PCA II: a summary of the data with maximal variance

Data Science 101 Team
PCA II: finding a linear combination with maximal variance

- We are going to arrive at the idea behind principal components from a different perspective
- We are going to describe an explicit rule with which to solve this problem
- Note: understanding the rule requires some knowledge in linear algebra that not all of you might have. This is OK: you will not be required to understand the derivation for the purpose of this class, but

1. If you do, you will enjoy the application of linear algebra
2. If you do not, you will be motivated to learn more linear algebra
The test/re-test problem

You and your friends want to compare the speeds of your pets

You want to have them run between two points and record arrival times, but it is somewhat difficult to convince cats, bunnies, hamsters, chickens, dogs etc to comply with the rules of track and field

So, keeping the total distance constant, you design a set of different close courses (in the grass, dirt, black-top...), with different incentivisation devices (a loud noise at the start; you running ahead of your pet and prompting it to follow; the pet’s food clearly visible at the end of the course; you pursuing the pet with a torch; you saying “ready set go,” ...) and you record the pet times in each of them

When you look at the results, you see that the course/incentive combinations lead to variable spread of times for the animals and variable average times. How would you combine the results to get a good ranking of the speed of the pets?
Aggregating the results of questions

You want to evaluate how much each of your friends likes to travel and decide to design a questionnaire.

You want to use 20 questions, each of the type:

- How much do you agree with the following statement? […] Indicate your answer with a number between 0 (“Strongly disagree”) and 10 (“Strongly agree”).

In deciding on the statements, you try to pick 20 such that “Strong agreement” with the statement should indicate strong affinity to travel.

After, you convinced 50 of your friends to take the survey, realize that some of your questions are not very discriminatory. For example, everybody expressed agreement between 8 and 10 with the statement “It is great to discover new things.”
A meaningful “average”

- When we have many measurements of “the same underlying quantity”, it is often useful to combine them in one “average” value
- Before we rush to take the arithmetic mean, however, there are few things worth considering.
- Each measurement might have an inherently different mean (animals run faster from a fire than from a laud noise), so that it is useful to subtract the mean from each measurement before combing them
- Measurements might have different variances
- When we summarize all the measurements in one “average” we are projecting multidimensional data onto one dimension: it is a good idea to choose a dimension that separates things well.
Projections
Good Projections
Good Projections – going back to last time

Blue points are one dimensional summary of black points

slope = -0.5
Sum of SE = 21.58

slope = -1
Sum of SE = 53.85

slope = -2
Sum of SE = 58.04

slope = -4
Sum of SE = 42.16

slope = 4
Sum of SE = 5.88

slope = 2
Sum of SE = 6.03

slope = 1
Sum of SE = 14.53

slope = 0.5
Sum of SE = 10.67
Looking at two variables and setting up the problem

- Let $X_1$ and $X_2$ be our initial variables, centered so that their mean is 0.
- We want to choose $w_1$ and $w_2$ so that $t_i = w_1 \cdot X_{1i} + w_2 \cdot X_{2i}$ has maximal variance $n^{-1} \sum_i t_i^2$
- We need to add a constraint to make the problem meaningful (otherwise the bigger $w_1$ or $w_2$ the bigger the variance)
- We choose the constraint $w_1^2 + w_2^2 = 1$

Note that then we are not quite finding an “average” of the two variables ($w_1 + w_2 \neq 1$);

- Pythagoras’ theorem tells us that the solution to this problem is the projection onto the line that minimizes the sum of squared euclidean distances from the points.
- We are going to solve the maximal variance problem
Rephrasing the problem with matrix notation

Let

\[ w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad X = \begin{bmatrix} X_{11} & X_{21} \\ X_{12} & X_{22} \\ X_{13} & X_{23} \\ \vdots & \vdots \\ X_{1n} & X_{2n} \end{bmatrix} \]

Then

\[ t = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix} = Xw \]

We can write the problem as follows

\[
\max_{w:w'w=1} \frac{1}{n} \sum_i t_i^2 = \max_{w:w'w=1} w'(n^{-1}X'X)w.
\]
Maximal variance: an eigenvalue-eigenvector problem

Define the Lagrangian \( \mathcal{L} = w'(n^{-1}X'X)w - \lambda(w'w - 1) \).
Differentiate with respect to \( w \) (squint for a second and pretend it is a one dimensional object).

\[
\frac{\partial \mathcal{L}}{\partial w} = 2(n^{-1}X'X)w - 2\lambda w = 0 \implies (n^{-1}X'X)w = \lambda w
\]

\( \implies \) \( w \) is an eigenvector of \( n^{-1}X'X \) and \( \lambda \) an eigenvalue.

Note that

\[
n^{-1} \sum_{i} t_i^2 = w'(n^{-1}X'X)w = w'\lambda w = \lambda w'w = \lambda.
\]

\( \implies \) To get maximal variance we need to pick the eigenvector associated to the maximal eigenvalue. This is the major axis of Galton's ellipse and the **principal component**.

The second eigenvector is the minor axis of Galton's ellipse!
More than 2 variables

- Note that there is nothing special about 2 variables in the matrix notation framework.

Let

\[
  \begin{bmatrix}
    w_1 \\
    w_2 \\
    \vdots \\
    w_p 
  \end{bmatrix}
\]

\[
  X = \begin{bmatrix}
    X_{11} & X_{21} & \cdots & X_{p1} \\
    X_{12} & X_{22} & \cdots & X_{p2} \\
    \vdots & \vdots & \ddots & \vdots \\
    X_{1n} & X_{2n} & \cdots & X_{pn} 
  \end{bmatrix}
\]

\[
t = Xw
\]

We still have

\[
\max_{w: w'w = 1} \sum_i t_i^2 = \max_{w: w'w = 1} w'X'Xw.
\]

- The solution is still given by eigenvalues and eigenvectors
Calculating Principal Components in R: the decathlon dataset

Performances of 33 men’s decathlon at the 1988 Olympic Games, as reported in the R package ade4.
The decathlon dataset: scatter plot of performances
The decathlon dataset: who does well
Calculating principal components in R

```r
> deca.pc = prcomp(deca[,1:10],center=TRUE,scale=TRUE)
> summary(deca.pc)

Importance of components:

<table>
<thead>
<tr>
<th></th>
<th>PC1</th>
<th>PC2</th>
<th>PC3</th>
<th>PC4</th>
<th>PC5</th>
<th>PC6</th>
<th>PC7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard deviation</td>
<td>1.8488</td>
<td>1.6144</td>
<td>0.97123</td>
<td>0.9370</td>
<td>0.74607</td>
<td>0.70088</td>
<td>0.65620</td>
</tr>
<tr>
<td>Proportion of Variance</td>
<td>0.3418</td>
<td>0.2606</td>
<td>0.09433</td>
<td>0.0878</td>
<td>0.05566</td>
<td>0.04912</td>
<td>0.04306</td>
</tr>
<tr>
<td>Cumulative Proportion</td>
<td>0.3418</td>
<td>0.6025</td>
<td>0.69679</td>
<td>0.7846</td>
<td>0.84026</td>
<td>0.88938</td>
<td>0.93244</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>PC8</th>
<th>PC9</th>
<th>PC10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard deviation</td>
<td>0.55389</td>
<td>0.51667</td>
<td>0.31915</td>
</tr>
<tr>
<td>Proportion of Variance</td>
<td>0.03068</td>
<td>0.02669</td>
<td>0.01019</td>
</tr>
<tr>
<td>Cumulative Proportion</td>
<td>0.96312</td>
<td>0.98981</td>
<td>1.00000</td>
</tr>
</tbody>
</table>
```

- Note that by default the program looks at all eigenvalues and associated eigenvectors – we will learn more about this next time.
Exploring the principal component object: $\lambda$

Here are the **eigenvalues** – rather their square root.

```
deca.pc$sdev
```

```r
## [1] 1.8488478 1.6144328 0.9712345 0.9370279
## [5] 0.7460742 0.7008762 0.6561975 0.5538936
## [9] 0.5166715 0.3191460
```

\[
\sum_i \lambda_i = \text{trace}(X'X) = \sum_i \text{Var}(X_i)
\]

They inform us about the portion of variance explained.
Exploring the principal component object: $w$

**eigenvectors**: coefficients for the linear combination

```r
> deca.pc$rotation

<table>
<thead>
<tr>
<th></th>
<th>PC1</th>
<th>PC2</th>
<th>PC3</th>
<th>PC4</th>
<th>PC5</th>
<th>PC6</th>
<th>PC7</th>
<th>PC8</th>
<th>PC9</th>
<th>PC10</th>
</tr>
</thead>
<tbody>
<tr>
<td>m100</td>
<td>-0.415823</td>
<td>0.1488081</td>
<td>-0.26747198</td>
<td>0.08833244</td>
<td>-0.442314456</td>
<td>0.03071237</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LongJump</td>
<td>0.3940515</td>
<td>-0.1520815</td>
<td>-0.16894945</td>
<td>0.24424963</td>
<td>0.368913901</td>
<td>-0.09378242</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShotPut</td>
<td>0.2691057</td>
<td>0.4835374</td>
<td>0.09853273</td>
<td>0.10776276</td>
<td>0.009754680</td>
<td>0.23002054</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HighJump</td>
<td>0.2122818</td>
<td>0.0278985</td>
<td>-0.85498656</td>
<td>-0.38794393</td>
<td>-0.001876311</td>
<td>0.7454380</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m400</td>
<td>-0.3558474</td>
<td>0.3521598</td>
<td>-0.18949642</td>
<td>-0.08057457</td>
<td>0.146965351</td>
<td>-0.32692886</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m110Hurdles</td>
<td>-0.4334816</td>
<td>0.0695682</td>
<td>-0.12616012</td>
<td>0.38229029</td>
<td>-0.088802794</td>
<td>0.21049130</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discus</td>
<td>0.1757923</td>
<td>0.5033347</td>
<td>0.04609969</td>
<td>-0.02558404</td>
<td>0.019358607</td>
<td>0.61491241</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PoleVault</td>
<td>0.3840821</td>
<td>0.1495820</td>
<td>0.13687235</td>
<td>-0.14396548</td>
<td>-0.716743747</td>
<td>-0.34776037</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Javelin</td>
<td>0.1799436</td>
<td>0.3719570</td>
<td>-0.19232803</td>
<td>0.60046566</td>
<td>0.095582043</td>
<td>-0.43744387</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m1500</td>
<td>-0.1701426</td>
<td>0.4209653</td>
<td>0.22255233</td>
<td>-0.48564231</td>
<td>0.339772188</td>
<td>-0.30032419</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m100</td>
<td>0.2543985</td>
<td>-0.66371286</td>
<td>0.10839531</td>
<td>-0.10948045</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LongJump</td>
<td>0.7505343</td>
<td>-0.14126414</td>
<td>-0.04613910</td>
<td>-0.05580431</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShotPut</td>
<td>-0.1106637</td>
<td>-0.072505560</td>
<td>-0.42247611</td>
<td>-0.65073655</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HighJump</td>
<td>-0.1351242</td>
<td>0.155435871</td>
<td>0.10206505</td>
<td>-0.11941181</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m400</td>
<td>0.1413388</td>
<td>0.146839303</td>
<td>-0.65076229</td>
<td>0.33681395</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m110Hurdles</td>
<td>0.2725296</td>
<td>0.639003579</td>
<td>0.20723854</td>
<td>-0.25971800</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discus</td>
<td>0.1439726</td>
<td>-0.009400445</td>
<td>0.16724055</td>
<td>0.53450315</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PoleVault</td>
<td>0.2732665</td>
<td>0.276873049</td>
<td>0.01766443</td>
<td>0.06589572</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Javelin</td>
<td>-0.3419099</td>
<td>-0.058519366</td>
<td>0.30619617</td>
<td>0.13093187</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m1500</td>
<td>0.1868704</td>
<td>-0.007310045</td>
<td>0.45688227</td>
<td>-0.24311846</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Exploring the principal component object: $X_W$

The linear combinations of the original variables defined by the eigenvector are called Principal Components

```r
> dim(deca.pc$x)
[1] 33 10
> head(deca.pc$x)
          PC1         PC2         PC3        PC4        PC5         PC6
[1,] 1.732961  1.2313696 -2.79089389 -0.29003667  0.15622630  1.74397282
[2,] 2.787241 -0.1006798  0.83841190  0.13433649  0.06087808  0.09530507
[3,] 1.879222 -0.1368925  0.04459194  0.97006947 -0.91131154 -0.22215243
[4,] 2.313145  0.7945733  0.34689208 -0.25951896  1.34021373 -0.32968229
[5,] 2.260710 -2.0191111  0.48105873  0.07332298 -0.96067748 -0.10001486
[6,] 2.675217 -1.4325423 -0.52242032 -1.64275774  0.88508390  0.30958894
          PC7         PC8         PC9        PC10
[1,] 0.26554894  0.09359083  0.2533807 -0.17971152
[2,] 0.03842105 -0.01900623  0.3275702 -0.10364744
[3,] 0.43765023 -0.23283058  0.2806787  0.28922751
[4,] -0.29060776  1.10765204  0.1018245  0.07577141
[5,] 0.25106516 -0.27468209  0.2630097  0.53448629
[6,] 0.89152974 -0.03891612 -0.1928160  0.22205169
```
Coefficients of the events in the first principal component

par(las=3)
barplot(deca.pc$rotation[,1],main=paste(" PCA", as.character(1)),xlab="",ylab="Coefficients", names.arg=names(deca)[1:10])
The first principal component vs the Final Olympic Score

```
plot(deca.pc[,1], deca$Score, main="", xlab="First Principal Component", ylab="Olympic score")
```
The first principal component of the decathlon dataset

- We grouped events correctly (those for which a small number is better and those for which a large number is better)
- It provided us with a method to summarize the results that is quite close to the official scoring system – and we used no information on typical performance or difficulty of the event, just the results for these 33 subjects
- Summarizing variables by looking at the linear combination with maximal variants seems indeed to be doing something useful!