Principal Components Analysis

36-290 – Statistical Research Methodology

Week 4 Tuesday – Fall 2021

The Setting

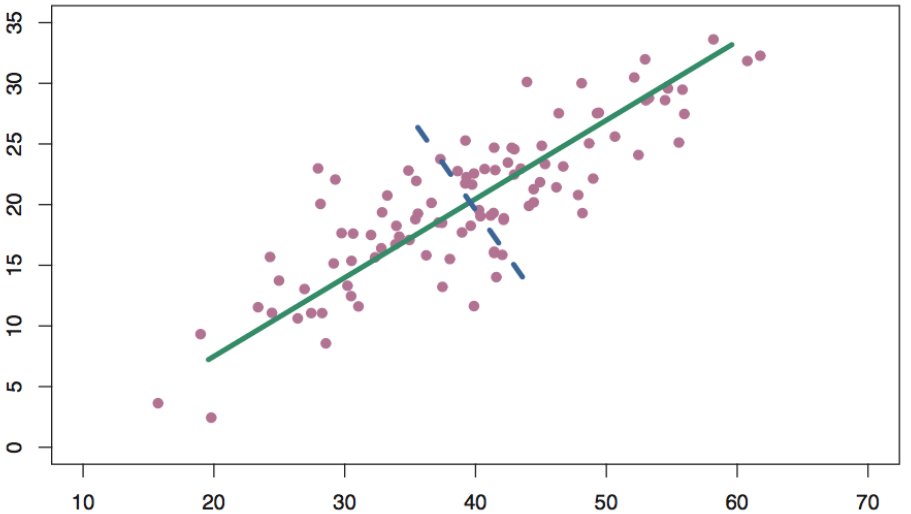
The setting for *principal components analysis* or *PCA* is similar to the setting for unsupervised learning: you have a collection of p measurements for each of n objects $X_1, \ldots, X_n$, where for a given object i

$$X_{i1}, X_{i2}, \ldots, X_{ip} \sim P$$

where P is some p-dimensional distribution that we might not know much about *a priori*. Note that when we apply PCA, the measurements X may be all the data, or may represent the predictors if there is a response variable.

PCA

Let's build up intuition for PCA by starting off with a picture:



Here, p = 2. What does the PCA algorithm do? Effectively, it moves the coordinate system origin to the centroid of the point cloud, then rotates the axes such that "PC Axis 1" lies along the direction of greatest variation (the solid line above), and "PC Axis 2" lies orthogonal to to "PC Axis 1." If $p > 2$, then "PC Axis 2" would lie along the direction of greatest residual variation, and "PC Axis 3" would then be defined orthogonal to axes 1 and 2, etc.

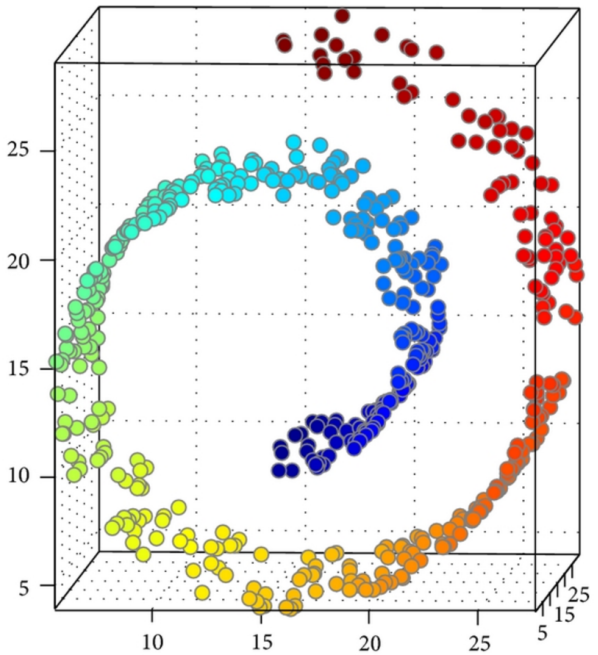
PCA as Dimension Reduction Tool

Imagine that we have a p-dimensional point cloud that we wish to visualize. One way to do this is to utilize PCA "to find a low-dimensional representation of the data that captures as much of the [statistical] information [present] as possible."

For instance, if it would so happen that the data in our p-dimensional space actually all lie on a two-dimensional plane embedded within that space, PCA would uncover this structure and we would subsequently be able to visualize the data using two-dimensional scatter plots. (Another possibility is to perform, e.g., linear regression using a subset of principal components rather than the original data frame. "Principal components regression" is covered in ISLR but not explicitly covered in this class.)

In this example, the key word is "[hyper]plane." The main limitation of PCA is that it is a *linear* algorithm: it projects data to hyperplanes. If the data inhabit a curved surface within the p-dimensional space, PCA would not be the optimal algorithm by which to explore the data. In this case, we'd use *nonlinear* techniques like diffusion map or local linear embedding, or self-organizing maps or tSNE. These are beyond the scope of the class.

PCA: It Won't Work Well For, e.g., the Swiss Roll



These data lie on a curving two-dimensional strip embedded in a three-dimensional space. Projecting these data to a two-dimensional plane or a one-dimensional line would effective randomize the statistical information they contain (i.e., the projection would not preserve the color ordering, which might represent the value of, say, a response variable).

PCA: What It Is Not

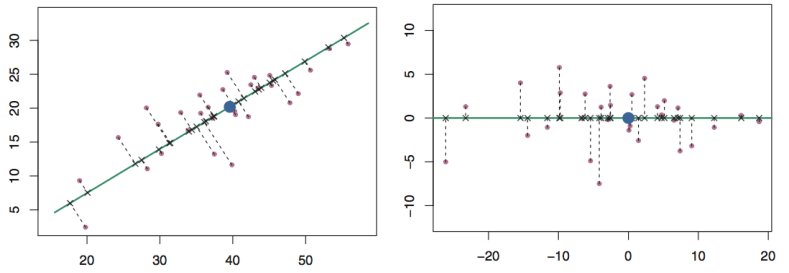
PCA is one of a family of related algorithms commonly dubbed *factor analyses*, and, for instance, some use the phrases "PCA" and "factor analysis" interchangeably. Note that:

1. The PCA algorithm is a **deterministic algorithm**. The algorithm does not take into account that the data to which it is applied are random variables; it just ingests the data values as they are and produces an answer, full stop.

2. *Exploratory factor analysis* is a variant of PCA which *does* attempt to take randomness into account. The idea is to map the p observed variables to $k < p$ factors; "exploratory" means we do not know *a priori* how that mapping may occur. Contrast this with...

3. *Confirmatory factor analysis*, which basically adds on a hypothesis-testing layer for confirming that links actually exist between the k factors uncovered by EFA and the p original variables.

PCA: Algorithm



The "score" or coordinate of the i^{th} observation along PC j is

$$Z_{ij} = \sum_{k=1}^{p} X_{ik}\phi_{kj}$$

where k represents the k^{th} variable or feature (i.e., the k^{th} column of your [predictor] data frame). The algorithm determines the rotation (or loading) matrix ϕ, which is normalized such that $\sum_{k=1}^{p} \phi_{kj}^2 = 1$. Note that j ranges from 1 to p.

Note that since we are "mixing axes" when doing PCA, it is generally best to standardize (or scale) the data frame before applying the algorithm.

PCA: Algorithm (Deeper Detail)

PCA proceeds by factoring the (scaled) data frame via *singular value decomposition*, or *SVD*:

$$X = UDV^T$$

where X is the scaled data frame, U and V are eigenvector matrices, and D is the diagonal matrix of eigenvalues. (V^T means "the transpose of V.") The PC coordinates are then $Z = XV$, meaning that ϕ from the last slide is the matrix of eigenvalues V.

Note that PCA will really only work if n, the number of rows in your data frame, is greater than p, the number of variables.

PCA: Choosing the Number of Dimensions to Retain

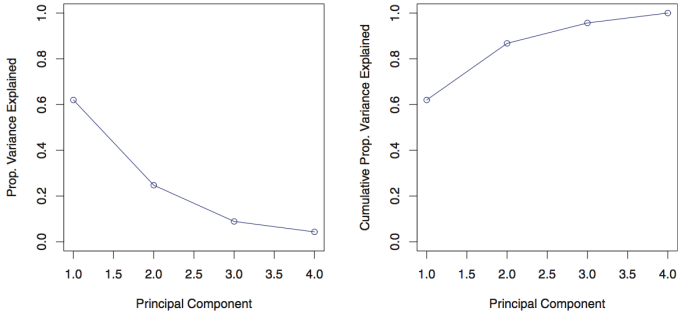
Well, that's the million-dollar question now, isn't it? And guess what: there is no simple answer to this! ("Embrace the Ambiguity" indeed.)

The ideal would be the following: choose $M < p$ such that

$$x_{ij} = \sum_{m=1}^{p} z_{im}\phi_{jm} \approx \sum_{m=1}^{M} z_{im}\phi_{jm} \, .$$

In other words, we don't lose much ability to reconstruct the input data X by dropping the last $p - M$ principal components, which we assume represent random variation in the data (i.e., noise).

One convention: sum up the amount of variance explained in the first M PCs, and adopt the smallest value of M such that 90% or 95% or 99%, etc., of the overall variance is "explained."

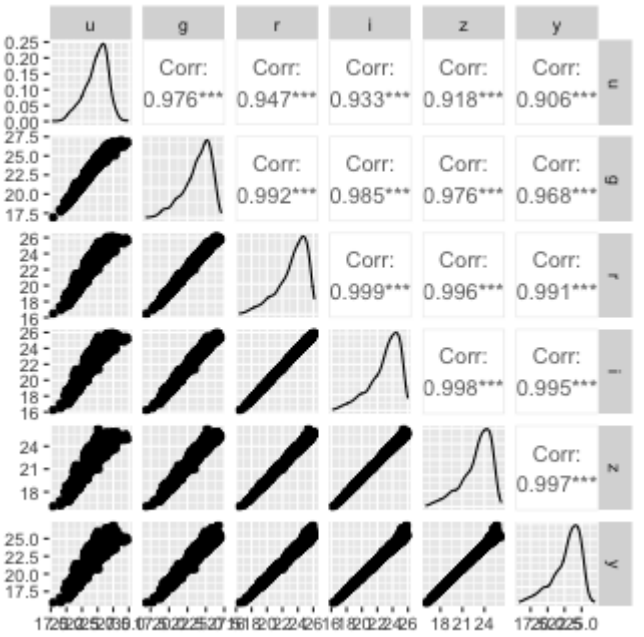


Another convention: look for an "elbow" in the PC scree plot (the left panel above). An elbow is where the percentage of variance explained transitions from falling off rapidly to falling off slowly. Above, it is not necessarily clear if an elbow exists and if it does, where it exactly is.

PCA: Example

Below we load in examples of galaxy magnitudes in different wavelength bands, from u (for ultraviolet) to z and y in the near-infrared. Because a galaxy that is bright (has low magnitude) in one band tends to be bright in all bands, we see that the magnitude data are obviously correlated.

```
suppressMessages(library(GGally))
ggpairs(df,progress=FALSE)
```



PCA: Example

```
pca.out = prcomp(df,scale=TRUE)
v = pca.out$sdev^2
round(cumsum(v/sum(v)),3)
```

```
## [1] 0.977 0.998 0.999 1.000 1.000 1.000
```

What we observe is that the first principal component explains 97.7% of the variance in the dataset: the data can safely be transformed from a six-dimensional space to a one-dimensional one with minimal loss of statistical information.

What do we do now? Let's print the column(s) for the PCs that we retain:

```
round(pca.out$rotation[,1],3)
```

```
##     u     g     r     i     z     y
## 0.396 0.411 0.413 0.412 0.410 0.408
```

What we would do is *inference*: if we retain only this PC, what are the variables that map most strongly to it? (Sign doesn't matter: we just want to know which variables are associated with the largest numbers for each PC.) Here, we see that all the original variables contribute almost equally to the first PC.

We'd conclude that we can view the data as lying along a one-dimensional line that exists in the native six-dimensional space, and that the orientation of the line is such that it spans all six of the original dimensions, or that all six of the original variables appear to contain useful statistical information. Full stop.

PCA: Example

Before we leave: what information is in that second PC we ignored, but could have kept?

```
round(pca.out$rotation[,1:2],3)
```

```
##     PC1    PC2
## u 0.396  0.793
## g 0.411  0.247
## r 0.413 -0.089
## i 0.412 -0.205
## z 0.410 -0.320
## y 0.408 -0.398
```

We see that the second PC primarily maps to u-band magnitude...basically, the PC 2 vector, orthogonal to the PC 1 vector, points largely along the u-band axis, meaning there is some variability along that axis that PC 1 did not pick up. So PC 2 is primarily about the u band. (To build intuition, go back to the `ggpairs` plot, and you'll see that the u-band data tends to be "fuzzier" [scientific term] than the data in the other bands. So the PC 2 direction makes sense.) If you decided to retain the second PC, the above information is all the client needs.

Some Parting Notes

ISLR: "In practice, we tend to look at the first few principal components in order to find interesting patterns in the data. If no interesting patterns are found in the first few principal components, then further principal components are unlikely to be of interest."

Note: PCA is **not** a variable selection tool. For instance, if you have a dataset with eight predictor variables, did PCA, discovered you only needed to retain two PCs, and saw that only four predictor variables were represented in those two PCs...you would *not* select those variables and go on to do, e.g., linear regression using only those four variables. However, you *can* do PC regression using the first two PCs. We will come back to why you might want to do that in a future lecture.

To recap: you can use PCA to select (and perhaps further use) PCs, but not to select a subset of the original variables.

Context: Where Does PCA Fit In?