

Principal Component Analysis

Suzie Liu, Chase Slocum, Summer Wang

August 4, 2016 - Part 2

Overview

In this module, we focused on expanding principal component analysis (PCA) beyond a single principle component and applied it in datasets with more than two dimensions.

ISL 10.2 has a good breakdown of PCA.

Second Principle Component

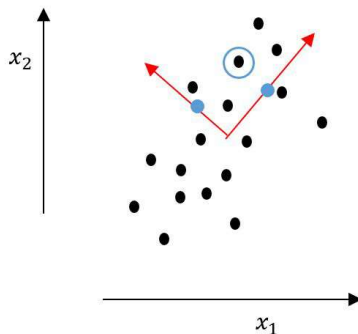
In a 2 dimensional dataset, a second principle component will always be perpendicular to the first. Once we add more dimensions, it will complicate this assumption.

In algebra terms:

original point (from first module on PCA): $x_i = (x_{i1}, x_{i2})$.

- With 1 principle component, we have an approximation of x_i .
 - $x_i \approx \alpha_i V$ where V is the first principle component and α is the score.
- With 2 principle compents, we have an exact equality.
 - V_1 = first principle component
 - V_2 = second principle component
 - $x_i = \alpha_{i1} V_1 + \alpha_{i2} V_2$
 - alphas are scores for x_i on those principle components

We can think of principle components as redefining the coordinate system.



Beyond 2 dimensions

As we expand beyond 2 dimensions, x_i is no longer defined by two coordinates. Instead, it is defined by some number, D , of coordinates.

$$x_i \in \mathbb{R}^D$$
$$x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$$

Because we are working with more than 2 dimensions, two principle components will no longer explain all of the variation in our dataset.

Our first 2 dimensional vectors will look like this:

V_1 = first principle component with scores α_{i1} ($\alpha_{i1} = x_i \cdot v_1$)

V_2 = second principle component with scores α_{i2} ($\alpha_{i2} = x_i \cdot v_2$)

With these two vectors: $x_i = \alpha_{i1}V_1 + \alpha_{i2}V_2$.

As we include more vectors, we will explain more of the variation in x_i . When we reach D vectors, all of the variation will be explained, but this would be stupid in practice as the purpose of PCA is to compress the data. Having D vectors, would keep the same amount of data, but it would be more difficult to interpret. The goal is to explain a lot of the variation with significantly fewer variables.

NOTE: PCA in this context is meant to help us get a better understanding/summary/visualization/structure of the data.

Example 1 in R

For this example, we used [pca_intro.R](#).

ggplot will help us create better plots:

```
library(ggplot2)
```

We will be using the iris dataset again.

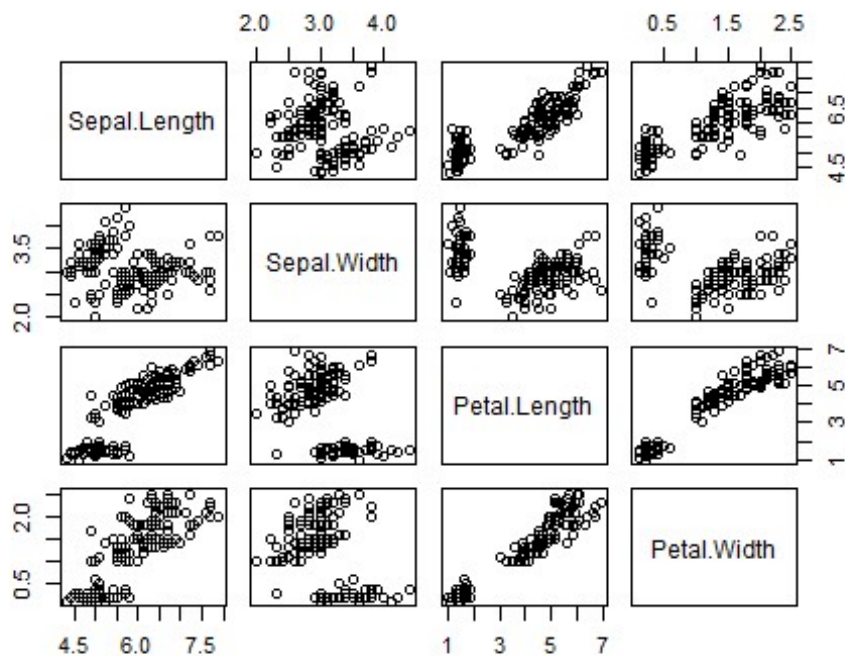
```
data(iris)
```

We want to use all of the flower structure predictors to build our principle components.

```
# Pick out the numerical columns from the data set (run ?iris for further info)
Z = iris[,1:4]
```

There is clearly a lot of correlation between these variables:

```
pairs(Z)
```



Now, we are going to run our principle component analysis.

```
# Run PCA
```

```
pc1 = prcomp(Z, scale.=TRUE) #different from K-means, scaling may not be necessary for PCA
```

Let's look at the principle components that were created.

```
# Look at the basic plotting and summary methods
```

```
pc1
```

```
## Standard deviations:
```

```
## [1] 1.7083611 0.9560494 0.3830886 0.1439265
```

```
##
```

```
## Rotation:
```

```
##           PC1      PC2      PC3      PC4
## Sepal.Length 0.5210659 -0.37741762 0.7195664 0.2612863
## Sepal.Width -0.2693474 -0.92329566 -0.2443818 -0.1235096
## Petal.Length 0.5804131 -0.02449161 -0.1421264 -0.8014492
## Petal.Width 0.5648565 -0.06694199 -0.6342727 0.5235971
```

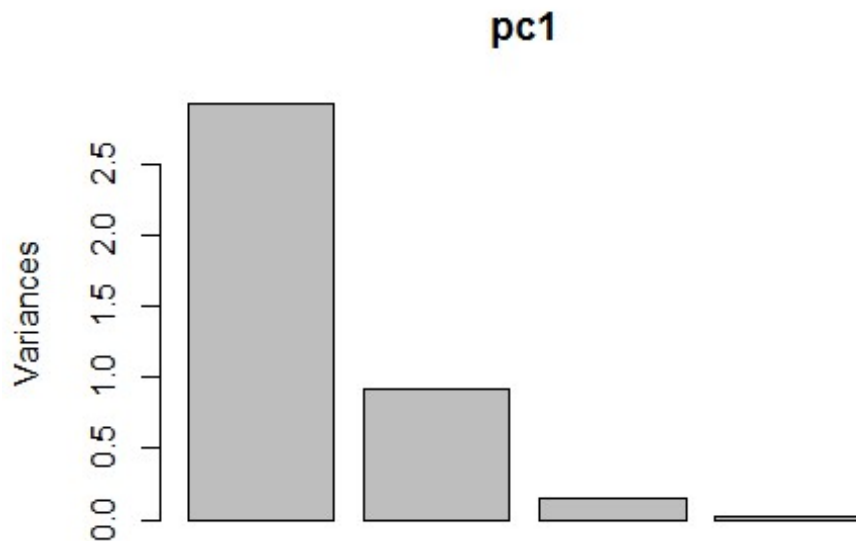
```
summary(pc1)
```

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

```
##           PC1      PC2      PC3      PC4
## Standard deviation 1.7084 0.9560 0.38309 0.14393
## Proportion of Variance 0.7296 0.2285 0.03669 0.00518
## Cumulative Proportion 0.7296 0.9581 0.99482 1.00000
```

Which principle components are more powerful? A scree plot will summarize the variance of the original dataset captured by each component:

```
plot(pc1)
```

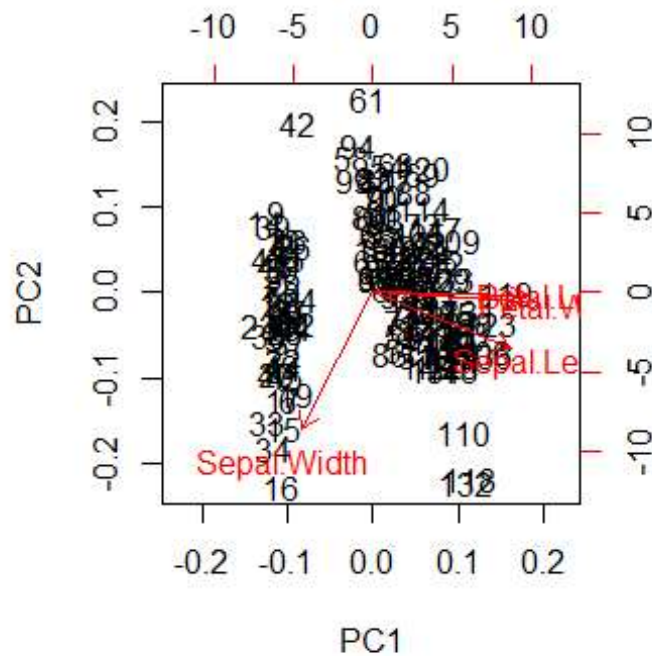


If we used all four of the components, we will capture all of the variation, but we can capture a significant portion of the variation if we just use 1 or 2 of the principle components.

**Note:* The amount of variance captured by the principle components is entirely dependent on the nature of the variation in the original dataset. Highly correlated variables will allow more variation to be captured by the individual principles components.

A biplot will help us see the data points plotted within the new coordinate system created by the principle components.

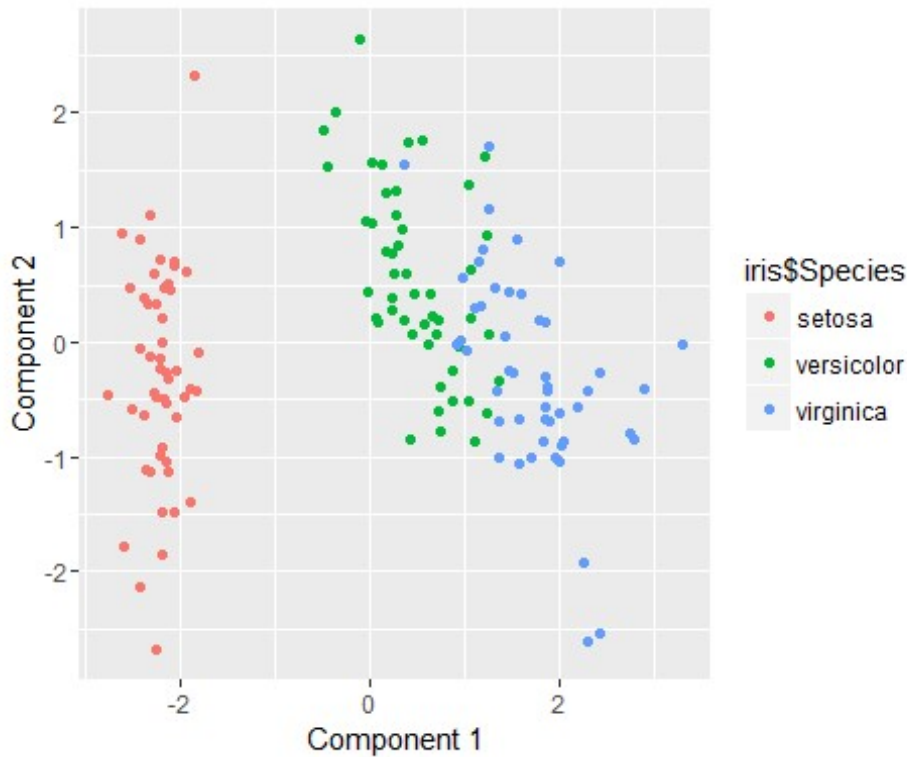
```
biplot(pc1)
```



#the points in the new coordinate space

Using ggplot, we can get a much more informative plot:

```
# A more informative biplot
loadings = pc1$rotation
scores = pc1$x
qplot(scores[,1], scores[,2], color=iris$Species, xlab='Component 1',
ylab='Component 2')
```



Notice that the axes represent the alphas given to each datapoint along that principle component.

A note about clusters: In this plot, we can see that PC1 separates the datapoints by species quite well. This is a side effect of our PCA and is related to the nature of the dataset. What it tells us is that the points' relative distances along these dimensions are highly related to their species. PCA is NOT a clustering method.

If we look at the loadings here, we can see what type of flowers will score well (get higher alphas) in the individual principle components.

#this is equal to pc1\$rotation (see last code chunk)
loadings

	PC1	PC2	PC3	PC4
## Sepal.Length	0.5210659	-0.37741762	0.7195664	0.2612863
## Sepal.Width	-0.2693474	-0.92329566	-0.2443818	-0.1235096
## Petal.Length	0.5804131	-0.02449161	-0.1421264	-0.8014492
## Petal.Width	0.5648565	-0.06694199	-0.6342727	0.5235971

Example: On PC1, the flowers with the higher alphas have a higher sepal length, petal length, and pedal width than average. However, they have a lower sepal width than average.

Example 2 in R

A more complex dataset

The iris dataset is useful for understanding the essential concepts of PCA, but the value of the process is not evident on a dataset with only 4 predictors. To demonstrate this, we will need to look at a more complex dataset.

For this example, we will use a congressional data.

- The r script can be found [here](#)
- The datasets are
 - [countdata](#)
 - [memberdata](#)

```
countdata = read.csv("congress109.csv", header=TRUE, row.names=1)
memberdata = read.csv("congress109members.csv", header=TRUE, row.names=1)
```

Countdata tells us how many times certain members of congress evoke certain phrases in floor speeches. There are 1000 columns in this dataset; we need compression, which is why PCA will be useful here. It will help us find the underlying dimensions of the data.

Memberdata includes basic info about the member:

```
head(memberdata)
```

##	party	state	chamber	repshare	cs1	cs2
## Chris Cannon	R	UT	H	0.7900621	0.534	-0.124
## Michael Conaway	R	TX	H	0.7836028	0.484	0.051
## Spencer Bachus	R	AL	H	0.7812933	0.369	-0.013
## Mac Thornberry	R	TX	H	0.7776520	0.493	0.002
## Randy Neugebauer	R	TX	H	0.7746633	0.506	0.070
## Nathan Deal	R	GA	H	0.7707892	0.474	0.058

Before we get started, we will want to normalize the frequencies.

```
# Normalize phrase counts to phrase frequencies
Z = countdata/rowSums(countdata)
```

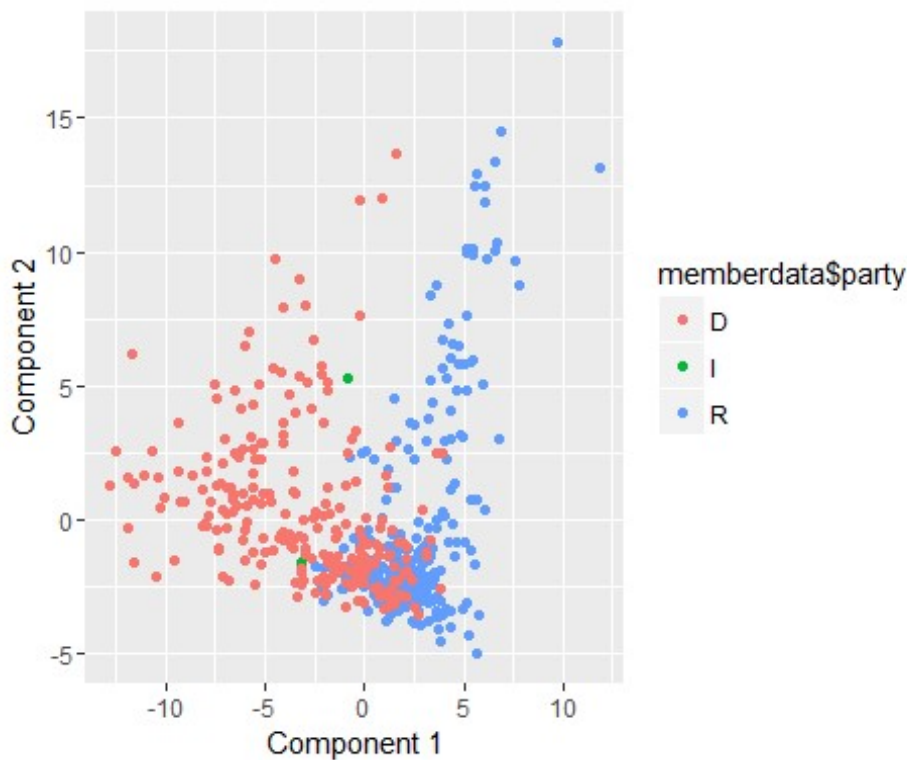
Then, we can run PCA. We will scale the data this time.

```
# PCA
pc2 = prcomp(Z, scale=TRUE)
#store the principles components
loadings = pc2$rotation
#store the alphas for each PC for each member
scores = pc2$x
```

If we look at a biplot of the data that is color coded according to the members party from the memberdata dataset, we can see that PC1 is a vector that divides the data by party:

```
#qplot is a part of ggplot2
```

```
qplot(scores[,1], scores[,2], color=memberdata$party, xlab='Component 1',  
ylab='Component 2')
```



Members that are close to each other in this plot are evoking certain phrases in a relatively similar amount, meaning members who are concerned about similar issues will be close to each other on these principle components. This explains why the points are being divided up along party lines by PC1. *We have gone from 1000 features to 2 principle components and are already able to divide the data up reasonably well by party.*

We can look at the top words that affect a member's score (alpha) on PC1:

```
# The top words associated with each component
```

```
o1 = order(loadings[,1])
```

```
#what words contribute to negative on PC1
```

```
colnames(Z)[head(o1,25)]
```

```
## [1] "private.account" "cut.medicaid"  
## [3] "student.loan" "tax.cut.wealthy"  
## [5] "budget.cut" "child.support"  
## [7] "privatizing.social.security" "privatize.social.security"  
## [9] "billion.tax.cut" "cut.food.stamp"  
## [11] "privatization.plan" "president.plan"  
## [13] "cut.student.loan" "plan.privatize"  
## [15] "tax.break" "care.cut"  
## [17] "medicaid.cut" "cut.benefit"
```



```
## [19] "cut.social.security"          "plan.privatize.social"
## [21] "social.security.privatization" "president.social.security"
## [23] "education.cut"                "live.poverty"
## [25] "pay.tax.cut"
```

These look like Republican phrases.

```
#what words contribute to positive on PC1
colnames(Z)[tail(o1,25)]
```

```
## [1] "law.review"          "urban.affair"
## [3] "housing.urban.affair" "judge.john.robert"
## [5] "chief.justice.william" "committe.commerce.science"
## [7] "date.time"           "justice.william.rehnquist"
## [9] "chief.justice"       "supreme.court.united"
## [11] "majority.vote"       "american.bar.association"
## [13] "partial.birth.abortion" "partial.birth"
## [15] "birth.abortion"      "john.robert"
## [17] "committe.foreign.relation" "business.meeting"
## [19] "judicial.nomine"     "roe.wade"
## [21] "justice.supreme.court" "court.judge"
## [23] "circuit.court.appeal" "circuit.court"
## [25] "court.appeal"
```

And these look like democrat phrases. That matches our plot.

Note: The abundance of judicial phrases is related to the confirmation of Justice Roberts to the Supreme Court. He was a conservative judge, explaining the democrat's discussion of him.

Let's look at what affects a member's score on PC2:

```
#order the words
o2 = order(loadings[,2])
#what words contribute to negative on PC2
colnames(Z)[head(o2,25)]
```

```
## [1] "strong.support"      "embryonic.stem.cel" "embryonic.stem"
## [4] "urge.support"        "stem.cel"          "cel.research"
## [7] "death.tax"           "business.owner"    "adult.stem"
## [10] "adult.stem.cel"      "illegal.immigration" "pass.bil"
## [13] "repeal.death.tax"    "private.property"   "illegal.immigrant"
## [16] "border.security"     "stem.cel.line"      "cel.line"
## [19] "estate.tax"          "post.office"        "percent.growth"
## [22] "family.business"     "driver.license"     "sex.offender"
## [25] "oil.food"
```

```
#what words contribute to negative on PC2
colnames(Z)[tail(o2,25)]
```

```
## [1] "minority.right"      "judge.robert"
## [3] "final.minute"        "circuit.court.judge"
## [5] "justice.supreme.court" "john.robert"
```

```
## [7] "committe.commerce.science" "circuit.court"
## [9] "circuit.court.appeal"      "justice.priscilla.owen"
## [11] "majority.vote"             "change.senate.rule"
## [13] "supreme.court.united"      "banking.housing.urban"
## [15] "committe.foreign.relation" "nuclear.option"
## [17] "housing.urban.affair"      "american.bar.association"
## [19] "urban.affair"              "president.judicial.nomine"
## [21] "court.judge"               "circuit.judge"
## [23] "president.pro.tempore"     "court.appeal"
## [25] "judicial.nomine"
```

It is much less clear what PC2 is measuring. Subject area knowledge might help us to understand this.

A final note on PCA: The more you compress a feature (i.e. the fewer principle components you use) the more detail/variation from the original data will be lost. Think about a picture. If you compress it a lot, you lose clarity and detail from the image.