Lecture 6 - Unsupervised Learning

Sondre Elstad

2022-06-25

Introduction

So far we have talked a lot about supervised learning. We have talked about regression and classification. Linear and logistic regression are in no way techniques that solely belong to machine learning. Neither have they become particularly more popular as a result of machine learning. Linear regression is one of the most basic and most used statistical models in the world, and many students learn about them already at high school. If nothing else, linear regression is usually one of the first models taught in introductory courses in statistics. Logistic regression might be a bit less popular than linear regression (then, again, most statistical concepts and models are). However, it is in no way a novelty.

One of the truly revolutionary things about machine learning, however, is the use of unsupervised learning techniques. This is often much more challenging than supervised learning. There might not be a clear predetermined goal for the analysis, rather than "looking for some structure in the data". There is, in a way, a complementarity of supervised and unsupervised learning. They might support one another. A gargantuan dataset might be 'left to its own devices' in that we use an unsupervised learning technique to find clusters or to reduce the dimensionality of the feature space. This structure could then guide the choice of supervised learning model. It could suggest a natural set of predictors. It could even guide what kind of questions we ask of the data. In this way, unsupervised learning has a natural place in exploratory data analysis.

Larger and larger data sets are both a curse and a blessing at the same time. It is a curse because larger data sets does not always imply structure and clear information. It is a blessing because there might be a lot of information in the data - if we can find it. Unsupervised learning can aid us in this regard. In this lecture we will briefly review two basic techniques of unsupervised learning - principal component analysis and K-means clustering. Principal component analysis helps us in reducing the dimensionality of the feature space, but what does this mean? Let's say that you have a dataset with 100 variables. If you were to draw a bivariate scatter plot for each pair of all of these variables, in the hope of uncovering some interesting structure, you would need a lot of scatter plots! In general, there would be $\frac{1}{2}p(1+p)$ number of scatter plots when there are p predictors and 1 response variable. When you have 99 predictors, you would have a total of 4950 scatter plots to inspect. Principal component analysis allows us to reduce this number significantly by creating linear combinations of the predictors that maximize variance. Let's take a closer look at how this is done!

Principal Component Analysis

We have a $n \times p$ data set $\mathbf{X}$ and we want to compute the first principal component. Assume that each of the variables in $\mathbf{X}$ has been centered (mean zero). It follows that each column in $\mathbf{X}$ has mean 0. We look for a linear combination of the sample predictors, and this takes the following form:

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + ... + \phi_{p1}x_{ip}.$$

Since the mean has been set to zero, we are able to isolate the variance. The underlying idea of Principal Component Analysis (PCA) is to extract linear combinations of the predictors which explains the largest possible fraction of the overall variation in Y. Recall that if we have predictors which explains a large fraction of the overall variation in Y, we have a large R^2 and we have a low RSS. The linear combination of the predictors that we are looking for is therefore the one with the largest variance and which is subject to the constraint $\sum_{j=1}^{p} \phi_{j1}^2 = 1$. This amounts to solving the following optimization problem

$$\max_{\phi_{11},\ldots,\phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} \phi_{j1}^2 = 1.$$

As $\frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0$, the average of the $z_{i1}, ..., z_{n1}$ must be zero as well. Thus, what we are maximizing above is the sample variance of the n z_{i1}. We refer to $z_{11}, ..., z_{n1}$ as the scores of the first principal component. The vector of $\phi_{i1}, ..., \phi_{p1}$ is called the loadings vector.

We can illustrate this idea using data on the number of arrests, per 100,000 residents, in different U.S. states. We have data on three crimes: assault, murder and rape. We also record the percent of the population living in urban areas, UrbanPop. The score vectors have length 50 ($= n$) and the loadings vectors have length 4 ($= p$). The chart plots the first two principal components.

Clustering

There are many different kinds of clustering. I will only review K-means clustering. The basic idea of clustering is to find subgroups (i.e., clusters) in a data set. We hope to achieve a clustering in which observations that belong to the same cluster resembles each other. On the other hand, we want observations in one cluster to be different from those in other clusters. K-means clustering partitions the data set into K distinct and non-overlapping clusters. We start by specifying the desired number of clusters K. The below chart illustrates how choice of K affects the clustering of a simulated data set.

We must define a measure $W(C_k)$. This is the within-cluster variation and it measures to what extent observations within a cluster differ from one another. We want to solve the following minimization problem:

$$\min_{C_1,\ldots,C_K} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$$

Hence, now we want to minimize variance! There are many different candidates for $W(C_k)$. The most common choice is the squared Euclidean distance:

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

$|C_k|$ denotes the number of observations in the kth cluster.

R Lab (ISL 2017)

Let's start with some simple K-means clustering. The function kmeans() performs K-means clustering in R. We begin with a simple simulated example in which there truly are two clusters in the data: the first 25 observations have a mean shift relative to the next 25 observations.

```
set.seed(2)
x <- matrix(rnorm(50 * 2), ncol = 2)
x[1:25, 1] <- x[1:25, 1] + 3
x[1:25, 2] <- x[1:25, 2] - 4
```

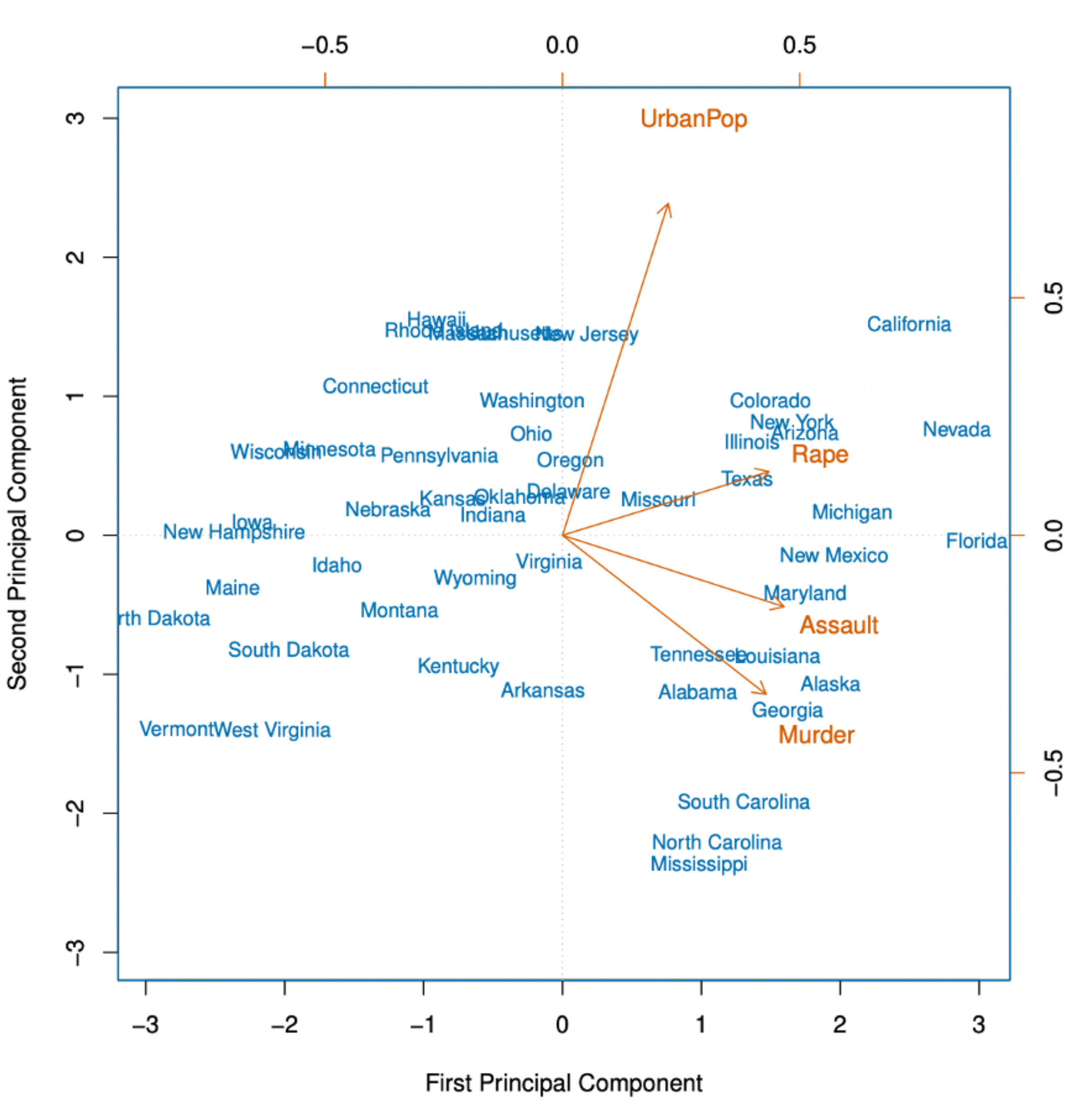


Figure 1: The first two principal components for U.S. arrests data. The blue state names represent scores for the first two principal components. Orange arrows indicate first two principal component loading vectors. This is known as a biplot, as it displays both the scores and the loadings for the principal components. Source: ISL (2017).

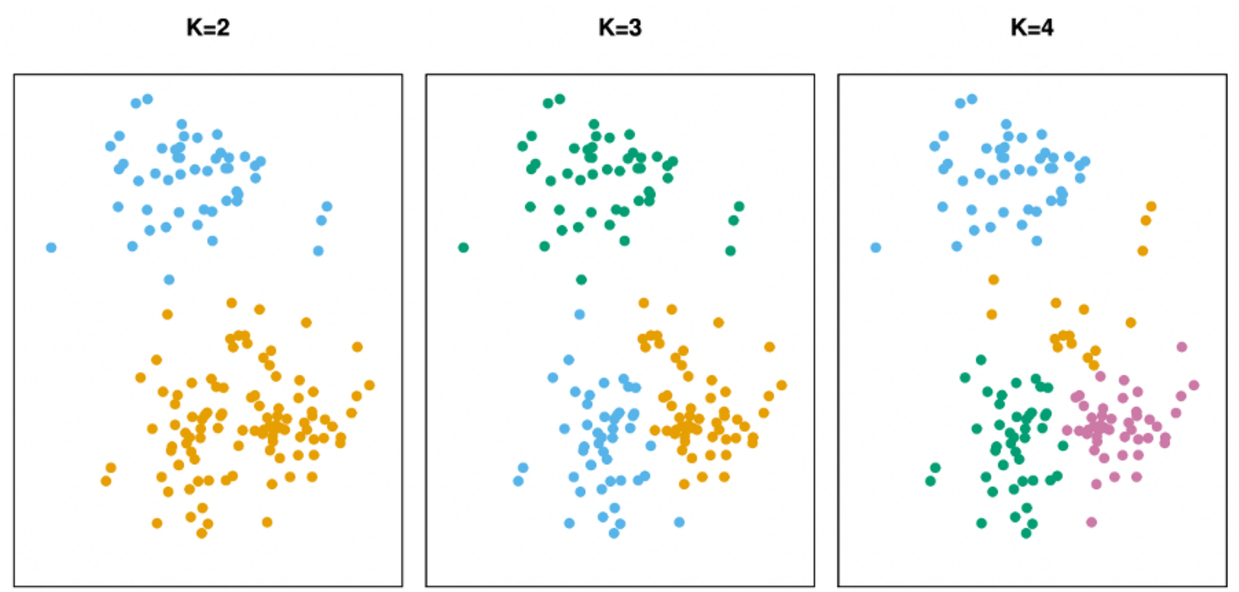


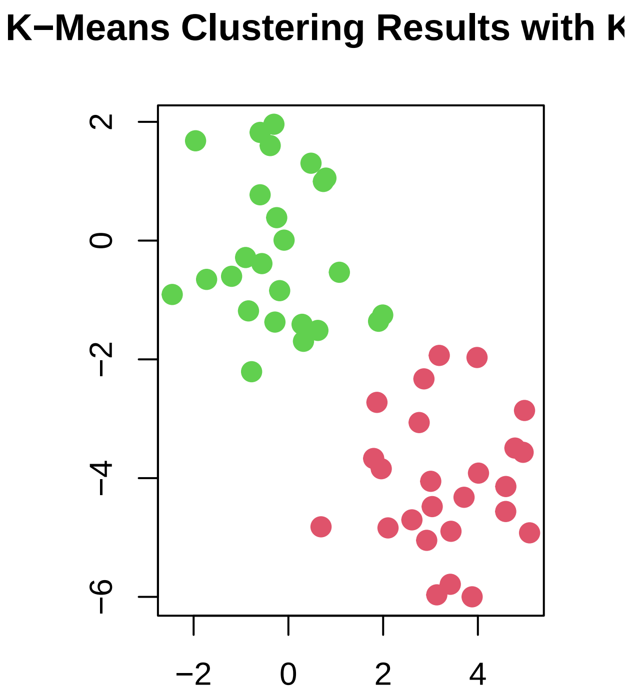
Figure 2: A simulated data set with 150 observations. The chart illustrates the result of applying different number of clusters to the data set. Observations with the same colour belong to the same cluster. There is no ordering of the clusters and so the clustering is arbitrary. Source: ISL (2017).

We now perform K-means clustering with $K = 2$.

```
km.out <- kmeans(x, 2, nstart = 20)
```

We can plot the data, with each observation colored according to its cluster assignment.

```
par(mfrow = c(1, 2))
plot(x, col = (km.out$cluster + 1),
     main = "K-Means Clustering Results with K = 2",
     xlab = "", ylab = "", pch = 20, cex = 2)
```



We illustrate PCA on the NC160 cancer cell microarray data, which consists of 6,830 gene expression measurements on 64 cancer cell lines.

```
library(ISLR2)
nci.labs <- NCI60$labs
nci.data <- NCI60$data
```

Each cell line is labeled with a cancer type, given in nci.labs. We do not make use of the cancer types in performing PCA, as these are unsupervised techniques. We begin by examining the cancer types for the cell lines.

```
nci.labs[1:4]
```

```
## [1] "CNS"    "CNS"    "CNS"    "RENAL"
```

```
table(nci.labs)
```

```
## nci.labs
##      BREAST         CNS       COLON K562A-repro K562B-repro    LEUKEMIA
##           7           5           7           1           1           6
## MCF7A-repro MCF7D-repro    MELANOMA       NSCLC     OVARIAN    PROSTATE
##           1           1           8           9           6           2
##       RENAL     UNKNOWN
##           9           1
```

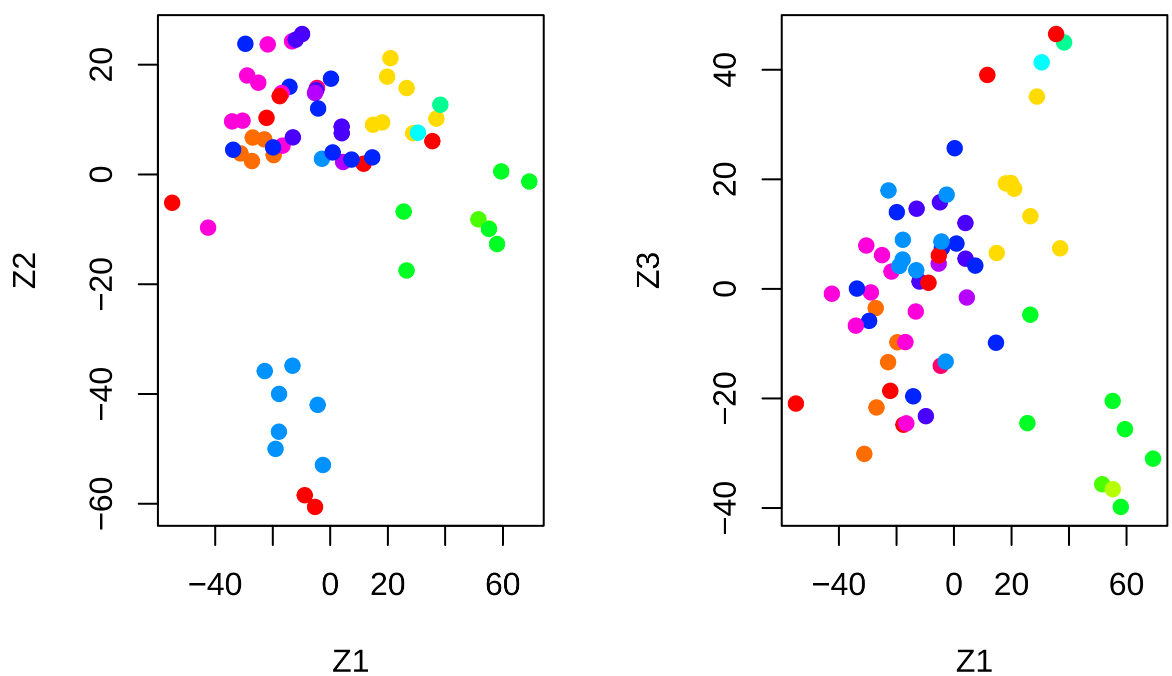
We first perform PCA on the data after scaling the variables (genes) to have standard deviation one, although one could reasonably argue that it is better not to scale the genes.

```
pr.out <- prcomp(nci.data, scale = TRUE)
```

The observations (cell lines) corresponding to a given cancer type will be plotted in the same colour, so that we can see to what extent the observations within a cancer type are similar to each other.

```
Cols <- function(vec) {
  cols <- rainbow(length(unique(vec)))
  return(cols[as.numeric(as.factor(vec))])
}
```

```
par(mfrow = c(1, 2))
plot(pr.out$x[, 1:2], col = Cols(nci.labs), pch = 19,
     xlab = "Z1", ylab = "Z2")
plot(pr.out$x[, c(1, 3)], col = Cols(nci.labs), pch = 19,
     xlab = "Z1", ylab = "Z3")
```



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

```
summary(pr.out)
```

```
## Importance of components:
```

```
##                            PC1      PC2      PC3      PC4      PC5      PC6
## Standard deviation     27.8535 21.48136 19.82046 17.03256 15.97181 15.72108
## Proportion of Variance  0.1136  0.06756  0.05752  0.04248  0.03735  0.03619
## Cumulative Proportion   0.1136  0.18115  0.23867  0.28115  0.31850  0.35468
##                             PC7      PC8      PC9     PC10     PC11     PC12
## Standard deviation     14.47145 13.54427 13.14400 12.73860 12.68672 12.15769
## Proportion of Variance  0.03066  0.02686  0.02529  0.02376  0.02357  0.02164
## Cumulative Proportion   0.38534  0.41220  0.43750  0.46126  0.48482  0.50646
##                            PC13     PC14     PC15     PC16     PC17     PC18
## Standard deviation     11.83019 11.62554 11.43779 11.00051 10.65666 10.48880
## Proportion of Variance  0.02049  0.01979  0.01915  0.01772  0.01663  0.01611
## Cumulative Proportion   0.52695  0.54674  0.56590  0.58361  0.60024  0.61635
##                            PC19    PC20     PC21    PC22    PC23    PC24
## Standard deviation     10.43518 10.3219 10.14608 10.0544 9.90265 9.64766
## Proportion of Variance  0.01594  0.0156  0.01507  0.0148 0.01436 0.01363
## Cumulative Proportion   0.63229  0.6479  0.66296  0.6778 0.69212 0.70575
##                           PC25    PC26    PC27   PC28    PC29    PC30    PC31
## Standard deviation     9.50764 9.33253 9.27320 9.0900 8.98117 8.75003 8.59962
## Proportion of Variance 0.01324 0.01275 0.01259 0.0121 0.01181 0.01121 0.01083
## Cumulative Proportion  0.71899 0.73174 0.74433 0.7564 0.76824 0.77945 0.79027
##                           PC32    PC33    PC34    PC35    PC36    PC37    PC38
## Standard deviation     8.44738 8.37305 8.21579 8.15731 7.97465 7.90446 7.82127
## Proportion of Variance 0.01045 0.01026 0.00988 0.00974 0.00931 0.00915 0.00896
## Cumulative Proportion  0.80072 0.81099 0.82087 0.83061 0.83992 0.84907 0.85803
##                           PC39    PC40    PC41   PC42    PC43   PC44    PC45
## Standard deviation     7.72156 7.58603 7.45619 7.3444 7.10449 7.0131 6.95839
## Proportion of Variance 0.00873 0.00843 0.00814 0.0079 0.00739 0.0072 0.00709
## Cumulative Proportion  0.86676 0.87518 0.88332 0.8912 0.89861 0.9058 0.91290
##                          PC46    PC47    PC48    PC49    PC50    PC51    PC52
## Standard deviation     6.8663 6.80744 6.64763 6.61607 6.40793 6.21984 6.20326
## Proportion of Variance 0.0069 0.00678 0.00647 0.00641 0.00601 0.00566 0.00563
## Cumulative Proportion  0.9198 0.92659 0.93306 0.93947 0.94548 0.95114 0.95678
##                           PC53    PC54    PC55    PC56    PC57   PC58    PC59
## Standard deviation     6.06706 5.91805 5.91233 5.73539 5.47261 5.2921 5.02117
## Proportion of Variance 0.00539 0.00513 0.00512 0.00482 0.00438 0.0041 0.00369
## Cumulative Proportion  0.96216 0.96729 0.97241 0.97723 0.98161 0.9857 0.98940
##                           PC60    PC61    PC62    PC63      PC64
## Standard deviation     4.68398 4.17567 4.08212 4.04124 2.148e-14
## Proportion of Variance 0.00321 0.00255 0.00244 0.00239 0.000e+00
## Cumulative Proportion  0.99262 0.99517 0.99761 1.00000 1.000e+00
```