Principle Components Analysis (PCA)

Rebecca C. Steorts, Duke University

STA 325, Chapter 10 ISL

Cars dataset

We're going to look at a small dataset of 388 cars from the 2004 model year, with 18 features, with incomplete records removed). Eight features are binary indicators. The other 11 features are numerical. All of the features except `Type` are numerical. PCA only works with numerical features, so we have ten of them to play with. Below we load in the data and show the first few lines of the data.

```
library(knitr)
cars04 = read.csv("cars-fixed04.dat")
kable(head(cars04))
```

	Sports	SUV	Wagon	Minivan	Pickup	AWD	RWD	Retail	Dealer	Engine	Cylin
Acura 3.5 RL	0	0	0	0	0	0	0	43755	39014	3.5	
Acura 3.5 RL Navigation	0	0	0	0	0	0	0	46100	41100	3.5	
Acura MDX	0	1	0	0	0	1	0	36945	33337	3.5	
Acura NSX S	1	0	0	0	0	0	1	89765	79978	3.2	
Acura RSX	0	0	0	0	0	0	0	23820	21761	2.0	
Acura TL	0	0	0	0	0	0	0	33195	30299	3.2	

Let's now run PCA on the dataset and be sure to scale all the variables to have variance 1.

```
cars04.pca <- prcomp(cars04[,8:18], scale.=TRUE)
```

Let's now extract the weight or loading matrix from the `cars04.pca` object. Specifically, let's just look at the first two principal components.

```
# grab loadings, look at first two PCs
round(cars04.pca$rotation[,1:2], 2)
```

```
##              PC1   PC2
## Retail     -0.26 -0.47
## Dealer     -0.26 -0.47
## Engine     -0.35  0.02
## Cylinders  -0.33 -0.08
## Horsepower -0.32 -0.29
## CityMPG     0.31  0.00
## HighwayMPG  0.31  0.01
## Weight     -0.34  0.17
## Wheelbase  -0.27  0.42
## Length     -0.26  0.41
## Width      -0.30  0.31
```

How can we interpret these two components?

Our results suggest that all the variables except the gas-mileages have a negative projection on to the first component. This means that there is a negative correlation between mileage and everything else. The first principal component tells us about whether we are getting a big, expensive gas-guzzling car with a powerful engine, or whether we are getting a small, cheap, fuel-efficient car with a wimpy engine.

The second component is a little more interesting. Engine size and gas mileage hardly project on to it at all. Instead we have a contrast between the physical size of the car (positive projection) and the price and horsepower. Basically, this axis separates mini-vans, trucks and SUVs (big, not so expensive, not so much horse-power) from sports-cars (small, expensive, lots of horse-power).

How can we see this more clearly? We can look at visualizations, such as a bi-plot. A bi-plot plots the data along with the projections of the original features, on to the first two components.

```
biplot(cars04.pca, scale=0, cex=0.4)
```

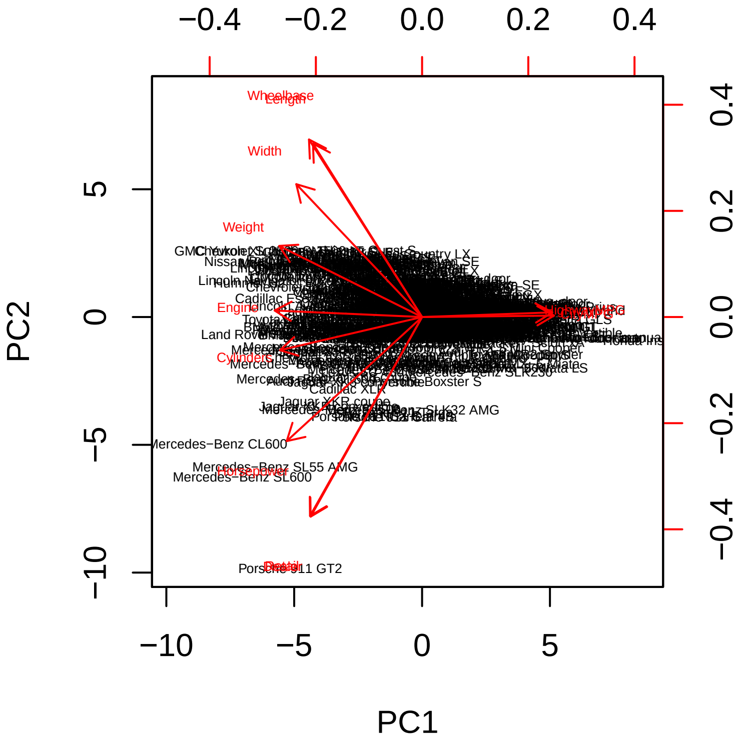

Figure: "Biplot" of the 2004 cars data. The horizontal axis shows projections on to the first principal component, the vertical axis the second component. Car names are written at their projections on to the components (using the coordinate scales on the top and the right). Red arrows show the projections of the original features on to the principal components (using the coordinate scales on the bottom and on the left).

Notice that the car with the lowest value of the second component is a Porsche 911, with pick-up trucks and mini-vans at the other end of the scale. Similarly, the highest values of the first component all belong to hybrids.

How to Interpret PCA Plots

There is a more-or-less standard recipe for interpreting PCA plots, which goes as follows.

To begin with, find the first two principal components of your data. (I say "two" only because that's what you can plot; see below.) It's generally a good idea to standardized all the features first, but not strictly necessary.

Coordinates Using the arrows, summarize what each component means. For the cars, the first component is something like size vs. fuel economy, and the second is something like sporty vs. boxy.

Correlations For many datasets, the arrows cluster into groups of highly correlated attributes. Describe these attributes. Also determine the overall level of correlation (given by the R^2 value). Here we get groups of arrows like the two MPGs (unsurprising), retail and dealer price (ditto) and the physical dimensions of the car (maybe a bit more interesting).

Clusters Clusters indicate a preference for particular combinations of attribute values. Summarize each cluster by its prototypical member. For the cars data, we see a cluster of very similar values for sports-cars, for instance, slightly below the main blob of data.

Funnels Funnels are wide at one end and narrow at the other. They happen when one dimension affects the variance of another, orthogonal dimension. Thus, even though the components are uncorrelated (because they are perpendicular) they still affect each other. (They are uncorrelated but not *independent*.) The cars data has a funnel, showing that small cars are similar in sportiness, while large cars are more varied.

Voids Voids are areas inside the range of the data which are unusually unpopulated. A **permutation plot** is a good way to spot voids. (Randomly permute the data in each column, and see if any new areas become occupied.) For the cars data, there is a void of sporty cars which are very small or very large. This suggests that such cars are undesirable or difficult to make.

Projections on to the first two or three principal components can be visualized; however they may not be enough to really give a good summary of the data. Usually, to get an R^2 of 1, you need to use all p principal components.[1] How many principal components you should use depends on your data, and how big an R^2 you need. In some fields, you can get better than 80% of the variance described with just two or three components. A sometimes-useful device is to plot $1 - R^2$ versus the number of components, and keep extending the curve it until it flattens out.

[1] The exceptions are when some of your features are linear combinations of the others, so that you don't really have p *different* features, or when $n < p$.