PCA - Principal Component Analysis

Introduction

Principal Components Analysis (PCA) is a multivariate procedure which rotates the data such that maximum variabilities are projected onto the axes. Essentially, a set of correlated variables are transformed into a set of uncorrelated variables which are ordered by reducing variability. The uncorrelated variables are linear combinations of the original variables, and the last of these variables can be removed with minimum loss of real data.

The main use of PCA is to reduce the dimensionality of a data set while retaining as much information as is possible. It computes a compact and optimal description of the data set.

The first principal component is the combination of variables that explains the greatest amount of variation. The second principal component defines the next largest amount of variation and is independent to the first principal component. There can be as many possible principal components as there are variables.

It can be viewed as a rotation of the existing axes to new positions in the space defined by the original variables. In this new rotation, there will be no correlation between the new variables defined by the rotation. The first new variable contains the maximum amount of variation, the second new variable contains the maximum amount of variation unexplained by the first and orthogonal to the first, etc...

There are several algorithms for calculating the Principal Components. Given the same starting data they will produce the same results with the one exception (are you surprised?). This exception is that, if at some point, there are two or more possible rotations that contain the same "maximum" variation, then which one is used is indeterminate. In two dimensions the data cloud would look like a circle, instead of an ellipse. In a circle, any rotation would be equivalent. In an elliptical data cloud, the first component would be parallel to the major axis of the ellipse.

It can be viewed as finding a projection of the observations onto orthogonal axes contained in the space defined by the original variables. The criteria being that the first axis "contains" the maximum amount of variation, or "accounts" for the maximum amount of variation. The second axis contains the maximum amount of variation orthogonal to the first. The third axis contains the maximum amount of variation orthogonal to the first and second axis and so on until one has the last new axis which is the last amount of variation left. As you can see these are really two slightly different ways of saying the same thing!

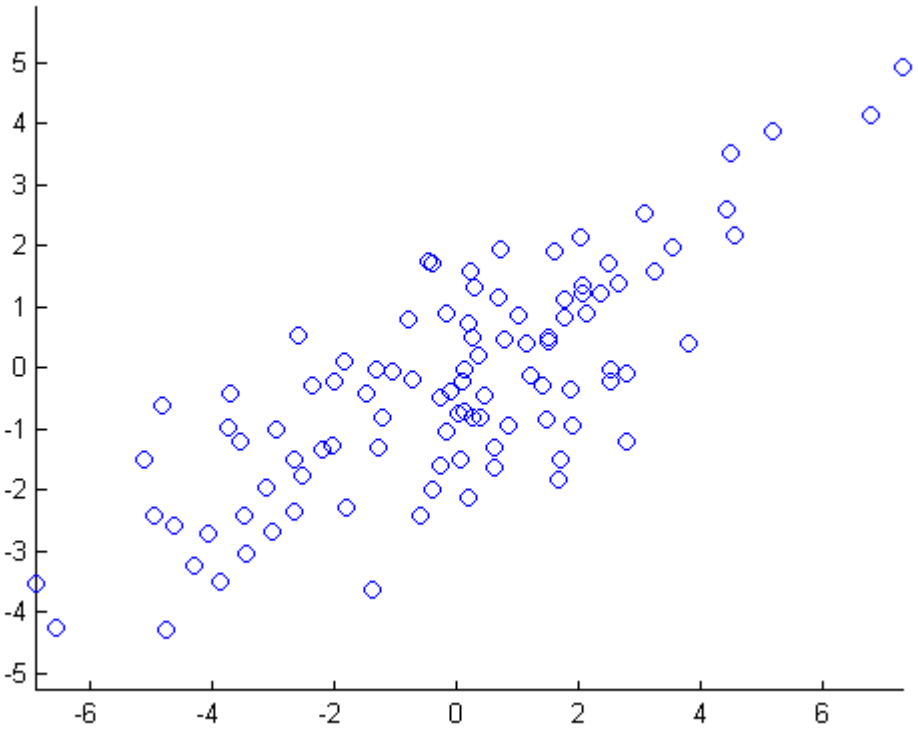
Simple Example

In this example, we take a simple set of 2-D data and apply PCA to determine the principal axes. Although the technique is used with many dimensions, 2 dimensional data will make it simpler to visualise.

First of all we generate a set of data for analysis:

```
x =

   -3.0139   -2.6748
    0.1810   -2.1227
   -6.5559   -4.2424
    1.5960    1.9161
    2.7825   -1.2017
    2.0292    2.1421
   -2.5200   -1.7787
    0.2340   -0.8122
    0.8263   -0.9407
   -4.6227   -2.5955
   -4.3052   -3.2307
   -2.6013    0.5435
    3.5409    1.9755
    5.1704    3.8629
      .         .
      .         .
      .         .
```



The Principal Component Analysis is then performed. First the covariance matrix is calculated:

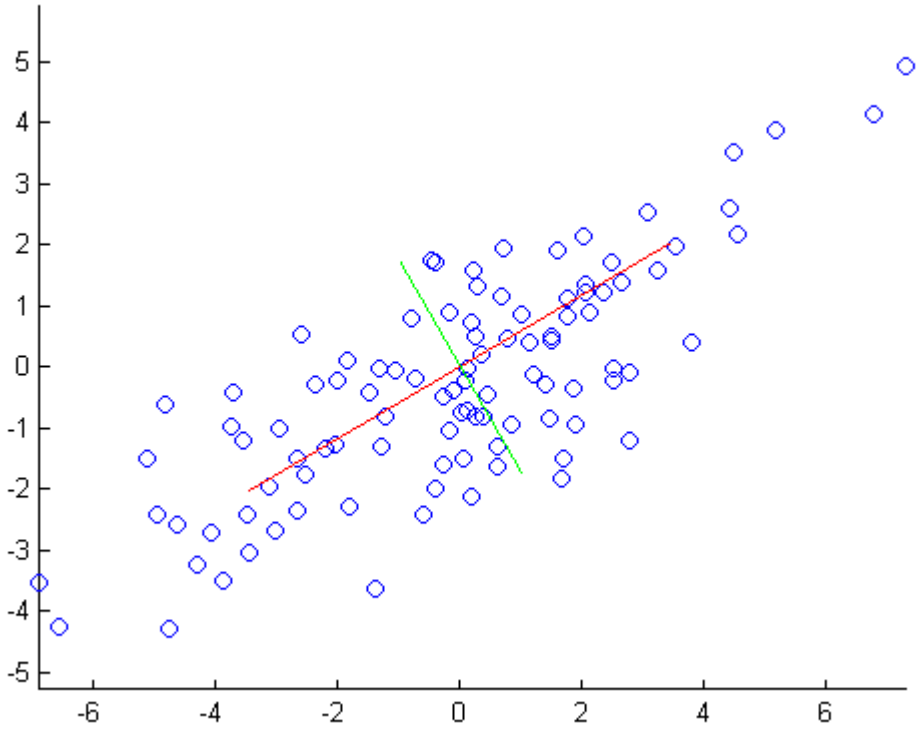
```
covx =

    7.5649    3.8464
    3.8464    3.2451
```

From this the PC's are calculated:

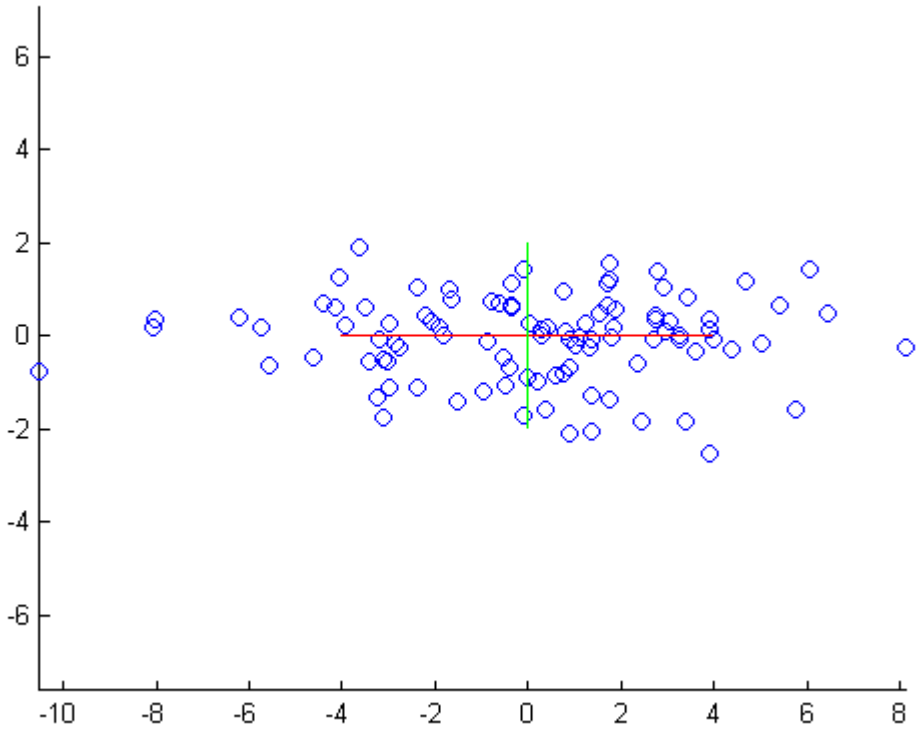
```
pc =

    0.8630   -0.5052
    0.5052    0.8630
```

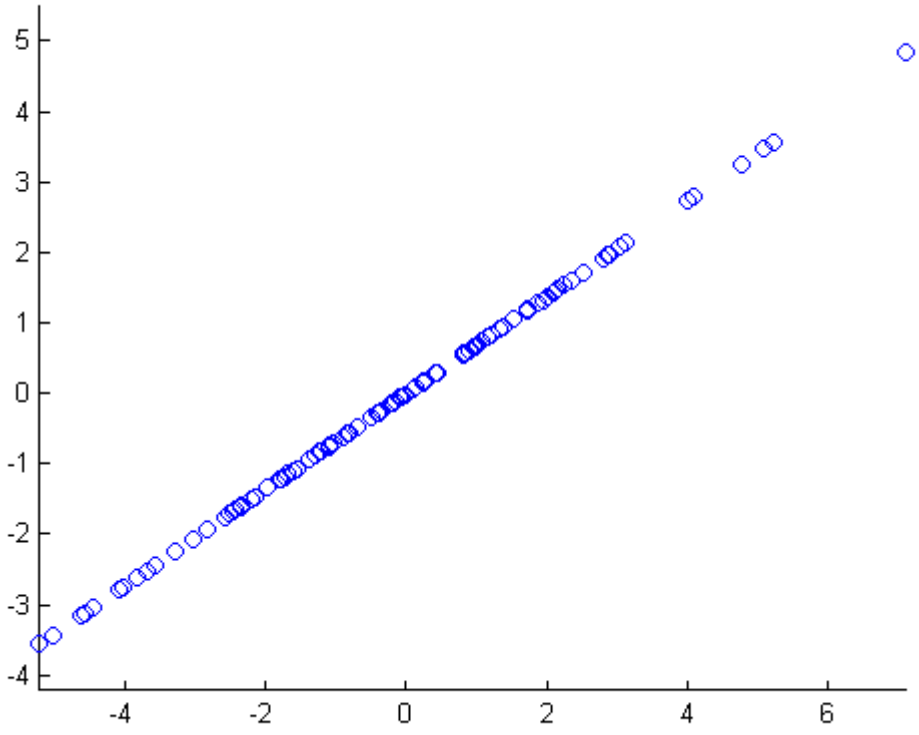


The red line represents the direction of the first principal component and the green is the second. Note how the first principal component lies along the lineo of greatest variation, and the second lies perpendicular to it. Where there are more than two dimensions, the second component will be both perpendicular to the first, and along the line of next greatest variation.

By multiplying the original data-set by the principal components, the data is rotated so that the PC's lie along the axes:



The most common use for PCA is to reduce the dimensionality of the data while retaining the most information.



MATLAB Code

The following MATLAB code generates the above images:

```
% Clear all variables and figures.
  clear all;
  close all;

% Generate elliptical cloud of data-poitns.
  x(1,:) = randn(1,100);
  x(2,:) = randn(1,100)*3;

% Rotate the cloud for demonstration.
  [p(1,:),p(2,:)] = cart2pol(x(1,:),x(2,:));
  p(1,:) = p(1,:)-pi/3;
  [x(1,:),x(2,:)] = pol2cart(p(1,:),p(2,:));

% Plot data.
  scatter(x(1,:),x(2,:));
  axis equal;
  drawnow;
  pause;

% Calculate PC's.
  [pc, latent, explained] = pcacov(cov(x'));

% Draw PC's on top of data.
  hold on;
  plot([-4 4]*pc(1,1),[-4 4]*pc(2,1),'r-');
  plot([-2 2]*pc(1,2),[-2 2]*pc(2,2),'g-');
  pause;

% Rotate the data to the PC's
  y = (x'*pc)';

% Plot data.
  figure;
  scatter(y(1,:),y(2,:));
  axis equal;
  drawnow;
  pause;

% Calculate PC's, to demonstrate they now lie on the axes.
  [pc2, latent, explained] = pcacov(cov(y'));

% Draw PC's on top of data.
  hold on;
  plot([-4 4]*pc2(1,1),[-4 4]*pc2(2,1),'r-');
  plot([-2 2]*pc2(1,2),[-2 2]*pc2(2,2),'g-');
  pause;

% Set the second component of y to zero, reducing the dimensionality to one.
  y(2,:) = 0;

% Transform back to the original data.
  x = (y'*inv(pc))';

% Plot data.
  figure;
  scatter(x(1,:),x(2,:));
  axis equal;
  drawnow;
  pause;

% Tidy up.
  close all;
```

Related Case Studies

Vibration Monitoring - This is an example of using PCA for data reduction.
Eigenflames - This is an example of using PCa for data analysis.

Limitations

The technique is linear, therefore any non-linear correlation between variables will not be captured.