	7	15	2.973068
13	2	4	2.99574
14	3	6	3.099471
15	8	11	3.251878
16	1	3	3.293555
17	7	17	3.349042
18	2	5	3.435891
19	1	2	3.826466
20	1	7	4.034781
21	1	8	4.230111



Reference:

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*: Springer, 2009.

---- x ----