Principal Components

1. Introduction

Principle components are linearly independent combinations of selected continuous data variables. Principal components are calculated variables that we can use in models, graphics and as basis for sorting cases.

Common convention numbers the principle component is decreasing variance order. The sum of variances for the first few principal components may be substantial percent such as 95% of the sum of data variable variances.

Models: A common practice uses the first few principal components in a regression model in place of the larger number variables in the data sets. In this sense the principal components provide variable reduction.

Graphics: We can plot one, two, or three principle components in one, two or three dimensional scatterplot plot to look for patterns such as clusters of cases. We can plot more principal components using scatterplot matrices and parallel coordinate plots.

In spatial studies such as in studying atmospheric patterns over oceans, researcher may use a sequence of choropleth maps. The earth grid cell colors will encode values of single selected principal component.

A non-standard plot I suggest for 4-D data is the 4-D ray glyph plot. This is 3-D scatterplot with depth conveyed by slow rotation and/or stereo parallax. This is augmented by a ray glyph that encodes the 4^{th} variable. The ray direction from -90 to 90 degrees encode values from the

minimum to the maximum. The line segments from the 3-D dot always appear parallel to the plane of the display.

Sorting: The first principle component provides a basis for sorting cases. (If there are ties these can be broken by the second principal component.) Ordering cases is sometime called seriation and can be useful in linked micromaps and many other graphics. This includes heat maps if one should chose to produce them.

The R seriation package has many algorithms for ranking cases. This include traveling salesman ordering that in my opinion is competitive to the first principal component order. Another competitive algorithm is the breadth traversal of a minimal spanning tree available in SPlus. Unfortunately the mstree() function was not ported R. In have compared sorting results with a modest number of data sets and would pick one of the three be include to pick one of these three algorithms. For me the clustering dendrogram order was not competitive.

When the variables are in the same data units we may a good reason use the min, mean, median, max or other computed value for each case to sort the cases. There may an available index or variable that may provide a contextual basis for sort cases. Lacking a context driven logical choice, I suggest first trying the first principal component and then perhaps, if there is time, compare this to the traveling salesman order.

2. Principal component construction choice

The first step in constructing principal components is to center the variables (subtract each variable's mean so the new mean is zero. The R prcomp() function passes the variables to R's scale function along with argument default argument center=TRUE and scale = FALSE. **The first basic choice is whether to set the scale argument to TRUE or**

leave it as FALSE. Scaling divides the centered variables by their standard deviations. When the variables have different units of measure, the choice is usually to used scale =TRUE. This makes the new variables unitless. The correlation matrix becomes the foundation for constructing principal components. We can then interpret a principal component in terms the coefficients in the linear combination of variables and the variables names were the coefficients in terms of unitless standard deviations.

Most researchers choose to divide by the standard deviations in the different units of measure context. If the data is in a matrix called mat, the R script is typically prcomp(mat, scale=TRUE)

When all of the variables are in the same unit of measure we may choose to leave the scale argument as FALSE and the principal component will be based on the covariance matrix. We will interpret linear combination coefficients in the same unit of measure as the variables. This can be advantageous when researchers are very familiar with the variables. Other may prefer to stay with basing the principal components on the correlations matrix.

3. Selecting the number of principal components to use.

Typically we select principal components with the largest variances to use for further analysis tasks. The based question is how many to use. An R a screeplot shows the variances for the principal component in descending variance order. We might select the smallest number of components that combined have at least 80% of the variance relative the sum of variances for all the original variables. We may stop when adding the next principal component begins adding a marginal increase of the cumulative percentage of variance to total varianace.

4. Interpretation of principal components in terms of the original variables.

I look at the coefficients of the variables and often ignore the small magnitude ones to provide relatively simple description. In homework with male and female cancer rates, the first principal component is primarily the sum of the two variables with the male variable having a higher weight than the female variable since the male variance is larger. This rough summing of variables pattern occurs fairly often even when there are many more variables.

Another pattern is a contrast among a subset of variables. Technically the sum of the coefficients for a contrast is zero but the term is used loosely here. The coefficients for 10 variables might be (-0.31, 0.43, small, 0.52, -0.38, small, -0.37, small, small, small). This could be loosely interpreted as average of variables 2 and 4 minus the average of variables 1, 5 and 7. Contrasts may make sense to the researcher who knows the variables and their interactions.

5. Notes

5.1 Rotating the scaled data matrix and principal components

Here the scaled data matrix refers to the matrix returned to prcomp() from the scale() function. After this the prcomp() function uses the singular value decomposition function svd() to produce a “rotation” matrix and a matrix of principal components.

The rotation matrix columns provide coefficients for the linear combination of the scaled data matrix columns. When the scaled data matrix has linearly independent columns post multiplying this n x c matrix by the c x c “rotation” matrix produces the principal component matrix.

The rotation matrix is construction follows the convention principle components are number in decrease variance order. The first column in the principle component matrix will have the large variance.

A rotation matrix has a determinant of 1. The quotes around the word rotation indicate that the matrix many include an odd number of reflections and hence may have determinant of -1. A reflection multiples a principal component by -1 so changes positive values to negative values and vice versa The principal components remain linearly independent and their variances remain the same. The home work male and female mortality rate provide an example with a determinant of -1.

The situation is a little different when the scaled data matrix columns are linearly dependent. The "rotation" matrix may drop columns because using them would just produce principal component with all zero values.

5.2 Comments on the scale() function

The scale() function is useful in many settings. The center and scale arguments allow us to provide vectors or functions that evaluate to vectors as argument values. For example if x is matrix of values, consider

scale(x, center= apply(x, 2, min), scale= diff(apply(x, 2, range)).

This linearly scales all variables to fall in the interval [0, 1]. We could change the min() function to the median() function and the range function to the IQR() function to obtain a more resistant to outliers centering and scaling of the variable.