Processing and Analysis of Biological Data

A primer of multivariate analysis

Øystein H. Opedal

29 Oct 2022

Introduction

Biological data are very often multivariate, in the sense that we are dealing with potentially large sets of correlated variables. Some times the patterns of covariation among variables is the focus of the investigation, and other times we may have taken several measurements that describe similar aspects of the biology in question. In any case, we very often have to deal with analyses of multiple variables. Note that when we refer to multivariate analyses, we mean analyses where there is more than one response variable. Thus, a multiple regression is normally considered a univariate analysis even though there are multiple correlated predictor variables.

Variance matrices and eigendecomposition

Multivariate data can be summarized as *variance matrices*, which are symmetrical matrices with variances on the diagonal and covariances on the off-diagonals. We will work with the following example variance matrix **CM**.

```
cm = matrix(c(0.7, 0.2, -0.3,
              0.2, 1.2, 0.4,
              -0.3, 0.4, 0.6),
              nrow=3)
cm
```

```
##      [,1] [,2] [,3]
## [1,]  0.7  0.2 -0.3
## [2,]  0.2  1.2  0.4
## [3,] -0.3  0.4  0.6
```

EXERCISE: One way to confirm that a matrix is symmetrical is to show that it is identical to it´s transpose ($\mathbf{A} = \mathbf{A}^T$). Confirm that **CM** is symmetrical.

```
cm == t(cm)
```

```
##      [,1] [,2] [,3]
## [1,] TRUE TRUE TRUE
## [2,] TRUE TRUE TRUE
## [3,] TRUE TRUE TRUE
```

EXERCISE: Translate the covariance matrix into a correlation matrix.

We can use a variance matrix to simulate data from the multivariate normal distribution $MVN(\bar{x}, \Sigma)$, where Σ is a variance matrix.

```
library(MASS)

X = data.frame(mvrnorm(200, mu=c(5,3,7), Sigma=cm))
colnames(X) = c("z1", "z2", "z3")
head(X)
```

```
##         z1       z2       z3
## 1 5.915670 2.019195 6.348190
## 2 5.112695 5.096272 7.093866
## 3 5.013278 1.776102 6.132074
## 4 3.225220 3.118558 8.734926
## 5 4.456616 3.132123 6.839903
## 6 4.613846 2.294005 6.952148
```

Variance matrices have several useful properties for multivariate analysis. A common operation is a so-called eigendecomposition (or spectral decomposition).

A vector v is an eigenvector of the matrix $\mathbf{A}$ if it satisfies the condition

$\mathbf{Av} = \lambda\mathbf{v}$,

where λ is an eigenvalue of $\mathbf{A}$. From this follows also the relation

$\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^{-1}$.

where $\mathbf{Q}$ is a matrix with the eigenvectors in columns, and Λ is a square matrix with the eigenvalues on the diagonal.

Biologically, the eigen analysis allows us to 'rotate' the variation in the data (say, in the multivariate phenotype of an organism) so that the first 'trait' (leading eigenvector) represents the multivariate direction of most variation. Often, this can be interpreted roughly as the size of the organism. The subsequent eigenvectors represent other axes of variation, that could e.g. represent shape. The subsequent eigenvectors are *orthogonal*, so that e.g. the second eigenvector is perpendicular to the first.

In `R`, the function `eigen` performs the eigendecomposition and returns the eigenvectors and corresponding eigenvalues.

```
eigen(cm)
```

```
## eigen() decomposition
## $values
## [1] 1.4034727 0.9349665 0.1615607
##
## $vectors
##            [,1]       [,2]       [,3]
## [1,] 0.07739007  0.8285983  0.5544688
## [2,] 0.90374270  0.1765485 -0.3899741
## [3,] 0.42102245 -0.5312773  0.7351766
```

The eigenvalues represent the amount of variance associated with each eigenvector (given in columns). We can thus compute the proportion of variance associated with each eigenvector as $\lambda_i/\sum\lambda$.

Before continuing, we need to recall the rules for matrix multiplication. There are several forms of matrix multiplication, but the 'normal' matrix multiplication requires that the number of columns in the first matrix equals the number of rows in the second matrix, and the resulting matrix will have the same number of rows as the first matrix, and the same number of columns as the second matrix. If we multiply a matrix of dimensions $m \times n$ with one of dimensions $n \times l$, we get a matrix of dimensions $m \times l$. The matrix multiplication operator in `R` is `%*%`.

EXERCISE: Compute the proportion of variance associated with each eigenvector of **CM**.

```
## [1] 0.56138909 0.37398661 0.06462429
```

EXERCISE: Confirm that the eigenvectors are of unit length (length = 1) and that the angle between them is 90 degrees.

Recall that the length of a vector is the square root of the sum of the vector elements, and the angle between two vectors u_1 and u_2 is $\frac{180}{\pi} cos^{-1}(u_1 u_2)$.

Length of the eigenvectors

```
## [1] 1 1 1
```

Angle between first and second eigenvector

```
##      [,1]
## [1,]   90
```

EXERCISE: Reconstruct the matrix **CM** from the eigenvalues and eigenvectors.

```
##      [,1] [,2] [,3]
## [1,]  0.7  0.2 -0.3
## [2,]  0.2  1.2  0.4
## [3,] -0.3  0.4  0.6
```

```
##      [,1] [,2] [,3]
## [1,]  0.7  0.2 -0.3
## [2,]  0.2  1.2  0.4
## [3,] -0.3  0.4  0.6
```

Principal Component Analysis

Eigenanalysis is a core component of principal component analysis. In it's simplest form, the principal components are the same as the eigenvectors. Let us derive some new traits along the eigenvectors of **CM**.

```
dim(as.matrix(X))
```

```
## [1] 200   3
```

```
dim(as.matrix(eigen(cm)$vectors[,1]))
```

```
## [1] 3 1
```

```
t1 = as.matrix(X) %*% eigen(cm)$vectors[,1]
t2 = as.matrix(X) %*% eigen(cm)$vectors[,2]
t3 = as.matrix(X) %*% eigen(cm)$vectors[,3]

c(var(X[,1]), var(X[,2]), var(X[,3]))
```

```
## [1] 0.5859019 1.1045851 0.5718627
```

```
c(var(t1), var(t2), var(t3))
```

```
## [1] 1.2925683 0.7949348 0.1748466
```

Notice that the variances of new traits decreases from the first to the third trait, which was not the case for the original traits. However, the total variance stays the same (because we have just reorganized the variation).

```
var(t1) + var(t2) + var(t3)
```

```
##         [,1]
## [1,] 2.26235
```

```
var(X[,1]) + var(X[,2]) + var(X[,3])
```

```
## [1] 2.26235
```

The eigenvectors are *orthogonal*, i.e. they are not correlated with each other.

A very similar operation is performed by several `R` packages for principal component analysis, e.g. `prcomp`. The principal components are not exactly the same as defining traits along the eigenvectors, but are subject to some further rotation. However, the principal components will be strongly correlated with the traits defined along the eigenvectors.

```
pca = princomp(X)
summary(pca)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3
## Standard deviation     1.1410874 0.8833489 0.41067272
## Proportion of Variance 0.5784356 0.3466425 0.07492192
## Cumulative Proportion  0.5784356 0.9250781 1.00000000
```

The proportion of variance explained by each principal component is computed as the variance of each principal component divided by the total, which is basically equal to the corresponding eigenvalue divided by the sum of the eigenvalues, i.e. $\lambda_i / \sum \lambda$.

```
pca$sdev^2/sum(pca$sdev^2)
```

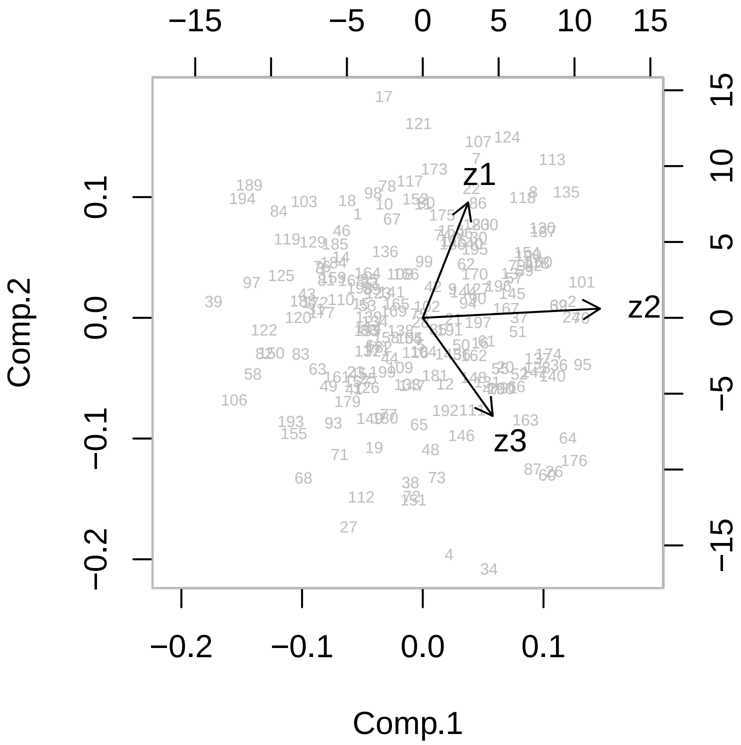
```
##     Comp.1     Comp.2     Comp.3
## 0.57843562 0.34664246 0.07492192
```

```
eigen(cm)$values/sum(eigen(cm)$values)
```

```
## [1] 0.56138909 0.37398661 0.06462429
```

The small difference is again due to how the principal components are calculated, but biologically the interpretation is the same. A PCA can be illustrated by a biplot.

```
biplot(pca, col=c("grey", "black"), cex=c(.5, 1))
```



Principal component regression

And the chong thing