Dimension Reduction

Exploratory Data Analysis with **R**

Teiresias, the blind seer seated in conversation with Odysseus. Late Classical (420-330 BCE)

Boris Steipe

UNIVERSITY OF TORONTO

Department of Biochemistry
Department of Molecular Genetics

This module includes some material originally developed by Raphael Gottardo, FHCRC and by Sohrab Shah, UBC

Objectives

Understand Principal Component Analysis (PCA) as a method for dimension reduction;

Be able to perform PCA on your data and interpret the results;

Be able to use the results to identify data of interest;

Know about alternatives such as projection and embedding methods.

Introduction to PCA

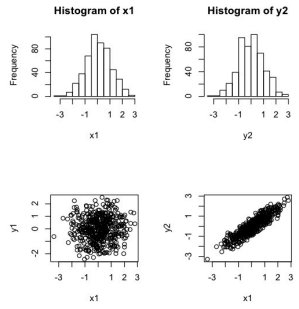
The goal of **P**rincipal **C**omponent **A**nalysis (PCA) is to *transform* a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components.

The possibly smaller number of variables can be used for data reduction and visualization.

PCs are orthogonal (i.e. uncorrelated).

What Does Correlation Really Mean?

```
set.seed(2707)
x1 <- rnorm(500,0,1)
y1 <- rnorm(500,0,1)

y2 <- 2*x1 + y1
y2 <- y2-mean(y2)
y2 <- y2 / sd(y2)
plot(x1, y1)
plot(x1, y1)
oPar <- par(mfrow = c(2,2))
hist(x1)
hist(y2)
plot(x1, y1)
plot(x1, y2)
par(oPar)
```



Correlations between real world data are ubiquitous, due to causal relationships or confounding factors.

PCA and Correlation

Principal component analysis (PCA) converts a set of observations of possibly correlated variables into a set of values of uncorrelated variables called *principal components*.

The first principal component is the projection of the data into a single dimension that has *as high a variance as possible* (that is, accounts for as much of the variability in the data as possible); each succeeding component in turn has the highest variance possible under the constraint that it be *orthogonal* to (uncorrelated with) the preceding components.

Therefore the PCs provide a view on the structure of the data that best explains its variance.

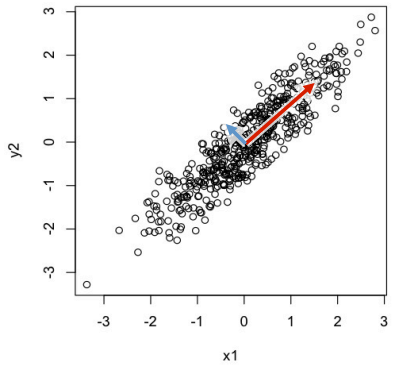
This is especially useful for EDA of high-dimensional data that can.t be intuitively visualized.

cf. Wikipedia: Principal component analysis

PCA and Correlation

The example data is two-dimensional, but most of the information is contained along a dimension shown here by the red vector.

We could thus restrict our analysis to a projection along that vector.

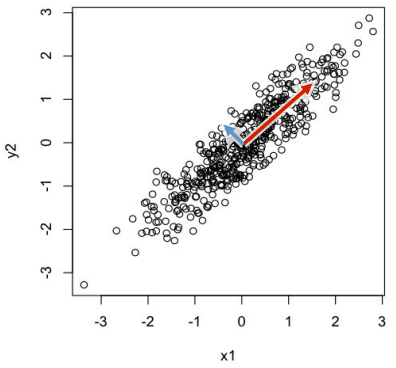


Interpretation of PCs

Given a set of points in Euclidean space, the first principal component corresponds to a line that passes through the multidimensional mean and minimizes the sum of squares of the distances of the points from the line.

The second principal component is calculated in the same way, after all correlation with the first principal component has been subtracted out from the points.

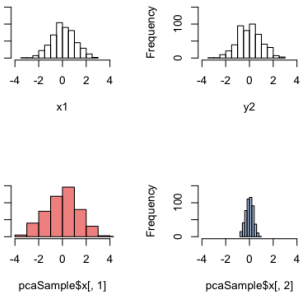
[...]



Interpretation of PCs

After applying the PCA, and plotting the original and the projected coordinates ...

```
# calculate a PCA
pcaSample <- prcomp(cbind(x1,y2))

Opar <- par(no.readonly=TRUE)
par(mfrow = c(2,2))
hist(x1,
   xlim=c(-4,4), ylim=c(0,150), main="")
hist(y2,
   xlim=c(-4,4), ylim=c(0,150), main="")
hist(pcaSample$x[,1],
   xlim=c(-4,4), ylim=c(0,150),
   main="", col=rgb(0.86,0,0,0.5))
hist(pcaSample$x[,2],
   xlim=c(-4,4), ylim=c(0,150),
   main="", col=rgb(0.31, 0.5, 0.74, 0.5))
par(Opar)
```



... we indeed find that most of the variance is now contained in the first histogram. In a sense, the single dimension of the first principal component contains most of the information of our originally two-dimensional data.

Two Alternatives ...

R has two different functions for PCA: `prcomp()` and `princomp()`.

They use different mathematical approaches but the results are virtually identical. `prcomp()` is numerically more stable.

However, they also use different names for the elements of their result lists.

`prcomp()`	`princomp()`	
`center`	`center`	The vector that was subtracted to center the data
`sdev`	`sdev`	Standard deviations for each dimension of the rotated data
`rotation`	`loadings`	The actual principal components
`x`	`scores`	The rotated data, i.e. after projection along each PC

e.g. use `data$x` for the rotated results of a `prcomp()` call, but use `data$scores` if the result came from `princomp()`

Scaling

PCA is sensitive to the scaling of the variables.

If we have just two variables and they have the same sample variance and are positively correlated, then the PCA will entail a rotation by 45 and the "loadings" for the two variables with respect to the principal component will be equal. But if we multiply all values of the first variable by 100, then the principal component will be almost the same as that variable, with a small contribution from the other variable, whereas the second component will be almost aligned with the second original variable.

This means that whenever the different variables have different units (like temperature and mass), PCA is a somewhat arbitrary method of analysis. (Different results would be obtained if one used Fahrenheit rather than Celsius for example.) One way to address this is to scale variables to have unit variance.

EDA with PCA

Dimension reduction (simplify a dataset)

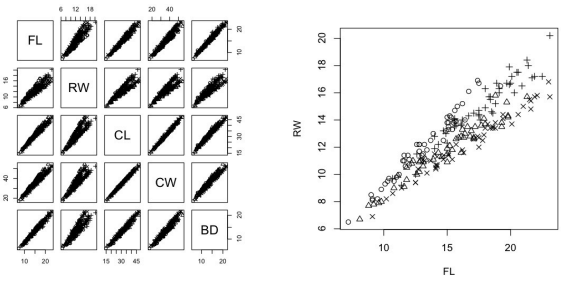
Analysis of the relative importance of dimensions

Crabs Data

The crabs data frame has 200 rows and 8 columns, describing 5 morphological measurements on 50 crabs each of two colour forms and both sexes, of the species *Leptograpsus variegatus* collected at Fremantle, W. Australia.

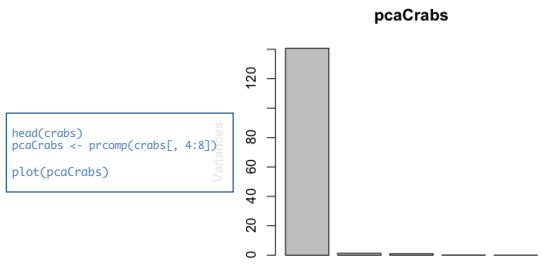
```
library(MASS)
data(crabs)
head(crabs)
# Two species: blue and orange,
# Two genders: female and male
# FL frontal lobe size (mm)
# RW rear width (mm)
# CL carapace length (mm)
# CW carapace width (mm)
# BD body depth (mm)

fac <- as.factor(paste(crabs[, 1], crabs[, 2],sep="."))
head(fac)
c(fac[1], fac[51], fac[101], fac[151])
as.numeric(c(fac[1], fac[51], fac[101], fac[151]))

plot(crabs[, 4:8], pch=as.numeric(fac))
plot(crabs[, 4:5], pch=as.numeric(fac))
```

Crabs Data



There are four distinct groups of crabs in the data, but it seems difficult to tell them appart based on the data.

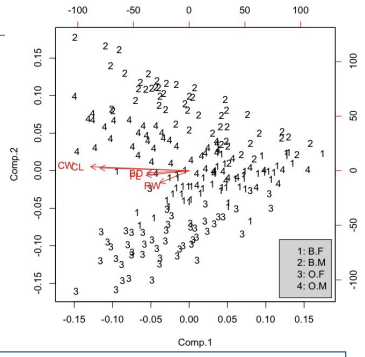
Crabs Data




```
head(crabs)
pcaCrabs <- prcomp(crabs[, 4:8])

plot(pcaCrabs)
```

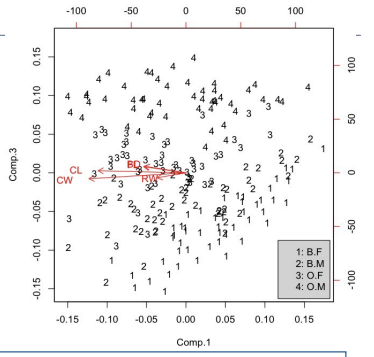
```
> summary(pcaCrabs)
Importance of components:
                          PC1     PC2     PC3     PC4     PC5
Standard deviation     11.862 1.13879 1.00013 0.36783 0.27913
Proportion of Variance  0.982 0.00906 0.00698 0.00094 0.00054
Cumulative Proportion   0.982 0.99153 0.99851 0.99946 1.00000
```


Crabs Data




```
biplot(pcaCrabs, xlabs=as.numeric(fac))
legend(81, -63,c("1: B.F", "2: B.M", "3: O.F", "4: O.M"), box.col=1, bg="lightgrey")
```


Crabs Data



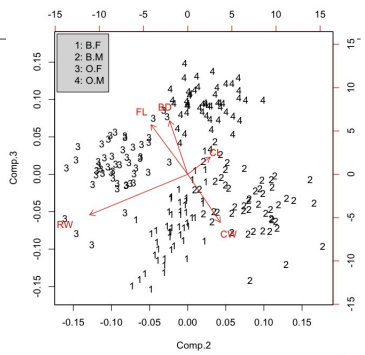

```
biplot(pcaCrabs, xlabs=as.numeric(fac), choices = c(1, 3))
legend(84, -63, c("1: B.F", "2: B.M", "3: O.F", "4: O.M"), box.col=1, bg="lightgrey")
```


Crabs Data

Plotting the second against the third principal component gives a very different picture.

Interpret this plot ...

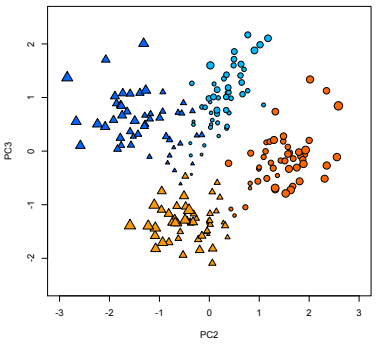
Do the measurements distinguish between the four different types of crabs?




```
biplot(pcaCrabs, xlabs=as.numeric(fac), choices = c(2, 3))
legend(-14.8, 16.2, c("1: B.F", "2: B.M", "3: O.F", "4: O.M"), box.col=1, bg="lightgrey")
```


Crabs Data ... Improved Visualization

We can see the dependence on the confounding factor "age" or "overall size" most clearly if we plot this with different colors and shapes ... and scale the plotting symbols by size.



Conclusion

PCA is a powerful tool for EDA.

Can be very useful to detect confounding variables.

Application to gene expression data may identify characteristic patterns.

- PCA of genes (eigen-genes)
- PCA of samples (eigen-assays)

Downside: You lose the interpretation of variables.

Model based methods can be applied on the fly.

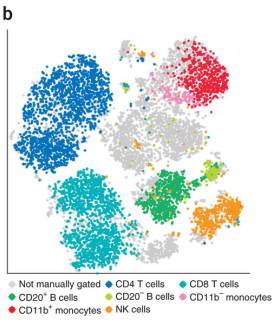
Alternatives

New algorithms for dimension reduction of high-dimensional data may be even more useful for EDA:

"viSNE allows visualization of high-dimensional single-cell data on a two-dimensional map. [...] Whereas viSNE plots resemble conventional biaxial plots, their utility comes from combining and representing information from all dimensions simultaneously. [...] We illustrated how viSNE can be used to characterize heterogeneity within cancer samples, mark disease progression from diagnosis to relapse, and identify rare cancer populations lurking among predominantly healthy cells."

El-ad David Amir *et al.* (2013) viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia

Nature Biotechnology doi:10.1038/nbt.2594
Published online 19 May 2013



t-Stochastic Neighbour Embedding (tSNE)

Developed in Geoff Hinton's lab at UofT.

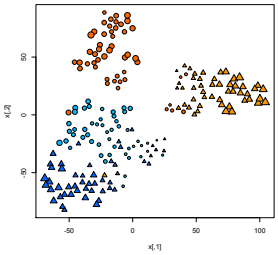
Matches joint-probabilities (~distances) of data points in high-dimensional data and low-dimensional embedding spaces.

R-code available:

```
install.packages("tsne")
```

Exercise: Explore tsne

Crabs data



Van der Maaten & Hinton (2008) Visualizing Data using t-SNE. Journal of Machine Learning Research **9**:*2579-2605*

boris.steipe@utoronto.ca