10. Unsupervised Learning

Unsupervised learning: A set of statistical tools intended for the setting in which we have only a set of features X1, X2, …,Xp measured on n observations

10.1 The Challenge of Unsupervised Learning

Unsupervised learning is often much more challenging. The exercise tends to be more subjective, and there is no simple goal for the analysis, such as prediction of a response.

1. Unsupervised learning is often performed as part of an exploratory data analysis. Furthermore, it can be hard to assess the results obtained from unsupervised learning methods.

10.2 Principal Component Analysis

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.

Principal Component analysis refers to the process by which principal components are computed, and the subsequent use of these components in the understanding of the data.

10.2.1 What Are Principal Components?

PCA finds a low-dimensional representation of a data set that contains as much as possible of the variation.

1. 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 dimension.
2. Each of the dimensions found by PCA is a linear combination of the p features.
3. The first principal component of a set of features X1, X2, …Xp is the normalized linear combinations of the features

Logo

Description automatically generated

By normalized, we meant that

A picture containing text

Description automatically generated

We refer to thetas as the loadings of the first principal component, together, the loadings make up the principal component loading vector.

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 an arbitrarily large variance.

The first principal component loading vector solves the optimization problem:

Diagram

Description automatically generated

Geometric Interpretation for the first principal component;

1. The loading vector defines a direction in feature space which the data vary the most.
2. The second principal component is the linear combination of X1,…Xp that has a maximal variance out of all linear combinations that are uncorrelated with Z1.

Once we have produced the principal components, we can plot them against each other in order to produce low-dimensional views of the data.

1. Geometrically, This amounts to projecting the original data down onto the subspace spanned by

Biplot: displays both the principal component scores and the principal component loadings

1. The principal component directions are the ordered sequence of eigenvectors of the matrix and the variance of the components are the eigenvalues. There are at most min(n-1,p) principal components

10.2.2 Another Interpretation of Principal Components

Principal components provide low-dimensional linear surfaces that are closest to the observation

1. The first principal component loading vector has a very special property:
2. 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)
3. 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.
4. The first M principal component score vectors and the first M principal component loading vectors provide the best M-dimensional approximation(in terms of Euclidean distance) to the ith observation xij.

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.

Text

Description automatically generated

10.2.3 More on PCA

Scaling the Variables

The results obtained hen we perform PCA will depend on whether the variables have been individually scaled(each multiplied by a different constant).

1. We typically scale each variable to have standard deviation one before we perform PCA

Uniqueness of the Principal Components

Each principal component loading vector is unique, up to a sign flip

1. The signs may differ because each principal component loading vector specifies a direction in p-dimensional space: flipping the sign has no effect as the direction does not change.
2. Similarly, the score vectors are unique up to a sign flip, since the variance of Z is the same as the variance of -Z.
3. If the sign is flipped on both the loading and score vectors, the final product of the two quantities is unchanged.

The Proportion of Variance Explained

The total variance present in a data set(assuming that the variables have been centered to have mean zero) is defined as:

A close up of a clock

Description automatically generated

And the variance explained by the mth principal component is:

A picture containing diagram

Description automatically generated

Therefore, the PVE of the mth principal component is given by:

Text

Description automatically generated

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 over each of the first M PVEs.

Deciding How Many Principal Components to Use

In general, a n\*p matrix X has min(n-1,p) distinct principal components.

1. We would like to use the smallest number of principal components required to get a good understanding of the data.

We typically decide on the number of principal components required to visualize the data by examining a scree plot,

1. We choose the smallest number of principal components that are required in order to explain a sizable amount of the variance in the data.
2. We are looking for a point at which the proportion of variance explained by each subsequent principal component drops off.—This is referred to as an elbow in the scree plot.
3. In practice, we tend to look at the first few principal components in order to find interesting patterns in the data. If no interesting patterns ate found in the first few principal components, then further principal components are unlikely to be of interest
4. In supervised setting: we can treat the number of principal component score vectors to be used in the regression as a tunning parameter to be selected via cross-validation or a related approach.

10.3 Clustering Methods

Clustering refers to a very broad set of techniques for finding subgroups, or clusters in a dataset.

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

Two best known clustering approaches:

1. K-means clustering
   1. In K-means clustering, we seek to partition the observations into a prespecified number of clusters
2. Hierarchical clustering
   1. We do not in advance how many clusters we want: we end up with a tree-like visual representation of the observations, called a dendrograms that allows us to view at once the clustering obtained for each possible number of clusters.

10.3.1 K-Means Clustering

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

Algorithm:

Let C1,…,Ck denote sets containing the indices of the observations in each cluster, these sets satisfy two properties:

1. Each observation belongs to at least one of the K clusters
2. The clusters are non-overlapping: no observation belongs to more than one cluster
3. A good clustering is one for which the within-cluster variation is as small as possible. We solve the problem:

A picture containing text

Description automatically generated

The formula states that we want to partition the observations into K clusters such that the total within-cluster variation, summed over all K clusters, is as small as possible

1. Square Euclidean distance:

A close up of a clock

Description automatically generated

The within-cluster variation for the kth cluster is the sum of all of the pairwise squared Euclidean distances between the observations in the kth cluster, divided by the total number of observations in the kth cluster.

1. Optimization problem for K-means clustering:

A close up of a clock

Description automatically generated

A picture containing text

Description automatically generated

1. As the algorithm is run, the clustering obtained will continually improve until the result no longer changes, a local optimum has been reached.
2. When the result no longer changes, a local optimum has been reached.

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.

1. It is important to run the algorithm multiple times from different random initial configurations. Then selects the best solution, i.e., that for which the objective is smallest
2. To perform K-means clustering, we must decide on how many clusters we expect in the data.

10.3.2 Hierarchical Clustering

Bottom-up agglomerative clustering: A dendrogram is built starting from the leaves and combining clusters up to the trunk.

Interpreting a Dendrogram

1. Each leaf represents an observations.
2. As we move up the tree, some leaves begin to fuse into the branches. These correspond to observations that are similar to each other.
3. As we move higher up the tree, branches themselves fuse, either with leaves or other branches.
4. The earlier fusions occur, the more similar the groups of observations are to each other
5. Observations that fuse later can be quite diiference
6. For any two observations, we can look the point in the tree where branches containing these two observations are first fused.
   1. The height of this fusion, as measured on the vertical axis, indicates how different two observations are
   2. The observations that fuse at the very bottom of the tree are quite similar to each other, whereas observations that fuse close to the top of the tree will tend to be quite differen.
   3. We cannot draw conclusions about the similarity of the two observations based on the proximity along the horizontal axis
   4. We draw conclusions about the similarity of two observations based on the location on the vertical axis where branches those two observations first are fused.

Identifying clusters on the basis of a dendrogram:

1. The height of the cut to the dendrogram serves the same role as the K in K-means clustering: it controls the number of clusters obtained
2. One single dendrogram can be used to obtain any number of clusters

The term hierarchical 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.

* 1. On an arbitrary dataset, this assumption of hierarchical structure is unrealistic.

The Hierarchical Clustering Algorithm:

* + 1. We begin by defining some sort of dissimilarity measure between each pair of observations. Most of the time, Euclidean distance is used.
    2. The algorithm proceeds iteratively. Starting out at the bottom of the dendropgram, each of the n observations is treated as its own clusters. The two clusters that are most similar to each other are then fused so that there are now n-1 clusters.
    3. Next, two clusters that are most similar to each other are fused together again, so there are now n-2 clusters.
    4. The algorithm proceed in this fashion until all of the observations belong to one single cluster, and the dendrogram is complete.

Text

Description automatically generated

Linkage Classifications:

Table

Description automatically generated

1. Complete and average linkage are generally preferred over single linkage, as they tend to yield more balanced dendrograms.
2. Centroid linkage is often used in genomics, but suffers from a major drawback in that inversion can occur, whereby two clusters are fused at a height below either of the individual clusters in the dendrogram.
3. The result of the dendrogram typically depends strongly on the type of the linkage used

Choice of Dissimilarity

Correlation-based distance considers two observations to be similar if their features are highly correlated, even though the observed values may be fat apart in terms of Euclidean distance.

1. Correlation-based distance focuses on the shapes of observation profiles rather than their magnitudes.

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 observation is computed.

1. If the variables are scaled to have standard deviation one before the inter-observation dissimilarities are computed, then each variable will in effect be given equal importance in the hierarchical clustering performed.

10.2.2 Practical Issues in Clustering

Small Decisions with Big Consequence:

1. In order to perform clustering, some decisions must be made:
   1. Should the observations or features first be standardized in some way
   2. In the case of hierarchical clustering:
      1. What dissimilarity measure should be used
      2. What type of linkage should be used
      3. Where should we cut the dendrogram in order to obtain the clusters
   3. In the case of K-means clustering, how many clusters should we look for in the data

Other considerations in Clustering:

If most of the observations truly belong to a small number of (unknow) subgroups, and a small subset of the observations are quite different from each other and from the other observations.

1. Since K-means and hierarchical 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.

Clustering methods generally are not very robust to perturbations to the data.

10.4 Lab1 : Principal Components Analysis

The apply() function allows us to apply a function-in this cane, the mean( ) function-to each row or column of the data set.

1. The second input here denotes whether we wish to compute the mean of the rows,1, or the columns,2

Graphical user interface, application

Description automatically generated

Note that the apply() function applyText

Description automatically generated

1. It is important to standardize the variables to have mean zero and standard deviation one before performing PCA

We now perform principal component analysis using the prcomp() function

By default, the prcomp() function centers the variables to have mean zero. By using the scale=True, we scale the variables to have standard deviation



Graphical user interface, text, application

Description automatically generated

1. Center and scale correspond to the means and standard deviations of the variables that were used for scaling prior to implementing PCA.
2. The rotation matrix provided the principal component loadings; each column contains the corresponding principal component loading vector.

Text

Description automatically generated

Using the prcomp() function, we do not need to explicitly multiply the data by the principal component loading vectors in order to obtain the principal component score vector. The matrix x has as its columns the principal component score vectors. That is, the kth columns is the kth principal component score vector.

A picture containing diagram

Description automatically generated

We can plot the first two principal components as follows:



Chart, scatter chart

Description automatically generated

Since the principal components are only unique up to a sign change, so we can reproduce the Figure by making a few small changes:

A picture containing text

Description automatically generated

Chart, scatter chart

Description automatically generated

Prcomp() function also outputs the standard deviation of each principal component.

A picture containing text

Description automatically generated

The variance can be explained by squaring the standard deviation:

A picture containing text

Description automatically generated

To compute the PVE, we simply divide the variance explained by each principal component by the total variance explained by all four principal components

A picture containing calendar

Description automatically generated

Plot the PVE explained by each component, as well as the cumulative PVE:

A picture containing text

Description automatically generated

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

Cumsum( ) computes the cumulative sum of the elements of a numeric vector.

10.5 Lab2:Clustering

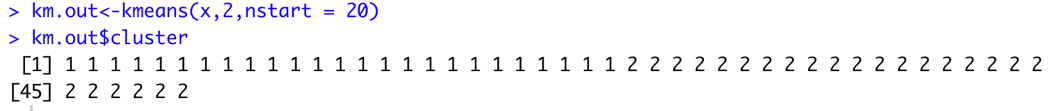
10.5.1 K-Means Clustering

1. simulates two data sets

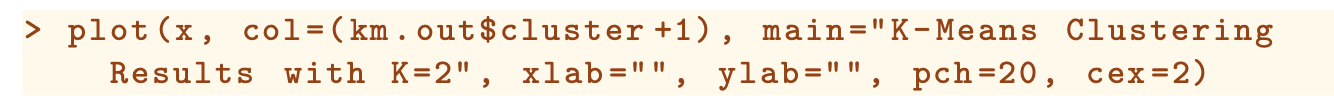
Text

Description automatically generated

1. Performs K-means clustering with K=2



Plot the data, with each observation colored according to its cluster assignment



Chart, scatter chart

Description automatically generated

Perform K-means clustering with k=3

To perform k-means( ) function in R with multiple initial cluster assignments, we use the nstart argument. If a value of nstart greater than one is used, the K-means clustering will be performed using multiple random assignments, and the kmeans function will report only the best result.

Graphical user interface, text

Description automatically generated

The km.tot.withinss is the total within cluster sum of squares which we seek to minimize by performing K-means clustering. The individual within-cluster sum-of -squares are contained in vector

Km.out$withinss

1. Strongly recommend running K-means clustering with a large value of nstart, such as 20 or 50, since otherwise an undesirable local optimum may be obtained.
2. When performing K-means clustering, in addition to using multiple initial cluster assignments, it is also important to set a random seed using the
   1. Set.seed() function

In this way, the initial cluster assignments in Step 1 can be replicated, and the K-means output will be fully reproducible

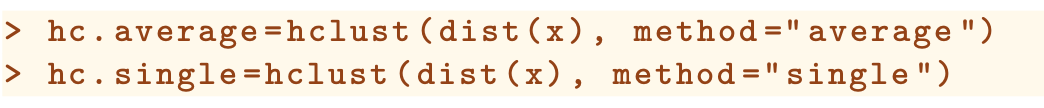
10.5.2 Hierarchical Clustering

Clustering observations using complete linkage, with Euclidean distance as the dissimilarity measure.

Graphical user interface, text, application

Description automatically generated

Perform hierarchical clustering with average or single linkage:

  
Plot the dendrogram using the plot() function. The number at the bottom identify each observation

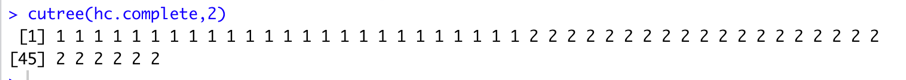
A picture containing text

Description automatically generated

A picture containing chart

Description automatically generated

To determine the cluster labels for each observation associated with a given cut of the dendrogram, we can use the cutree( ) function



To scale the variables before performing hierarchical clustering of the observations, we use the scale() function:



A picture containing chart

Description automatically generated

Correlation-based distance can be computed using as.dist() function, which converts an arbitrary square symmetric matrix into a form that hclust function recognizes as a distance matrix

A picture containing text

Description automatically generated

Chart, box and whisker chart

Description automatically generated

10.6 Lab: NCI60 Data Example

NCI60 microarray data:

Consists of 6,830 gene expression measurements on 64 cancer cell lines

1. Each cell line is labeled with a cancer type. We do not make use of the cancer types in performing PCA and clustering. After performing PCA and clustering, we will check to see the extent to which these cancer types agree with the results of these unsupervised techniques

A picture containing text

Description automatically generated

Examine the cancer types for the cell lines

A picture containing text

Description automatically generated

10.6.1 PCA on the NCI60 data

Perform PCA on the data, after scaling the variables to have standard deviation one.



Plot the first few principal component score vectors in order to visualize the data. We first create a simple function that assigns a distinct color to each element of a numeric vector. The function will be assigned a color to each of the 64 cell lines, based on the cancer type to which corresponds.

A picture containing graphical user interface

Description automatically generated

Rainbow() function takes as its argument a positive integer, and returns a vector containing that number of distinct colors.

Plot the principal component score vectors.

Chart, scatter chart

Description automatically generated

We can obtain a summary of the proportion of variance(PVE) of the first few principal components using the summary() method for a prcomp object

A close up of a newspaper

Description automatically generated

Using the plot() function, we can also plot the variance explained by the first few principal components

A picture containing text, icon

Description automatically generated

Chart, bar chart, histogram

Description automatically generated

Plot the PVE of each principal component(i.e. a scree plot) and the cumulative PVE of each principal component

A picture containing text

Description automatically generatedChart, histogram

Description automatically generated

10.6.2 Clustering the Observations of the NCI60 Data

1. Standardize the variables to have mean zero and standard deviation. (This step is optional and should be performed only if we want each gene to be on the same scale)

2. Perform hierarchical clustering of the observations using complete, single, and average linkage. Euclidean distance is used as the dissimilarity measure

Text

Description automatically generated

A picture containing schematic

Description automatically generated

From the graph, we see that the choice of linkage certainly does affect the results obtained.

1. Typically, a single linkage will tend to yield trailing clusters: very large clusters onto which individual observations attach one-by-one
2. Complete and average linkage tend to yield more balance, attractive clusters. Complete and average linkage are generally preferred to single linkage.

We can cut dendrogram at the height that will yield a particular number of clusters

A picture containing text

Description automatically generated

Table

Description automatically generated

Plot the cut on the dendrogram that produces these clusters

Text

Description automatically generated

A picture containing diagram

Description automatically generated

1. The abline() function draws a straight line on top of any existing plot in R.
2. The argument h=139 plots a horizontal line at height 138 on the dendrogram

Printing the output of the hclust gives a brief summary of the object:

A picture containing text

Description automatically generated

Compare hierarchical clustering results compare to what we get if we perform K-means clustering with K=4

A picture containing table

Description automatically generated

Rather than performing hierarchical clustering on the entire data matrix, we can simply perform hierarchical clustering on the first few principal component score vectors

A picture containing text

Description automatically generated

A picture containing chart

Description automatically generated