Chapter10. Unsupervised Learning

Supervised learning deals with a response Y and predictors X. Unsupervised learning only has the predictors, with no response variable. This is challenging because there is no way to check our work because there is no true answer.

10.2 Principal Components Analysis

Principal components in regression represent directions where the feature space is highly variable. They also define lines and subspaces that are as close as possible to the data cloud. In an unsupervised approach, principal components can be used to better understand datasets. Suppose we wish to visualize $X_{n\times p}$. Examining two-dimensional scatterplots requires ${}_PC_2$ = p(p−1)/2 plots; for example, p=10 requires 45 plots! We would rather like to find a low-dimensional representation of the data that captures most of the information. Principal components analysis performs this by creating variables that are interesting. The first principal component is the normalized linear combination of the features

$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{p1}X_p$ that has the largest variance. Normalized means the loadings are constrained $\|\Phi_1\|_2$=1, otherwise the variance could be arbitrarily large. We compute the first PC by centering each of the variables to have mean of zero. Then we solve the following optimization problem

$$\begin{aligned} \text{subject to} \quad & ||\phi_1||_2 = 1 \\ \max_{\phi_1} \quad & \frac{1}{n}\sum_{i=1}^{n}\mathbf{X}\phi_1 \end{aligned}$$

The objective could be written as $\frac{1}{n}\sum_{i=1}^{n} z_{ij}^2$. Since the result $\frac{1}{n}\sum_{i=1}^{n} x_{ij}$=0 (scaled predictors), then μ_{z_1}=0 as well. Hence the objective that we are maximizing is just the sample variance of z_1, which is referred to as the scores of the first principal component. This loading vector Φ_1 represents the direction in feature space along which the data vary most. This problem can be solved by eigen decomposition, a standard technique in linear algebra (which is on my list, see Strang's Introduction to Linear Algebra). The second principal component Φ_2 is the linear combination of XΦ_2 that has maximal variance and is uncorrelated with Z_1. It turns out constraining Z_2 to be uncorrelated with Z_1 is equivalant to $\Phi_1 \cdot \Phi_2$=0, they are orthogonal. Once we have the computed principal components, we can plot them against each other to produce low-dimensional views of the data. Geometrically, this amounts to projecting the original data down onto the subspace spanned by $\overline{\Phi}$.

10.2.2 Another Interpretation of Principal Components

We can view a three dimensional dataset projected onto two principal components whose loadings best represent a 2-D planes that minimizes the RSS (accounts for the most variance). In this interpretation, principal components create a low-dimensional linear surface that is closest to the observations. The first principal component loading vector has a very special property: it is the line in p-dimensional space that is closest to the observations (using average squared Euclidean distance). The appeal of this interpretation is clear: we seek a single dimension of data (a line) that lies as close as possible to all the data points. Thus M principal components, subject to the orthogonal constraint, build up a linear hyper-plane of M-dimensions that is close the the observations. This can be written as

$$\mathbf{X}_{n\times p} \approx Z_{n\times m}\phi_{m\times p}$$

In other words, together the M principal component score vectors and M principal component loading vectors can give a good approximation to the data when M is sufficiently large.

10.2.3 More on PCA

- Scaling the Variables

We already mentioned the observations must be centered, therefore, the results obtained from PCA will also depend on whether the variables have been individually scaled. For example: variables measuring *Murder*, *Rape*, and *Assualt*, are reported as number of occurences per 100,000 people, and *UrbanPop* is the percentage of urban population in a state, with variances 18.97, 87.73, 6945.16, and 209.5 respectively. Without scaling these variances, the principal components will skew the variance information based on context rather than content. If *Assault* were measured in occurences per 100 people, the unscaled variance will be scaled down by 1,000. In linear regression, multipling the coefficients by constants does not affect the model, but in PCA scaling does matter. Therefore, we typically scale the variables to have μ=0 and σ=1 before performing PCA. Only in certain settings, where the variables are measured in the same units, should scaling not be performed.

- The Proportion of Variance Explained

The principal component loading vector is unique in direction, not sign $\Phi = -\Phi$. Same is true with the score vectors Z=−Z. If the sign is flipped on both the loading vector and score vector, the signs will cancel and the final product of the two quantities is unchanged.

- Proportion of Variance Explained

How much information is lost by projecting the observations onto the first few principal components? Or, more generally, what is the proportion of variance explained (PVE) by each principal component. The PVE of the mth principal component is

$$\text{PVE} = \frac{\text{Variance of } Z_m}{\text{Total Variance}} = \frac{||\mathbf{X}\phi_m||_2^2}{\text{var}(\mathbf{X})}$$

- Deciding How Many Principal Components to Use

In general $X_{n\times p}$ has min(n−1, p) distinct principal components. But we only want the best ones! But there is no hard criteria for best. We can use a scree plot, plotting the PVE and cumulative PVE and looking for the elbow of the plot. It is ad-hoc, and there isn't a well defined cut-off (like for automation). Generally, we look at the first few principal components for trends, and if there are not any, more principal components will not improve the picture.

10.3 Clustering Methods

Clustering is a broad set of techniques for finding subgroups, or clusters, in a data set. Of course, what does it mean to be mathematically similar or different? Suppose $X_{n\times p}$ describes tissue samples of patients with breast cancer, described by clinical measurements like tumor stage or grade. We may assume the samples are heterogeneous, with unknown subtypes of breast cancer burried in the data. Another application in marketing would be to find groups of people who like similar movies and then recommend movies they haven't seen, but would probably like given the statistics of similar users. Clustering can find these subgroups.

- PCA looks to find a low-dimensional summary of the observations that explains a good fraction of the variance
- Clustering finds homogeneous subgroups amoung the observations.

10.3.1 K-Means Clustering

Partition data into K distinct, non-overlapping clusters. First, specify the number of clusters K, then the K-means algorithm will assign each observation to exactly one of the K clusters. The procedure results from a simple and intuitive mathematical problem. Let $C_1,...,C_K$ denote sets containing observations in each cluster satisfying two properties:

1. $C_1 \cup \ ... \ \cup C_K$ = {1, ... , n} each observation belongs to at least one of the K clusters.
2. $C_k \cap C_{k'} = \varnothing$ for all k≠k' the clusters are non-overlapping: no observation belongs to more than one cluster.

Example: the ith observation is in the kth cluster, then i∈ C_k. The within-cluster-variation should be as small as possible (homogeneous) defined as W(C_k). How do we define this? The most common choice is the squared Euclidean distance

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

This is the pairwise squared Euclidean distances between the observations in the kth cluster, weighted by the number of observations in the kth cluster. We formulate an optimization problem to minimize this quantity summed among all clusters. This is a difficult problem to solve precisely, since there are almost K^n ways to partition n observations into K clusters. But the following algorithm can find a pretty good local optimum. It is guaranteed to decrease the objective at each step.

⇒ **Algorithm K-Means Clustering**

1. Initially assign a random number 1 to K to each of the observations.
2. Iterate until the cluster assignments stop changing:
    a. For each of the K clusters, compute the centroid.
    b. Assign each observation to the closest cluster centroid (where closest is the Euclidean distance)

To understand why this works, we illustrate the following

$$W(C_k) = 2 \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$$

Where $\bar{x}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij}$ is the mean for feature j in cluster C_k. In Step 2(a) the cluster means are the constants that minimize sum-of-squared deviations, and in Step 2(b), reallocating the observations can only improve the variance. The algorithm continues until nothing changes, in other words a local optimum has been reached. Since K-means finds a local rather than a global optimum, and uses a random initialization, it

is important to run the algorithm multiple times to obtain a robust result.

10.3.2 Hierarchical Clustering

Hierarchial clustering helps us choose K using a dendrogram and a bottom-up or agglomerative clustering approach. We start with a dendrogram of all observations, and then move up the tree. Leaves fuse into branches, branches into other branches, and eventually a stump. The higher up the tree, the more different the observations. More precisely: for any two observations, we can look for the point in the tree where branche containing these two observations first fuse. The fusion height measures how different the two observations are. Horizontal similarity has no meaning. A single dendrogram describes the entire series of clusters from K=1, … ,n. The term hierarchial refers to the fact that clusters obtained by cutting the dendrogram at a given height are necessarily nested within the clusters obtained by cutting the dendrogram at any greater height.

The hierarchial clustering dendrogram algorithm is very simple. Define some sort of dissimilarity between each pair of observations (like Euclidean distance). For each iteration, the two observations that are most similar are fused with the branch node measured as the center between the observation pair. There are now n−1 clusters. This continues until the stump.

⇒ **Algorithm Hierarchial Clustering**

1. Begin with n observations and a measure (such as Euclidean distance) of all the ${}_nC_2$ =n(n−1)/2 pairwise dissimilarities. Treat each observation as its own cluster.
2. For i=n, n−1, …, 2:

    a. Examine all pairwise inter-cluster dissimilarities amongthe i clusters and identify the cluster pairs that are least dissimilar (that is, most similar). Fuse these two clusters The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.

    b. Compute the new pairwise inter-cluster dissimilarities amoung the i−1 remaining clusters. ***

One problem is that we have a dissimilarity between observations, but what about clusters and individual observations? We extend the dissimilarity idea using the linkage. The Average and Complete linkages are preferred because they yield the most balanced dendrograms. Centroid linkages can invert whereby two clusters are fused at a height below either of the individual clusters in the dendrogram.

Linkage	Description
Complete	Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in clusters A and B, and record the largest of these dissimilarities (least similar).
Single	Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between observations in clusters A and B, and record the smallest of these dissimilarities (most similar). Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean intercluster dissimilarity. Compute all pairwise dissimilarities between observations in clusters A and B, and record average of these dissimilarities.
Centroid	Dissimilarity between the centroid for Cluster A (a mean vector of length p) and cluster B centroid. Centroid linkage can result in undesirable inversions.

So far our dissimilarity measure has been the Eucidian distance. But other measures can be employed, like correlation-based distance that considers two observations similar if their features are highly correlated. For example, consider an online retailer interested in clustering shoppers based on their past shopping histories in order to identify subgroups of similar shoppers. Suppose the data takes the form of a matrix where rows are shoppers and columns are items available to purchase, and the values are the number of times the shopper has purchased the item. If Euclidian distance is used, shoppers who have bought nothing at all would be grouped together, not entirely desireable. On the other hand, correlation-based distance groups shoppers with similar preferences (similar shoppers, we all want our tribe), ignoring whether some shoppers are high volume customers or not. Scaling is another issue (like number of socks vs. number of computers).