Unsupervised methods

Paul Martins

Notes from ISLR book

Contents

1 Principal Components Analysis 1

1.1 Percentage of variance explained 1

2 Clustering 1

2.1 K-means 1

2.2 Hierarchical 2

1. Principal Components Analysis

PCA aims at reducing the dimensionality of a dataset while minimising the loss of information (ie keep high variance). To find the **first** principal component Z_1 of a set of features $X_1,\dots,X_p$, we first need to centre[1] all the X_j and look for a linear combination of the form

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + \cdots + \phi_{p1}x_{ip} \ , \sum_{j=1}^{p} \phi_{j1}^2 = 1 \qquad (1)$$

that has the largest variance.[2] The coefficients ϕ_{1j} are called loadings of PC1, they form the loading vector ϕ_1 and z_{i1} are the scores of PC1.

The loading vector ϕ_1 defines the direction in the feature space along which the data vary the most. Projecting the n data points $x_1,\dots,x_n$ into ϕ_1 gives the scores $z_{11},\dots,z_{n1}$.

We proceed the same way for finding PC2, with the added constraint that Z_2 and Z_1 are uncorrelated, which means $\phi_2 \perp \phi_1$. Figure 1 shows the projection of a 3D dataset into the plane defined by (ϕ_1, ϕ_2).

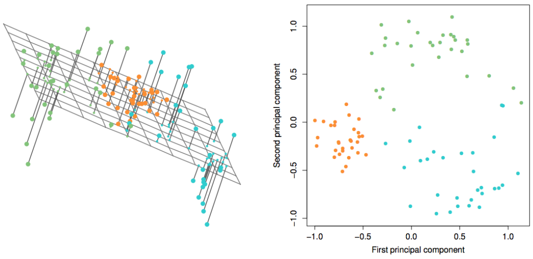

Figure 1. Projection into principal components plane: it minimises the sum of squared distances from each point to the plane.

It is possible to visualise the projected data and the loading vectors into a single biplot as in Figure 2, that highlight the importance of scaling the features.

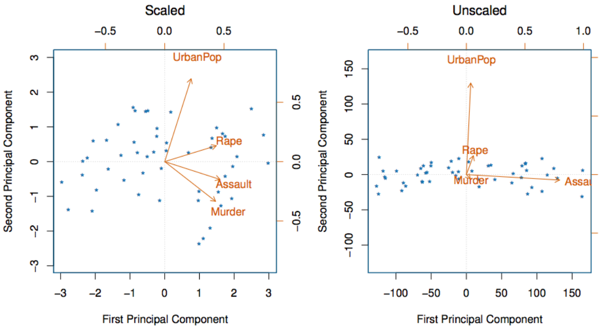

Figure 2. Biplots with scaled (left) and unscaled (right) features.

1.1 Percentage of variance explained

In order to decide how many principal components are necessary, we need to find out the proportion of variance explained (PVE) by each component.

The total variance (assuming features are centred) is

$$\sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2 \qquad (2)$$

and the variance explained by the mth PC is

$$\frac{1}{n}\sum_{i=1}^{n} z_{im}^2 = \frac{1}{n}\sum_{i=1}^{n}\left[\sum_{j=1}^{p}\phi_{jm}x_{ij}\right]^2 \qquad (3)$$

Hence the PVE is

$$\frac{\sum_{i=1}^{n}\left[\sum_{j=1}^{p}\phi_{jm}x_{ij}\right]^2}{\sum_{j=1}^{p}\sum_{i=1}^{n}x_{ij}^2} \qquad (4)$$

From here we can build a scree plot as shown in Figure 3.

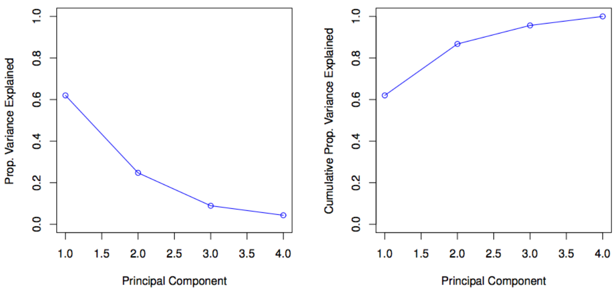

Figure 3. Left: scree plot of % of variance explained by each PC. Right: cumulative scree plot

2. Clustering

Clustering looks to find homogenous subgroups among the observations using some measure of similarity.

[1] rescale so that the mean is 0

[2] This can be viewed as an optimisation problem: $\underset{\phi_{11},\dots,\phi_{p1}}{\text{Maximize}}\left\{\frac{1}{n}\sum_{i=1}^{n}\underbrace{(\sum_{j=1}^{p}\phi_{j1}x_{ij})^2}_{z_{i1}^2}\right\}$ subject to $\sum_{j=1}^{p}\phi_{j1}^2 = 1$

2.1 K-means

K-means clustering splits the observations into K subgroups C_k ($k \in 1,\dots,K$) found by minimising the within-cluster variations $W(C_k)$. Note that all observations belong to exactly one cluster. If we use the squared Euclidean distance as similarity measure, we have

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \tag{5}$$

with $|C_k|$ the number of obs in C_k. Hence the clusters are found by solving the optimisation problem

$$\underset{C_1,\dots,C_k}{\text{Minimize}} \left\{ \sum_{k=1}^{k} W(C_k) \right\} \tag{6}$$

This is hard to find the global minima, but a local minima can be easily reached by the following algorithm (illustrated in Figure 4.

1 Randomly assign a number $1,\dots,K$ to each observation.

2 Iterate until the clusters stop changing:

  a For each cluster C_k, compute the centroid[3]

  b Assign each obs to the cluster whose centroid is the closest.

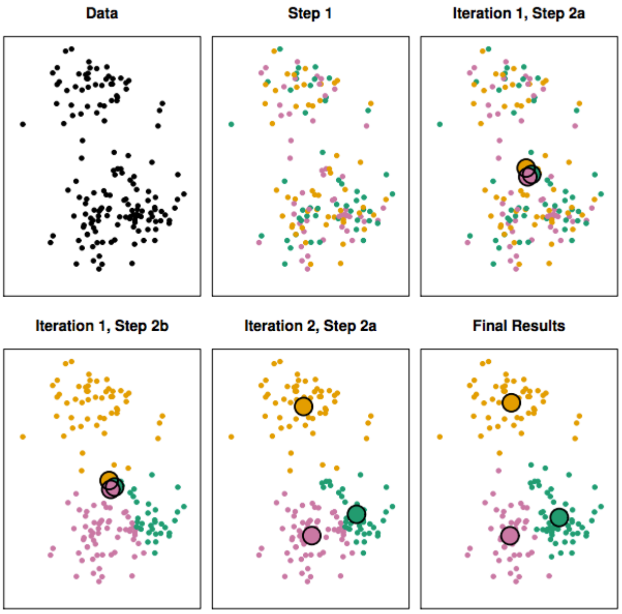

Figure 4. Illustration of the K-means algorithm.

Because there is no guarantee of reaching a global minima, the final clusters will depend on the initial random assignment. Therefore it is best to run the clustering several times and select the configuration which has the lowest objective value.

2.2 Hierarchical

If we don't know the exact number of cluster wanted, we can use hierarchical clustering, a bottom-up agglomerative approach represented by a dendrogram. Each leaf at the bottom represent a single observation. The leaves that fuse at the bottom of the tree are closer to each other than branches that fuse at the top, but it's not because 2 leaves are next to each other than they are similar (Figure 5)

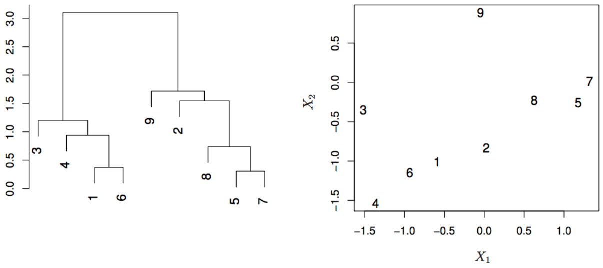

Figure 5. Even if point 9 seems close to point 2 on the dendrogram, it is not in the feature space.

To obtain the cluster, we can cut horizontally the dendrogram: the higher the cut, the less cluster we obtain as in Figure 6

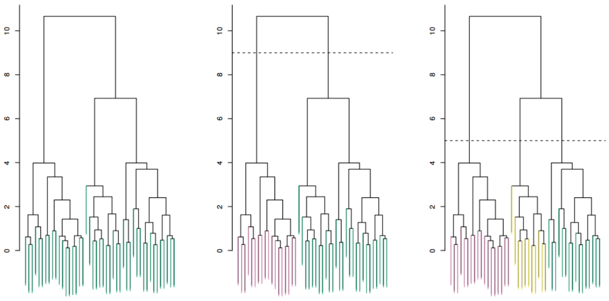

Figure 6. Illustration of hierarchical clustering

[3] the vector of p feature means for obs in cluster k