Introduction to Computational BioStatistics with R: unsupervised learning

Erik Spence

SciNet HPC Consortium

10 November 2022

SciNet compute • calcul CANADA

Today's slides

Today's slides can be found here. Go to the "Introduction to Computational BioStatistics with R" page, under Lectures, "Unsupervised learning".

`https://scinet.courses/1246`

SciNet compute • calcul CANADA

Today's class

Today we're going to explore some unsupervised learning algorithms:

- Factor Analysis,
- Principle Component Analysis,
- Clustering algorithms.

These are algorithms that don't require the target (label, or $\boldsymbol{y}$ value). As a result, they are "unsupervised", they have nothing to guide them.

Ask questions!

SciNet compute • calcul CANADA

Curse of dimensionality

The "curse of dimensionality" is a generic term which refers to the difficulty in properly fitting or modelling data in high-dimensional spaces.

- Each feature in your data set is another dimension.
- Each dimension gives more space for your solution to live in.
- The more space there is, the harder it can be to find the solution, or build a meaningful model.
- "Dimensionality reduction", or "feature selection" is the act of either
  - ignoring data which is obviously not important, or
  - modifying the data to put it into a form which has fewer dimensions.

We will examine Factor Analysis and Principle Component Analysis (PCA), which are both dimensionality-reduction techniques.

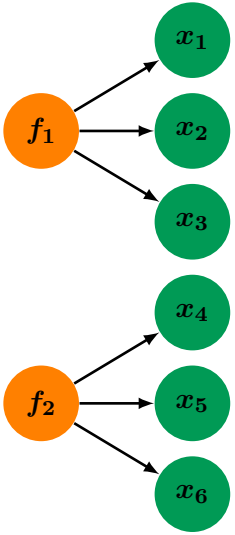
SciNet compute • calcul CANADA

Factor analysis

Imagine you've got a bunch of data, each with 6 features, x_1, x_2, ..., x_6. You've observed that there are correlations between some of the features.

It's possible that there might be some underlying, unobserved, 'factors', say f_1 and f_2, which are responsible for certain features, which is why some features are correlated to each other.

It would be useful to represent the data through these factors (also called "latent variables"), rather than the original features, since there are presumably fewer of them, and they are the actual cause of the feature's values.

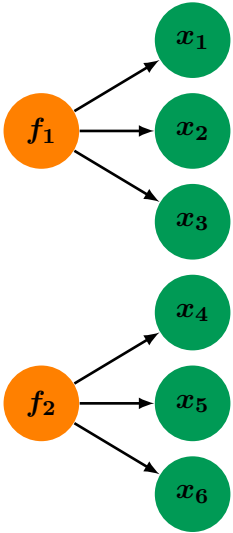


Factor analysis, continued

The goal of factor analysis (sometimes called "exploratory factor analysis") is to determine linear relationships between the factors and the features. Assuming there are only 2 factors, these relationships would take the form

$$x_1 = \beta_{10} + \beta_{11} f_1 + \beta_{12} f_2 + \epsilon_1$$
$$x_2 = \beta_{20} + \beta_{21} f_1 + \beta_{22} f_2 + \epsilon_2$$
$$\vdots$$
$$x_6 = \beta_{60} + \beta_{61} f_1 + \beta_{62} f_2 + \epsilon_6$$

Where the ϵ terms represent noise, and the β factors are known as "loadings".



Factor analysis, continued more

To calculate the relationships on the previous slide, we will make a few assumptions:

- The noise terms, ϵ, are independent, and have a mean of zero.
- The unobservable factors are independent of each other, have a mean of zero and a variance of 1 (the factors have been 'standardized').

The calculation of the loadings turns out to just be a whole lot of algebra, and correlations. We won't delve into the derivation details here.

We generally don't stop there. The calculation can be refined.

- The calculation of the factors is not unique (there are many combinations of loadings which will give the same answer).
- As such we can "rotate" the answer such that some of the loadings are large, and others are small. This makes the interpretation of the factors easier.

SciNet compute • calcul CANADA

Factor analysis, example

Let us do an example. The first thing that should be done is to test to make sure that the features have enough correlation. There are tests available to see if factor analysis is appropriate:

- Kaiser-Meyer-Olkin,
- Bartlett's test.

Generally a KMO test value greater than 0.6 is required for a factor analysis to proceed.

```
>
> survey.data <- read.csv('link below')
>
> x <- survey.data[,2:13]
>
> library(psych)
>
> KMO(x)
Kaiser-Meyer-Olkin factor adequacy
Call:  KMO(r = x)
Overall MSA = 0.83
MSA for each item =
 KM1  KM2  KM3  QC1  QC2  QC3  CT1  CT2  CT3  PC1  PC2  PC3
0.87 0.84 0.82 0.88 0.86 0.86 0.83 0.85 0.85 0.70 0.78 0.80
>
```

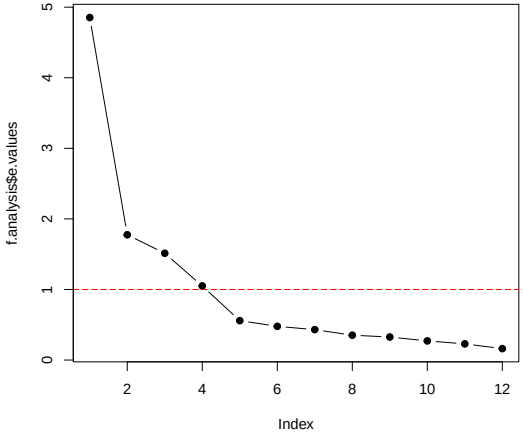
Data set: `https://raw.githubusercontent.com/housecricket/data/main/efa/sample1.csv`

Factor analysis, example, continued

We plot the eigenvalues which result from the analysis to determine how many factors we actually need to keep.

Anything less than 1 is not important, so we should keep the first four factors.

```
>
> f.analysis <- fa(x, nfactors = ncol(x))
>
> plot(f.analysis$e.values, type = 'b',
+       pch = 21, bg = 'black)
> abline(h = 1, col = 'red', lty = 5)
>
```

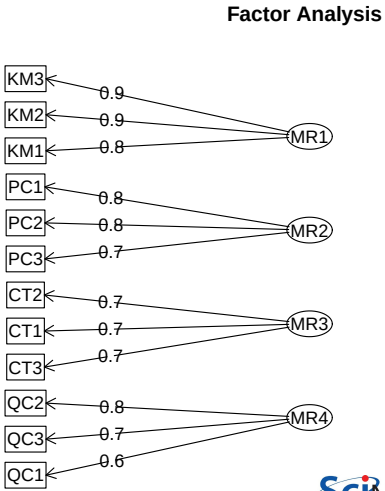


Factor analysis, example, continued more

We can plot loadings of the factors on the features.

We choose to not display any loadings which are less than 0.5.

```
>
> f.analysis2 <- fa(x, nfactors = 4)
>
> fa.diagram(f.analysis2, cut = 0.5)
>
```



Factor analysis, final notes

By representing the data using factors, we can now reduce the number of features. We then feed this representation of the data into our favourite machine learning algorithm.

Some things to be aware of:

- Generally, before performing factor analysis, you should perform an "adequacy test", which determines if the data can be factored:
  - Bartlett's test,
  - Kaiser-Meyer-Olkin test.
- There are different rotation algorithms available, which may affect the final results.

Note that factor analysis is sometimes controversial, since the final result is not unique, and thus the interpretation of the factors is subjective.

SciNet compute • calcul CANADA

Principle component analysis

Principle Component Analysis (PCA), seems similar to Factor Analysis on the surface, but it is different in important ways. Like Factor Analysis, it is a technique that allows dimensionality reduction in problems with purely continuous features.

It ignores the data targets; it merely imagines the data as points in a p-dimensional space.

- PCA performs a spatial transformation to the data space,
- the transformation rotates and scales the data space into directions defined by the variance in the data.
- Singular Value Decomposition is used to perform these steps.

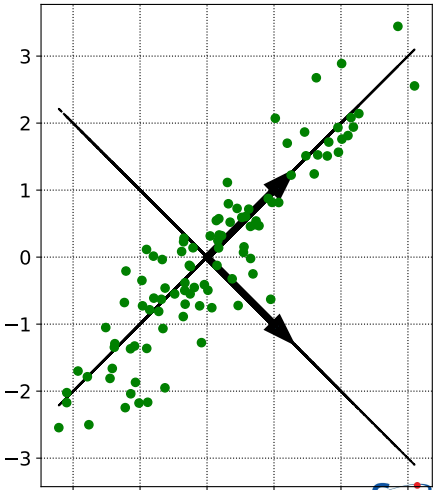
In essence (in linear-algebra-speak), PCA simply projects the data onto a new basis set, where the important features are clearer.

SciNet compute • calcul CANADA

PCA, continued

Principle component analysis picks out the directions the data takes that contain the most variance.

- The direction in which the variance is highest is rotated to point along the first axes (the first principal component). Next along the second axis, *etc.*
- Increasingly higher dimensions are flatter and flatter, as they have less variance.
- The dimensions which have very little variance contain very little information, and can be discarded.



PCA, example

We can perform PCA on the Parkinson's data set.

By assigning a column a NULL value the column will be removed from the data frame.

PCA is built into base R. No extra packages are needed.

Once built, the PCA object contains a tonne of information about the fit, including the principle components themselves.

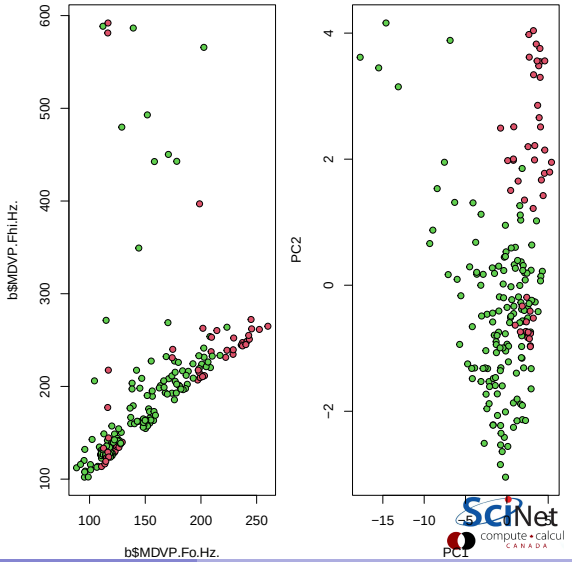
```
>
> parkinson <- read.csv('link below')
> status <- parkinson$status
>
> parkinson$name <- NULL
> parkinson$status <- NULL
>
> park.pca <- prcomp(parkinson,
+                    center = TRUE,
+                    scale = TRUE)
>
```

Data source: `https://archive.ics.uci.edu/ml/machine-learning-databases/parkinsons/parkinsons.data`

PCA, example, continued

```
>
> par(mfrow = c(1,2))
>
> plot(parkinson$MDVP.Fo.Hz.,
+      parkinson$MDVP.Fhi.Hz.,
+      bg = status + 2, pch = 21)
> plot(park.pca$x[,1], park.pca$x[,2],
+      bg = status + 2, pch = 21,
+      xlab = 'PC1', ylab = 'PC2')
>
```

The data is automatically transformed into the principal-component space.



PCA, continued more

Of what use is it?

- Once you have projected the data onto its principle components, you can determine which components are most important.
- Those dimensions which are least important can be discarded, resulting in a dimensionality reduction.
- Note that PCA in R will not automatically centre the data, this should be done explicitly.
- Once projected onto the new space, clustering is sometimes a useful next step. PCA can sometimes separate clusters that are otherwise difficult to detect.

But how do we decide which components to keep? What is a good criteria? If we don't throw away any dimensions we're no better off than we were before.

SciNet compute • calcul CANADA

Explained variance

The quality of the PCA, and which components are worth keeping, is indicated in the "Cumulative Proportion" of the variance that is explained by the component.

- 58.9% of the variance in the data set lies along the first principle component.
- 11.3% along the second, etc.

```
>
> summary(park.pca)
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6    PC7
Standard deviation     3.600 1.577 1.24178 1.21037 0.98687 0.85388 0.7431
Proportion of Variance 0.589 0.113 0.07009 0.06659 0.04427 0.03314 0.0251  ...
Cumulative Proportion  0.589 0.702 0.77209 0.83868 0.88295 0.91609 0.9412
>
```

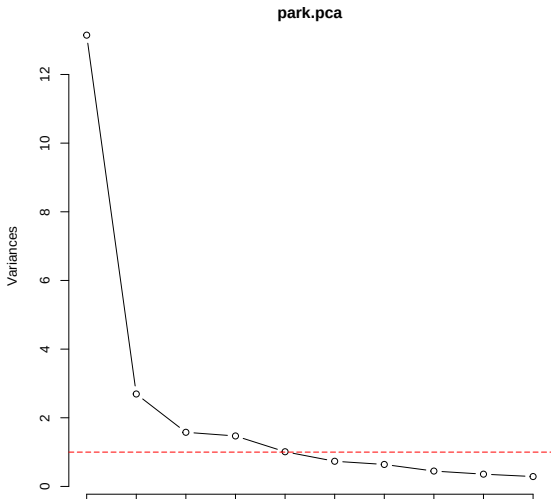
94% of the variance is explained by the first seven principle components.

Using eigenvalues

Alternatively, we can use the eigenvalues to determine which principal components are worth keeping.

An eigenvalue of less than 1 indicates that the component contains less information than the original data features.

```
> plot(park.pca, type = 'l')
> abline(h = 1, col = 'red', lty = 5)
>
```



PCA, summary

Some further notes about PCA:

- PCA doesn't drop features; rather, it generates combinations of all features in order of how significantly they vary.
- One generally keeps most of the information from all features, but expressed in a number of combinations $k < p$.
- However, especially in situations with a large number of dimensions, the least significant principal components can often be profitably ignored, as there is very little variation in those directions.
- Once projected onto the new space, clustering is sometimes a useful next step. PCA can sometimes separate clusters that are otherwise difficult to detect.

Note there are other types of dimensionality reduction as well: Locally Linear Embedding (LLE), Linear Discriminant Analysis (LDA), and others.

SciNet compute • calcul CANADA

Clustering

Let's switch to a different sort of classification approach: clustering.

- This is a type unsupervised learning.
- It's unsupervised because there are no targets (labels, $\boldsymbol{y}$ values) used.

This can show up in all sorts of applications:

- Finding patterns in properties of galaxies.
- Determine proteins with similar interaction types.
- Market segmentation.
- "Customers who buy X often buy...".

There are two main clustering approaches you'll run into: $\boldsymbol{k}$-means and hierarchical clustering.

SciNet compute • calcul CANADA

Clustering, continued

The reason for using algorithms to find clusters in the data is because

- It's difficult to find clusters in high-dimensional data (since you can't visualize it all).
- You might want to summarize a large number of observations into fewer, similar clusters.

Obviously, we haven't defined what we mean by "similar" or "cluster" yet.

- A "cluster" is a group of data points which are centred around some central, average point.
- The "similarity" between points is determined by some measure of "distance" between them, in the p dimensional space in which they live.
- In continuous spaces the distance can be Euclidean, or some other measure of distance (L1 norm).
- In ordinal spaces (bag-of-words counts, for example) you can use the "cosine similarity"

$$\cos\theta = \frac{\mathbf{A}\cdot\mathbf{B}}{|\mathbf{A}||\mathbf{B}|}$$

k-means clustering

k-means clustering is a geometric clustering algorithm which finds roughly-spherical blobs of clusters amongst the data. The algorithm is straightforward. Starting with k initial cluster centres:

- Assign each data point to the nearest centre.
- Recalculate the centre of each cluster, based on its members.
- Move the centres to the new locations.
- Repeat until converged (the centres stop moving).

The value of k must be specified before starting.

SciNet compute • calcul CANADA

k-means clustering, example

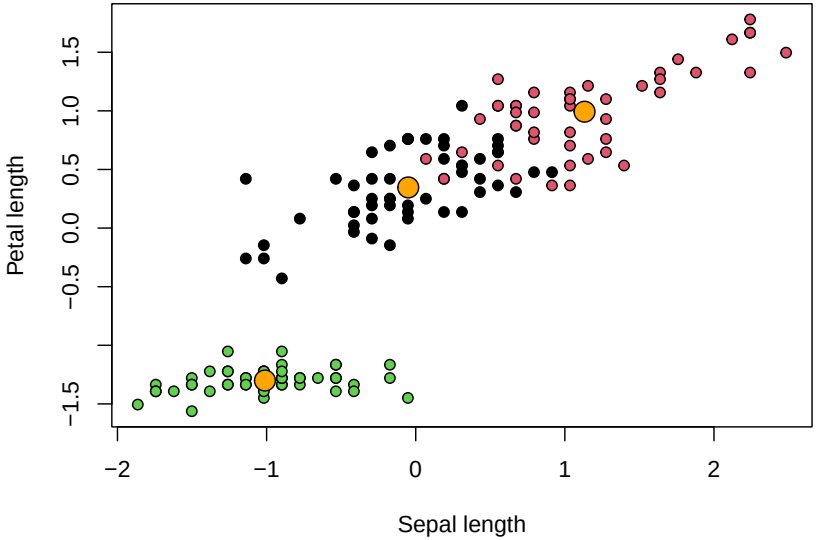
As you might expect, k-means is built into base R.

Because this is a geometric problem, we want to centre and scale the data. The 'scale' function will do this to each column.

Once the model is trained, you can get the centres of the clusters, and the predicted labels, using the "cluster" and "centres" model entries.

```
>
> x <- scale(iris[, -5])
>
> iris.kmeans <- kmeans(x, 3, nstart = 10)
>
> plot(x[,1], x[,3], bg = iris.kmeans$cluster,
+      pch = 21, xlab = 'Sepal length',
+      ylab = 'Petal length')
>
> points(iris.kmeans$centers[,1],
+        iris.kmeans$centers[,3], bg = 'orange',
+        cex = 2, pch = 21)
>
```

k-means clustering, example, continued



k-means clustering, continued

k-means has both strengths and weaknesses.

- You need to know what value of k to use.
- Random initialization of the centres can go badly wrong.
- For this to be robust, you need to repeat many times.
- We do this explicitly by specifying the nstart flag; the best result is returned.
- k-means has a tendency to make equally-populated clusters, which can lead to incorrect results.

For this to work consistently, we need a way to measure the quality of the model.

SciNet compute • calcul CANADA

k-means clustering, quality measures

A few measures of error have been developed for k-means.

- We'd like to minimize the within-cluster sum of squares, where μ_i is the centre of the ith cluster.

$$\mathbf{WCSS} = \sum_{i}^{k} \sum_{j \in S_i} |x_j - \mu_i|^2$$

- We'd like to maximize the between-cluster sum of squares.

$$\mathbf{ICSS} = \sum_{i}^{n} \sum_{j}^{n} \delta(S_i, S_j) \, |x_i - x_j|^2$$

These are output by standard k-means algorithms.

k-means and cross-validation

How do we pick k? You guessed it!

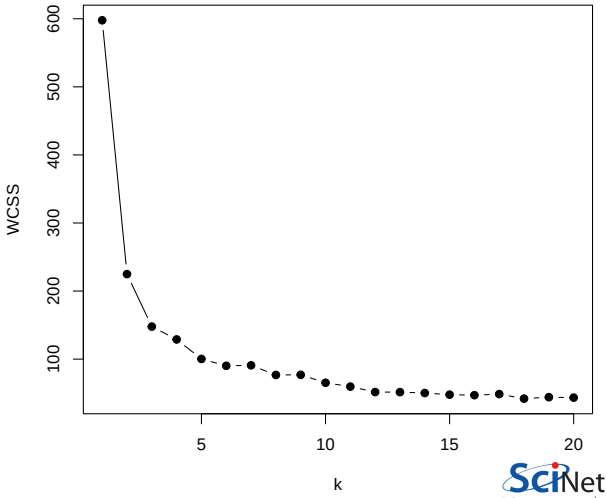
- For reasons aren't clear, cross-validation is not available for k-means.
- The createFolds function will return the indices of the folds.
- The cl_predict function will predict the categories (based on cluster) that each point belongs to.
- The total WCSS is returned.

```
> library(caret)   # needed for the createFolds function
> library(clue)   # needed for the cl_prediction function
> kmean.wcss <- function(k, i.data) {
+       folds <- createFolds(iris$Species, k = 10)
+       total <- 0
+       for (ind in 1:10) {
+           train.data <- i.data[-folds[[ind]],]
+           test.data <- i.data[folds[[ind]],]
+           model <- kmeans(train.data, k, nstart = 10)
+           pred <- cl_predict(model, test.data)
+           diff <- model$centers[pred,] - test.data
+           total <- total + sum(apply(diff, MARGIN = 1,
+                       function(x) return(sum(x**2))))
+       }
+       return(total)
+ }
>
```

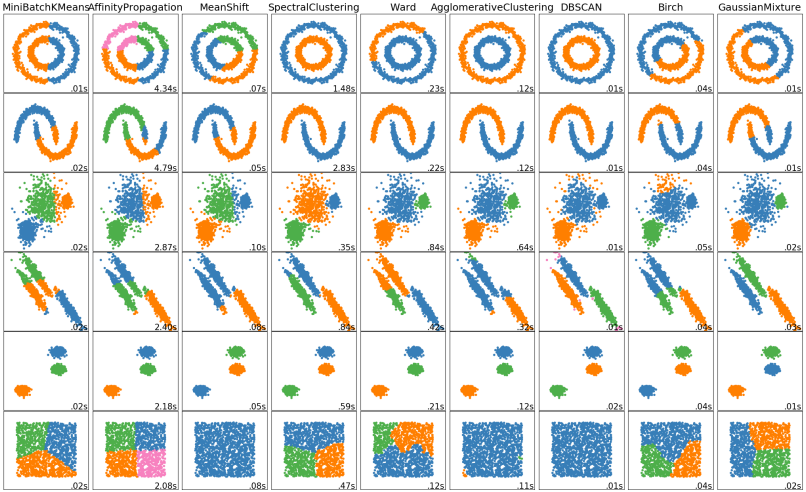
k-means and cross-validation, continued

```
>
> plot(sapply(1:20, kmean.wcss,
+              scale(iris[,-5])),
+      type = 'b', xlab = 'k',
+      ylab = 'WCSS')
>
```

Unlike other algorithms, like kNN or polynomial fits, the accuracy of k-means does not 'turn over', meaning start to get worse with increasing k.



Scikit-learn clustering algorithms



http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

Summary

Some things to remember:

- Factor Analysis, PCA and clustering require numeric data.
- Factor Analysis recasts the data as linear combinations of factors.
- PCA projects the data onto a new basis set, based on the variance in the data.
- PCA accounts for as much variance in the data as possible. Factor analysis accounts for as much covariance in the data as possible.
- By discarding the dimensions with minimal variance the dimensionality of the problem can be reduced.
- PCA results are also often used as the input to clustering algorithms, since the major sources of variance have already been isolated.
- Clustering algorithms group data into 'clusters' of common attributes.
- k-means find clusters by finding the centres of clusters of data. k-means requires k to be specified.

SciNet compute • calcul CANADA