post01_Zirun_Wu

Zirun Wu

October 31, 2017

title

An in-depth overview of PCA and its application to economics

introduction

Principal Component Analysis is an enormously useful tool for statisticians and data scientists to process and analyze data, through dimensionality reduction techniques. However, we were only able to cover a small portion of what that truly entails, and its applications to other fields in class. Through this post, I hope to further delve into the background of PCA, the different functionalities it provides, and its application to economics, specifically portfolio management and stock returns.

motivation

As an economics major at Cal, I was always curious about more quantitative methods to investments and portfolio selection, and this is part of the reason I decided to take Stat 133. Now with the Principal Component Analysis tool, I could learn a new way to evaluate financial investments, marking an intersection of economics, statistics and data science that I previously had no idea could be possible.

background

Principal Component Analysis, or PCA, is a tool we use to extract the most important elements of a data set in order to reduce the dimensionality of data. Often times, we are only able to visualize data that is in 2 or 3 dimensions, but we will get data sets with a large number of variables, and it is up to us to use data processing tools such as PCA to make the data easier to work with.

mathematical foundation

Let's suppose we have n observations of p different variables. Therefore, we can put our data inside of a (nXp) matrix, containing n points in p dimensional space. At its essence, PCA projects the p dimensional data onto a k dimensional subspace that minimizes the residual sum of squares of the projection, thereby minimizing the sum of squared distances from points to their projections. In fact, we can accomplish this by maximizing the covariance matrix. Through linear algebra, we know that the covariance matrix is equal to CDC^-1, where C is a matrix with columns formed by eigenvectors of the covariance matrix, and D is a diagonal matrix of non-negative eigenvalues in descending order. Therefore, the first principal component is a linear combination of the original variables that captures the maximum variance in the data set, the 2nd principal component likewise captures the remaining variance in the data set, and so on. The correlation between the two should be zero, or orthogonal, in order to capture the maximum amount of variation in the dtaa.

in depth overview of PCA functionalities in R

In addition to the ideas and functions introduced in lecture and labs, there are a few more interesting functionalities of PCA that we can explore. #princomp() princomp() comes with the default "stats" package, and it is very similar to prcomp(), the function we used in class.

```
pca1 = princomp(USArrests, cor = TRUE)
pca1$sdev
```

```
##    Comp.1    Comp.2    Comp.3    Comp.4
## 1.5748783 0.9948694 0.5971291 0.4164494
```

PCA()

Most convenient PCA function in R, allows us to tweak the data in many ways.

```
library(FactoMineR)
```

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

```
pca2 = PCA(USArrests, graph = FALSE)
pca2$eig
```

```
##        eigenvalue percentage of variance cumulative percentage of variance
## comp 1  2.4802416              62.006039                          62.00604
## comp 2  0.9897652              24.744129                          86.75017
## comp 3  0.3565632               8.914080                          95.66425
## comp 4  0.1734301               4.335752                         100.00000
```

```
pca2$var$coord
```

```
##              Dim.1      Dim.2      Dim.3       Dim.4
## Murder   0.8439764 -0.4160354  0.2037600  0.27037052
## Assault  0.9184432 -0.1870211  0.1601192 -0.30959159
## UrbanPop 0.4381168  0.8683282  0.2257242  0.05575330
## Rape     0.8558394  0.1664602 -0.4883190  0.03707412
```

dudi.pca()

dudi.pca() provides other functionalities, including figuring out the correlation between different variables.

```
library(ade4)
```

```
## Warning: package 'ade4' was built under R version 3.4.2
```

```
##
## Attaching package: 'ade4'
```

```
## The following object is masked from 'package:FactoMineR':
##
##     reconst
```

```
pca3 = dudi.pca(USArrests, nf = 5, scannf = FALSE)
pca3$eig
```

```
## [1] 2.4802416 0.9897652 0.3565632 0.1734301
```

```
pca3$c1
```

```
##                 CS1        CS2        CS3         CS4
## 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
```

```
pca3$co
```

```
##               Comp1      Comp2      Comp3       Comp4
## Murder   -0.8439764  0.4160354 -0.2037600  0.27037052
## Assault  -0.9184432  0.1870211 -0.1601192 -0.30959159
## UrbanPop -0.4381168 -0.8683282 -0.2257242  0.05575330
## Rape     -0.8558394 -0.1664602  0.4883190  0.03707412
```

prcomp()

What we used in class, quickest way to do PCA analysis.

```
pca4 = prcomp(USArrests, scale. = TRUE)
pca4$sdev
```

```
## [1] 1.5748783 0.9948694 0.5971291 0.4164494
```

acp()

Lastly, we could also use the acp() function from the package 'amap''

```
library(amap)
pca5 = acp(USArrests)
pca5$sdev
```

```
##    Comp 1    Comp 2    Comp 3    Comp 4
## 1.5748783 0.9948694 0.5971291 0.4164494
```

```
pca5$loadings
```

```
##             Comp 1     Comp 2     Comp 3      Comp 4
## 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
```

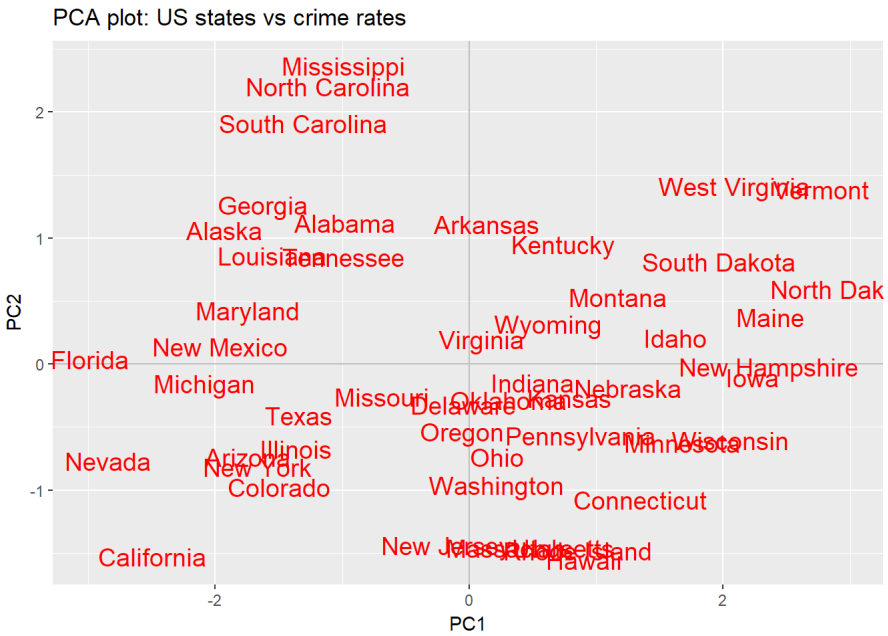
PCA Plots

In order to better visualize, we can use ggplot2, another tool we have encountered in this class, to plot our results from the PCA analysis.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.2
```

```
scores = as.data.frame(pca4$x)
ggplot(data = scores, aes(x = PC1, y = PC2, label = rownames(scores))) +
  geom_hline(yintercept = 0, colour = "gray") +
  geom_vline(xintercept = 0, colour = "gray") +
  geom_text(colour = "red", alpha = 1, size = 5) +
  ggtitle("PCA plot: US states vs crime rates")
```



#application of PCA to Bond

Portfolio management In bond portfolio management, there are already two instances in which PCA has been employed: -explaining the movement in the returns yield curve, and applying PCA to measure and manage the risk associated with the yield curve -identify risk factors beyond changes in the term structure For example, in determining the key factors that are truly affecting bond returns, we can take historical bond returns data, along with a set of variables that are believed to affect the rate of bond returns. Then, we can perform PCA analysis to obtain the Principal Components, comprised of linear combinations of these variables, that explain the majority of the variation. Gauthier and Goodman, working for Salomon Smith Barney Broad Investment, used PCA to identify risk factors that generated the bulk of the returns from 1992 to 2003. The first PC explained 92.7% of the variation, and the second PC explained 3.1% of the variation.

application of PCA to controlling interest rate risk

Using PCA, economists were able to isolate three variables that together explained the bulk of the change in historical returns of the US treasury portfolios, which were -changes in level of rates -changes in slope of the yield curve -changes in curvature of the yield curve Empirically, the first PC explained 90% of the variability, the second PC explained 8%, and the third PC explained about 2%. Using PCA to evaluate the dynamics of the yield curve for investments is now the principal component duration, estimating the probability associated with potential interest rate shocks and its effect on interest rate risk of a bond portfolio.

application of PCA to stock market

Let's say we are looking at the returns of S&P 500 companies, and we would like to find a few variables that would be representative of the set of all 500 companies. We can use PCA to accomplish this:

```
returns <- read.csv("/Users/kevin/Downloads/ind_nifty500list.csv") #a doc containing returns of S&P500 companies
#for(i in 2:ncol(returns))
#{returns1[, i] <- approx(returns$Year, returns1[ ,i], returns$Year)$y}
#ret <- as.matrix(returns1, nrow = dim(returns1)[1], ncol=dim(returns1)[2])
#princ.return <- princomp(ret)
#barplot(height=princ.return$sdev[1:10]/princ.return$sdev[1])
```

However, a quick note of caution: only use PCA if there are a lot of macroeconomic variables, and there is a high correlation between them.

conclusion - key takeaway

PCA is a very useful tool for data scientists to reduce variability in data so we can better visualize it. It has a solid mathematical foundation in linear algebra, and it's builtupon the idea of eigenvalues and eigenvectors. It could be expressed in R in several different forms, from the prcomp() that we are used to, to other forms with more functionalities. Lastly, PCA can be applied to many different areas, one of which is economics, which uses PCA to calculate risk, returns, and could be a very useful tool in portfolio management and for analyzing the stock market. Overall, this was a unique learning experience, through the application of something we learned in stat 133 to something that closely

relates to what my field of interest mainly lies.

references

https://tgmstat.wordpress.com/2013/11/21/introduction-to-principal-component-analysis-pca/
https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/
https://tgmstat.wordpress.com/2013/11/28/computing-and-visualizing-pca-in-r/ https://www.r-bloggers.com/principal-component-analysis-in-r/
http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/118-principal-component-analysis-in-r-prcomp-vs-princomp/ http://www.gastonsanchez.com/visually-enforced/how-to/2012/06/17/PCA-in-R/
https://statistik.econ.kit.edu/download/doc_secure1/Lecture13FinancialTimeSeries.pdf https://programming-r-pro-bro.blogspot.com/2011/10/principal-component-analysis-use.html https://www.r-bloggers.com/principal-component-analysis-use-extended-to-financial-economics-part-2/