Ch. 10. Unsupervised Learning

Unsupervised learning is a set of statistical tools intended for the setting in which we have only a set of features $X_1,..,X_p$ measured on n observations (and no responses $Y_1,...,Y_n$).
The goal is to discover interesting things about the measurements on $X_1, \ldots, X_p$. Is there an informative way to visualize the dat? Can we discover subgroups among the variables or among the observations?
In this chapter we will focus on two particular types of unsupervised learning: principal components analysis, a tool used for data visualization or data pre-processing before supervised techniques are applied, and clustering, a broad class of methods for discovering unknown subgroups in data.

Principal Components Analysis

We already discussed principal components in chapter 6 in the context of principal components regression.
When faced with a large set of correlated variables, principal components allow us to summarize this set with a smaller number of representative variables that collectively explain most of the variability in the original set.

The principal component directions were presented in chap. 6 as directions in feature space along which the original data are highly variable. These directions also define lines and subspaces that are as close as possible to the data cloud. To perform principal components regression, we simply use principal components as predictors in a regression model in place of the original larger set of variables.

Principal Components Analysis (PCA) refers to the process by which principal components are computed, and the subsequent use of these components in understanding the data. PCA is an unsupervised approach, since it involves only a set of features $X_1,...,X_p$, and no associated response Y.

Apart from producing derived variables for use in supervised learning problems, PCA also serves as a tool for data visualization (visualization of the observations or visualization of the variables).

What Are Principal Components?

Suppose that we wish to visualize n observations with measurements on a set of p features, $X_1, \ldots, X_p$, as part of an exploratory data analysis.
We could do this by examining two-dimensional scatterplots of the data, each of which contains the n observations' measurements on two of the features. However, there are $({}^{p}{}_{2})$ = p(p-1)/2 such scatterplots; for example, with p = 10 there are 45 scatterplots.

Moreover, most likely none of them will be informative since they each contain just a fraction of the total information of the whole dataset.
Clearly, a better method is required to visualize the n observations when p is large. In particular, we would like to find a low-dimensional representation of the data that captures as much of the information as possible.

PCA finds a low-dimensional representation of the data that captures most of the information, then we can plot the observations in this low-dimensional space. The **idea** is that each of the n observations lives in p-dimensional space, but not all of these dimensions are equally interesting. PCA seeks a small number of dimensions that are as interesting as possible, where the concept of **interesting** is measured by the amount that the observations vary along each direction.
Each of the dimensions found by PCA is a linear combination of all the p features.

The first principal component of a set of features $X_1, \ldots, X_p$ is the normalized linear combination of the features
$Z_1 = \Phi_{11}X_1 + \Phi_{21}X_2 + \ldots + \Phi_{p1}X_p$ that has the largest variance.

By normalized we mean that $\sum_{j=1}^{p} \phi_{j1}^{2} = 1$. We refer to the elements $\Phi_{11}, \ldots, \Phi_{p1}$ as the loadings of the first principal component; together, the loadings make up the principal component loading vector $\Phi_1 = (\Phi_{11}, \ldots, \Phi_{p1})^T$.
We constrain the loadings so that their sum of squares is equal to one, since otherwise setting these elements to be arbitrarily large in absolute value could result in arbitrarily large variance.

How to compute: given a n x p data set X, we assume, since we are only interested in variance, that each of the variables in X has been centered to have mean zero (that is, the column means of X are zero). We then look for the linear combination of the sample feature values of the form
$z_{i1} = \Phi_{11}x_{i1} + \Phi_{21}x_{i2} + \ldots + \Phi_{p1}x_{ip}$
that has the largest sample variance, subject to the constraint that $\sum_{j=1}^{p} \phi_{j1}^{2} = 1$

In other words, the first PC loading vector solves the optimization problem

maximize $\{\sum_{i=1}^{n}(\sum_{j=1}^{p} \phi_{j1}x_{ij})^{2}\}$ *subject to* $\sum_{j=1}^{p} \phi_{j1}^{2} = 1$
This problem can be solved via an eigen decomposition.

There is a nice geometric interpretation for the first principal component.
The loading vector defines a **direction** in feature space along which the data varies the most. If we project the n data points $x_1,\ldots,x_n$ onto this direction, the projected values are the principal components scores $z_{11}, \ldots, z_{n1}$ themselves.

After the first principal component Z_1 of the features has been determined, we can find the second principal component Z_2. The second principal component is the linear component of the features that has maximal variance out of all linear combinations that are uncorrelated with Z_1. Constraining Z_2 to be uncorrelated with Z_1 is equivalent to constraining the direction Φ_2 to be orthogonal to the direction Φ_1.

Another Interpretation of Principal Components

The first principal component loading vector has a very special property: it is the line in p-dimensional space that is closest to the n observations (using average squared Euclidean distance as a measure of closeness).
The appeal of this interpretation is clear: we seek a single dimension of the data that lies as close as possible to all of the data points, since such a line will likely provide a good summary of the data.
This notion can be extended beyond just the first principal component. For instance, the first two principal components of a data set span the plane that is closest to the n observations. The first 3 PC, span the 3-dimensional hyperplane that is closest to the n obs, and so forth.

Using this interpretation, together the first M principal component score vectors and the first M principal component loading vectors provide the best M-dimensional approximation to the ith observation x_{ij}.
This representation can be written $x \approx \sum_{m=1}^{M} z_{im\Phi jm}$ assuming the original data matrix X is column-centered.

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. When M = min(n - 1, p), then the representation is exact: $x = \sum_{m=1}^{M} z_{im\Phi jm}$.

More on PCA

We have already mentioned that before PCA is performed, the variables should be centered to have man zero. Furthermore, the results obtained when we perform PCA will also depend on whether the variables have been individually scaled (each multiplied by a different constant). This is in contrast to other learning techniques in which scaling the variables has no effect.

Why does it matter that we scale the variables? If our features are measured in different scales/units (e.g., Kilometers for one feature and meters for another feature, range is similar), this will affect the value of their variance.
example: the range is similar, e.g. 0-5 Kms (0 - 5000 m), the variance of the feature measured in meters will be much higher just because of the choice of unit.

In light of this, we typically scale each variable to have standard deviation one before we perform PCA.
In certain settings, however, the variables may be measured in the same units. In this case we might not wish to scale the variables to have standard deviation one before performing PCA.

The Proportion of Variance Explained

How much of the information in a given data set is lost by projecting the observations onto the first few principal components? That is, how much of the variance in the data is not contained in the first few principal components?
More generally, we are interested in knowing the proportion of variance explained (PVE) by each principal component.
The total variance present in a data set (assuming that the variables have been centered to have mean zero) is defined as

$$\sum_{i=1}^{n} Var(X) = \sum_{ij=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^{2}$$

and the variance explained by the mth principal component is

$$\frac{1}{n} \sum_{i=1}^{n} z_{im}^{2} = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{jm} x_{ij} \right)^{2}$$

therefore the PVE of the mth principal component is given by

$$\frac{\sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{jm} x_{ij} \right)^{2}}{\sum_{ij=1}^{p} \sum_{i=1}^{n} x_{ij}^{2}}$$

The PVE of each principal component is a positive quantity.
In order to compute the cumulative PVE of the first M principal components we can simply sum it over each of the first M PVEs. In total there are min(n - 1,p) principal components, and their PVEs sum to one.

Clustering Methods

Clustering refers to a very broad set of techniques for finding subgroups or clusters in a data set. When we cluster the observations of a data set, we seek to partition them into distinct groups so that the observations within each group are quite similar to each other, while observations in different groups are quite different from each other.
We must define what it means for two or more observation to be different or similar. Indeed, this is a domain-specific consideration that must be made based on knowledge of the data being studied.

Both clustering and PCA seek to simplify the dat via a small number of summaries but their mechanism are different:

- PCA looks to find a low-dimensional representation of the observations that explain a good fraction of the variance;
- Clustering looks to find homogeneous subgroups among the observations.

K-Means Clustering

K-means clustering is a simple and elegant approach for partitioning a data set into K distinct, non-overlapping clusters. To perform K-means clustering, we must first specify the desired number of clusters K; then the K-means algorithm will assign each observation to exactly one of the K clusters.

K-Means Clustering Algorithm:

1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.
2. Iterate until the cluster assignments stop changing (converge):
    a. For each of the K clusters, compute the cluster centroid. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.
    b. Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).

This algorithm is guaranteed to decrease the value of the objective (intra cluster distance).

When the result no longer changes, a local optimum has been reached.
K-means clustering derives its name from the fact that the cluster centroids are computed as the mean of the observations assigned to each cluster.

Because the K-means algorithm finds a local rather than a global optimum, the results obtained will depend on the initial (random) cluster assignment of each observation in step 1 of the algorithm.
For this reason, it is important to run the algo multiple times from different random initial configurations. Then one selects the best solution, i.e. that for which the intra cluster distance is smallest.

As we have seen, to perform K-means clustering, we must decide how many clusters we expect in the data. The problem of selecting K is far from simple.

Hierarchical Clustering

One potential disadvantage of K-means clustering is that it requires us to pre-specify the number of clusters K. Hierarchical clustering is an alternative approach that does not require that we commit to a particular choice of K.
Hierarchical clustering has an added advantage over K-means clustering in that it results in an attractive tree-based representation of the observations, called a dendrogram.

In this section we describe bottom-up or agglomerative clustering. This is the most common type of hier. clustering, and refers to the fact that the dendrogram is built starting from the leaves and combining clusters up to the trunk.

Hierarchical Clustering Algorithm

1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2}$ = 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 among the i clusters and identify the pair of clusters 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 among the i - 1 remaining clusters.

This algorithm seems simple enough, but one issue has not been addressed. We have a concept of the dissimilarity between pairs of observations, but how do we define the dissimilarity between two clusters if one or both of the clusters contains multiple observations?
The extension of this concept from two obs to two groups of obs (clusters) is achieved by developing the notion of linkage which defines the dissimilarity between two groups of observations.

The four most common types are complete, average, single, and centroid.
Average and complete are generally preferred over single linkage as they tend to yield more balanced dendrograms.
The dissimilarities computed in step 2(b) will depend on the type of linkage used, as well as on the choice of dissimilarity measure.

Choice of Dissimilarity Measure

Sometimes dissimilarity measures different from Euclidean distance might be preferred. For example, correlation-based distance considers two observations to be similar if their features are highly correlated, even though the observed values may be far apart in terms of Euclidean distance. This is an unusual use of correlation, which is normally computed between variables; here it is computed between the observation profiles for each pair of observations. Correlation-based distance focuses on the **shapes** of observation profiles rather than their magnitudes.

The choice of dissimilarity measure is very important, as it has a strong effect on the resulting dendrogram. In general, careful attention should be paid to the type of data being clustered and the scientific question at hand.

In addition to carefully selecting the dissimilarity measure used, one must also consider whether or not the variables should be scaled to have standard deviation one before the dissimilarity between the observations is computed. Again, it will depend on the type of data and questions we are facing.

Practical Issues in Clustering

In order to perform clustering, some decisions must be made:

- Should the observations or features first be standardized in some way? For instance, maybe the variables should be centered to have mean zero and scaled to have standard deviation one.
- In the case of hierarchical clustering,
    - What dissimilarity measure should be used?
    - What type of linkage should be used?
    - Where should we cut the dendrogram in order to obtain clusters?
- In the case of K-means clustering, how many clusters should we look for in the data?

Each of these decisions can have a strong impact on the results obtained.
In **practice** we try several different choices, and look for the one with the most useful or interpretable solution. With these methods there is no single right answer -- any solution that exposes some interesting aspects of the data should be considered.

Validating the Clusters Obtained → Check ESL

Both K-means and hierarch. will assign each observation to a cluster. However, sometimes this might not be appropriate. For instance, suppose that most of the observations truly belong to a small number of (unknown) subgroups, and a small subset of the observations are quite different from each other and from all other observations.
Then, since K-means and hierarch. clustering force **every** observation into a cluster, the clusters found may be heavily distorted due to the presence of outliers that do not belong to any cluster.
Mixture models are an attractive approach for accommodating the presence of such outliers. These amount to a soft version of K-means clustering → ESL.

In addition, clustering methods generally are not very robust to perturbations to the data