Learning about Principal Components Analysis and its Importance in Statistics

```
# install libraries
library(ggplot2)
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
```

Introduction

So far, in Stat 133, we have learned a lot about many very basic tools to work with data. Tools like how to find summary statistics using built in R functionality, utilize new packages like dplyr and ggplot2 for manipulating and graphing data, as well as build our own functions that make use of concepts like iteration to do something more specific. All of this is extremely important to understand, but in reality is not really what I consider to be the most interesting part of data science or statistics in general. These are great building blocks that have given us a fairly comprehensive toolbelt to attack problems and questions in data, but what I want to explore in this post is an actual application using some of these tools we have developed. Obviously one of the hottest subjects right now in data science is machine learning. The technique in machine learning I want to delve into is called Principal Components Analysis, or PCA for short. My motivation in writing this post about PCA is I have learned in my own work experience how useful it can be in the real world, and I want to share this skill with someone who may not have it.

PCA is regarded as one of the more important techniques in machine learning because of its widespread application. First let me establish what PCA is, and then we can move on to an example using a public dataset, and finally we'll conclude with understanding why this is so important.

```
# Reference: https://gab41.lab41.org/the-10-algorithms-machine-learning-engineers-need-to-know-f4bb63f5b2fa
```

Instructions

All of my work is easily reproducible as I am only using packages dplyr and ggplot2, as well as the public data set "USArrests" which is already in R by default. Any new functions I use, I will define as new as I go.

Part 1: What is PCA?

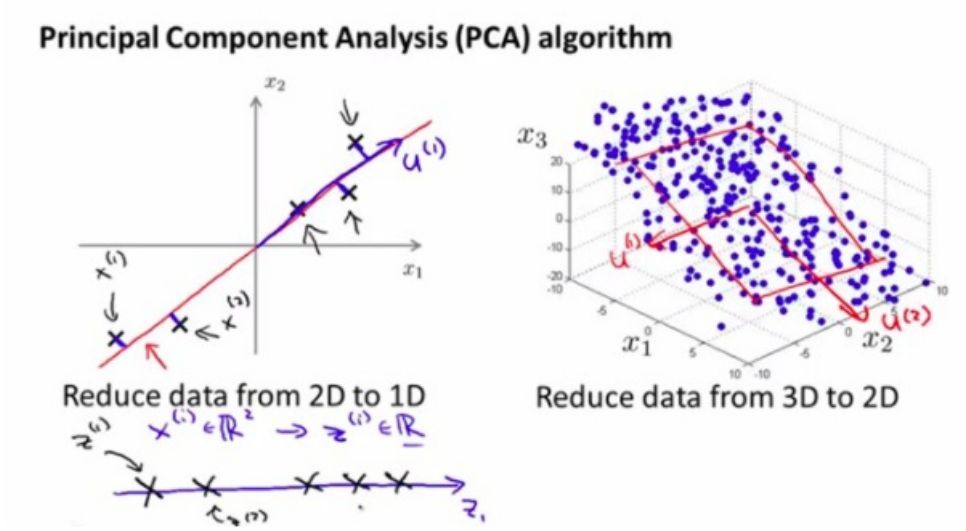
The main idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent. The same is done by transforming the variables to a new set of variables, which are known as the principal components (or simply, the PCs) and are orthogonal, ordered such that the retention of variation present in the original variables decreases as we move down in the order. So, in this way, the 1st principal component retains maximum variation that was present in the original components. The principal components are the eigenvectors of a covariance matrix, and hence they are orthogonal.

Importantly, the dataset on which PCA is to be used must be scaled. The results are also sensitive to the relative scaling. As a layman, it is a method of summarizing data. Imagine some wine bottles on a dining table. Each wine is described by its attributes like colour, strength, age, etc. But redundancy will arise because many of them will measure related properties. So what PCA will do in this case is summarize each wine in the stock with less characteristics.

```
# Reference: https://www.dezyre.com/data-science-in-python-tutorial/principal-component-analysis-tutorial
```

Here is a good image to better understand the key part of PCA - dimensionality reduction.

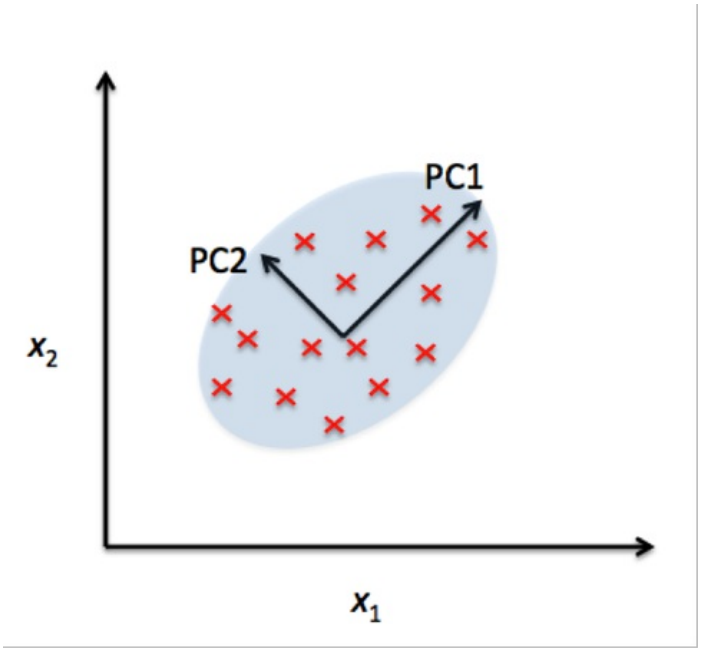
Dimensionality Reduction



```
# Reference: https://www.coursera.org/specializations/deep-learning?utm_source=gg&utm_medium=sem&campaignid=904733485&adgroupid=45435009512&device=c&keyword=deep%20learning%20coursera&matchtype=b&network=g&devicemodel=&adpostion=1t1&creativeid=231631799243&hide_mobile_promo&gclid=CjwKCAiAr_TQBRB5EiwAC_QCq5TLWjNZjqQGolDPjyzZ-3RCZ6pJY2JKiGBPBOCM2k-oSSZxE8WmABoCSxUQAvD_BwE
```

This image is another graphical representation showing a simple example of PCA. The data and our two components are graphed. Note how the components are orthogonal.

Sample PCA




```
# Reference: https://sebastianraschka.com/images/faq/lda-vs-pca/pca.png
```

Example of PCA

Now, we are going to see how to analyze a sample multivariate data set using PCA. The data set doe not have a ton of variables to demonstrate dimension reduction in all its glory, but it should still be good enough to understand how PCA works.

Before jumping into PCA logic in R, I am going to define a common function which is required to display all PCA related plots in a 2X2 grid. (Notice how we are pulling the “defining functions” tool from our Stat 133 toolbox to do this.)

```
# Defining a basic function to display PCA related plots in 2x2 grid
pcaCharts <- function(x) {
    x.var <- x$sdev ^ 2
    x.pvar <- x.var/sum(x.var)
    print("proportions of variance:")
    print(x.pvar)

    par(mfrow=c(2,2))
    plot(x.pvar,xlab="Principal component", ylab="Proportion of variance explained", ylim=c(0,1), type='b')
    plot(cumsum(x.pvar),xlab="Principal component", ylab="Cumulative Proportion of variance explained", ylim=c(0,1), type='b')
    screeplot(x)
    screeplot(x,type="l")
    par(mfrow=c(1,1))
}
```

USArrests dataset

Remember we must always scale and center the data. Luckily, R’s prcomp already has a parameter for this so we don’t have to do it manually. Let’s first take a quick look at our data set.

```
# First few rows of our data set
head(USArrests)
```

```
##            Murder Assault UrbanPop Rape
## Alabama      13.2     236       58 21.2
## Alaska       10.0     263       48 44.5
## Arizona       8.1     294       80 31.0
## Arkansas      8.8     190       50 19.5
## California    9.0     276       91 40.6
## Colorado      7.9     204       78 38.7
```

Now let’s apply PCA.

```
# Doing the PCA using prcomp
arrests.pca <- prcomp(USArrests,center = TRUE,scale. = TRUE)
names(arrests.pca)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

```
print(arrests.pca)
```

```
## Standard deviations (1, .., p=4):
## [1] 1.5748783 0.9948694 0.5971291 0.4164494
##
## Rotation (n x k) = (4 x 4):
##                 PC1        PC2        PC3         PC4
## Murder   -0.5358995  0.4181809 -0.3412327  0.64922780
## Assault  -0.5831836  0.1879856 -0.2681484 -0.74340748
## UrbanPop -0.2781909 -0.8728062 -0.3780158  0.13387773
## Rape     -0.5434321 -0.1673186  0.8177779  0.08902432
```

```
summary(arrests.pca)
```

```
## Importance of components%s:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.5749 0.9949 0.59713 0.41645
## Proportion of Variance 0.6201 0.2474 0.08914 0.04336
## Cumulative Proportion  0.6201 0.8675 0.95664 1.00000
```

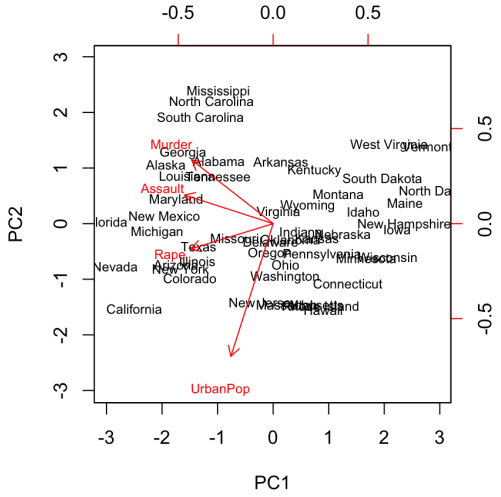
From the summary, we can see that the first principal component explains 62% of the variance in the data set, the second principal component explains 22% of the variance in the data set, and so on. Naturally, these will add up to 100%.

```
# Reference: http://webspace.ship.edu/pgmarr/Geo441/Lectures/Lec%2017%20-%20Principal%20Component%20Analysis.pdf
```

One very cool thing we can do now is actually graph our first two principal components on our X and Y axis, and see what our data looks like with the new axes.

Graph of data with PCA component axes

```
# Plotting our results with PC axes
biplot(arrests.pca,scale=0, cex=.7)
```



Now that we can get a clearer picture of the interrelations between our data, we can actually interpret it better.

Overall, we see that the crime-related variables are located close to each other, and that the urbanpop variable is located quite far from the other three. This indicates that the crime related variables are correlated with each other. States with high murder rates tend to have high assault and rape rates. Contrast this to the urabnpop variable, which is evidently less correlated with the other three.

```
# Reference: https://en.wikipedia.org/wiki/Principal_component_analysis
```

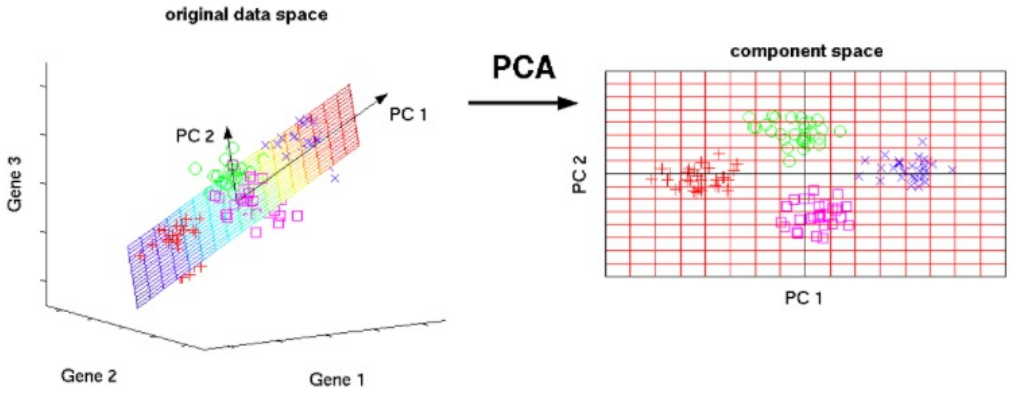
Conclusion

The message of my post is clear - learning about how to use Principal Components Analysis in depth is an important skill to have for any person looking to learn techniques in machine learning. While here we went over the definition, its use, and a basic example, there are many conceptual applications of PCA we haven't even gotten into. Feature selection is one very important application of PCA that is currently being developed in more detail. It is confirmed that PCA has the potential to perform feature selection and is able to select a number of important individuals from feature components.

```
# Reference: http://ieeexplore.ieee.org/document/5640135/
```

A more specific real world example of PCA being used today is in hospitals. PCA was instrumental in helping distinguish between patients with Schizophrenia and Healthy patients in a recent study done by the US National Library of Medicine.

Sample Hospital Data Example




```
# Reference: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2566788/
```

Evidently PCA is a very important technique for understanding more complex machine learning topics, and also for direct use in the real world. I hope this post convinced you of this, as well as taught you the basics of what PCA is and how to implement it in a simple example.

References

https://gab41.lab41.org/the-10-algorithms-machine-learning-engineers-need-to-know-f4bb63f5b2fa

https://www.dezyre.com/data-science-in-python-tutorial/principal-component-analysis-tutorial

https://www.coursera.org/specializations/deep-learning?utm_source=gg&utm_medium=sem&campaignid=904733485&adgroupid=45435009512&device=c&keyword=deep%20learning%20coursera&matchtype=b&network=g&d

https://sebastianraschka.com/images/faq/lda-vs-pca/pca.png

http://webspace.ship.edu/pgmarr/Geo441/Lectures/Lec%2017%20-%20Principal%20Component%20Analysis.pdf

https://en.wikipedia.org/wiki/Principal_component_analysis

http://ieeexplore.ieee.org/document/5640135/

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2566788/

http://www.stats.uwo.ca/faculty/braun/ss3850/notes/sas10.pdf

https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/