Principal Components Analysis: A relatively painless exploration (hopefully)

Introduction

I'm not a statistics major, but I took Data 8 last semester because I wanted to learn about data science. Unfortunately, I found the material covered in that class was a little too shallow for me to find much utility. So far this semester, PCA really stood out to me because it could be easily applied to many different dimensions; however, I didn't really understand the explanation given in lecture about the intuition behind how it works. The purpose of this post is to explore and get some deeper insight without becoming too technical. This post will NOT focus on the calculation side, as R takes care of that for us.

Big Picture

Reduce Dimensionality

In lecture and on his slides, Professor Sanchez mentioned that the first part of the goal of PCA is to reduce dimensionality, which he defines as "condensing information in variables". The motivation to do this is to reduce the number of variables that we have to focus on.

Going back to the ranking nba teams example from hw03, we know a lot of different properties of each team. Not all properties make a big difference in how well a team is performing, but it's not obvious which properties are and which ones aren't. Even if we knew, for example, that the number of fouls was the least important characteristic of a team, if we remove it from the dataset we will lose whatever small bits of information it gives us completely. In addition, some properties may have some hidden overlap when they describe how well a team plays.

PCA is about making new variables (these are the PC1, PC2, etc. we calculated in hw03) by taking linear combinations of the pre-existing in a specific way (I have deliberately left out the details here in the interest of time and because R takes care of this for us). Each of these new variables contain some information from all the old variables, so when we remove one of the new variables, the remaining ones will still carry information about the old variables. Therefore, we can reduce number of variables we are looking at without losing a part of the data we collected completely.

```
library(readr)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#reading in the nba data table we created in hw03
nba <- read_csv(file="nba2017-teams.csv")
```

```
## Parsed with column specification:
## cols(
##   team = col_character(),
##   experience = col_integer(),
##   salary = col_double(),
##   points3 = col_integer(),
##   points2 = col_integer(),
##   free_throws = col_integer(),
##   points = col_integer(),
##   off_rebounds = col_integer(),
##   def_rebounds = col_integer(),
##   assists = col_integer(),
##   steals = col_integer(),
##   blocks = col_integer(),
##   turnovers = col_integer(),
##   fouls = col_integer(),
##   efficiency = col_double()
## )
```

```
pca_ver <- nba %>% select(points3, points2, free_throws, off_rebounds, def_rebounds, assists, steals, blocks, turn
overs, fouls)

#performing the PCA
pca_values <- prcomp(x=pca_ver, scale. = TRUE)

#displaying the associated weights used to calculate PC1 and PC2
pca_values$rotation[, 1:1]
```

```
##      points3      points2  free_throws off_rebounds def_rebounds
##    0.1121782    0.3601766    0.3227564    0.3029366    0.3719432
##      assists       steals       blocks    turnovers        fouls
##    0.3125312    0.3447256    0.3162237    0.3353958    0.3072548
```

- We are creating a variable, which we call PC1, for each team in the nba. The way to calculate a team's value of PC1 is to multiple the team's value of the old variable with the listed weight (for example the number of three pointers that the team scored multiplied by 0.1121782). Continue doing this for all variables and sum them together in order to the that team's PC1.
    - This is the "PCs as linear combinations" portion of lecture.

Retain variation

The second goal he mentioned was to keep the variation of the data as much as we can when we reduce the number of variables. This means that we want to construct these new variables in a way that make the teams look as different from each other as possible. If we created a new variable that makes the teams look the same, then it would be difficult to figure out which team should be ranked the highest.

```
eigs <- data.frame(
  eigenvalue = round(pca_values$sdev ^ 2, 4),
  prop = round(pca_values$sdev ^ 2 / sum(pca_values$sdev ^ 2), 4),
  cumprop = round(cumsum(pca_values$sdev ^ 2 / sum(pca_values$sdev ^ 2)), 4)
)
eigs
```

```
##    eigenvalue   prop cumprop
## 1      4.6959 0.4696  0.4696
## 2      1.7020 0.1702  0.6398
## 3      0.9795 0.0980  0.7377
## 4      0.7717 0.0772  0.8149
## 5      0.5341 0.0534  0.8683
## 6      0.4780 0.0478  0.9161
## 7      0.3822 0.0382  0.9543
## 8      0.2603 0.0260  0.9804
## 9      0.1336 0.0134  0.9937
## 10     0.0627 0.0063  1.0000
```

- Here we can see that the eigenvalues of the different PCs are actually their variances. Therefore, PC1 (which has the highest eigenvalue of 4.6959) is the most important because it has the highest variance and describes the biggest proportion of the data. When we reduce the dimensionality, we want to take out the PCs that have the smaller eigenvalues so we lose less variation.

Other uses

Generally, PCA looks at attributes of different objects in order to find relationships between them. For example, we used the PCA to calculate a composite index to rank NBA teams in hw03. However, there are many other ways that statisticians use the PCA.

Selecting variables

One obvious way is to help a data scientist decide which variables to focus on. Outliers that have a big effect on a model that the statistician is building can be revealed during a PCA.

Classification

Another way PCA can be used is to classify different objects when only given attributes (these are the variables). We can teach an algorithm to sort objects into pre-existing classifications, or we can ask it to create a classification for us.

Example of an algorithm sorting objects into pre-existing classifications

I attended a presentation about machine learning where a group of students demonstrated how they trained their algorithm to classify different orchids by species when given only the measurements of the inner and outer tepals, anther, stigma, etc. They used the attributes directly, but we can use PCs instead and follow the same procedure they did afterwards.

- Unfortunately, I am unable to get access to a visualization of their data, but below is a diagram showing some of the parts of an orchid that they used.

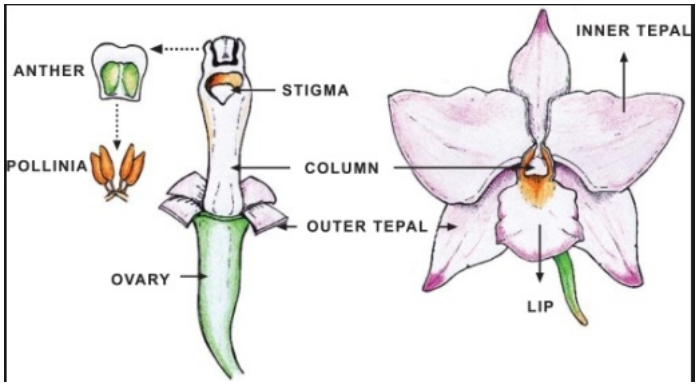
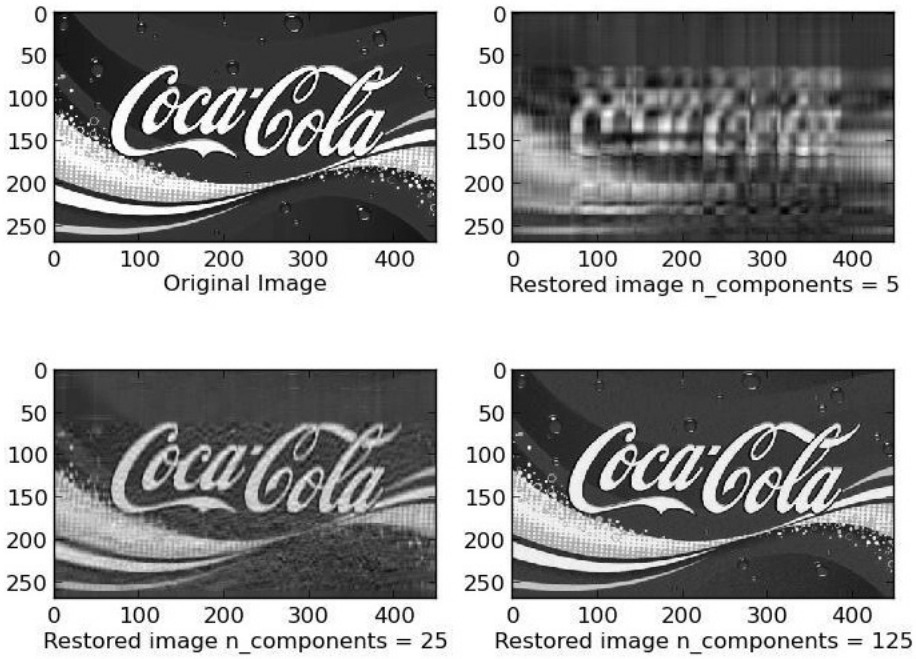


Diagram of an orchid

Image Compression

PCA is all about representing an object with less variables. This is useful for representing images when you want to save memory. One way to do this is to represent each pixel as a row, and each column tells you information about that specific pixel's intensity. By using less columns (less variables), we can still show the image. - Sidenote: This form of compression is considered lossy because you are losing a proportion of information by only choosing some of the PC values to create your pixel.



An example of image compression using PCA, albeit with a slightly more complex scheme

- This is also a good visual representation of the effect of the number PCs you choose to use on how much information is retained. From the picture, we can already see that using only 25 variables already gives us a very recognizable image.

Conclusion

In summation, PCA is about creating new attributes of objects using the characteristics we know. The new variables may be more difficult to understand, but the benefit of doing this is that we are able describe the objects using less variables which simplifies our analysis and lets us do cool things.

Sources

- Esbensen, Kim, and Paul Geladi. *Principal Component Analysis*. http://files.isec.pt/DOCUMENTOS/SERVICOS/BIBLIO/Documentos%20de%20acesso%20remoto/Principal%20components%20analysis.pdf
- Diagram of orchid. https://openi.nlm.nih.gov/imgs/512/221/3145264/PMC3145264_CG-12-342_F1.png
- Brems, Matt. *A One-Stop Shop for Principal Component Analysis*. https://medium.com/towards-data-science/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c
- A stackexchange post that gradually increases the technicality of the answer: https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues
- Sanchez, Gaston. *Slide about Introduction to PCA* https://github.com/ucb-stat133/stat133-fall-2017/blob/master/slides/15-principal-components1.pdf
- Here is a similar presentation to the one that I saw: https://plot.ly/~oikobill/33/vp-presentation/
- Smith, Lindsay. *A tutorial on Principal Components Analysis*. http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf
- Image compression example: https://pakallis.files.wordpress.com/2013/06/pca.jpeg?w=950