**Principal Components Analysis Tutorial**

1. **Introduction:**

**Principal component analysis (PCA)** is a mathematical procedure that uses an [orthogonal transformation](http://en.wikipedia.org/wiki/Orthogonal_matrix) to convert a set of observations of possibly correlated variables into a set of values of [linearly uncorrelated](http://en.wikipedia.org/wiki/Correlation_and_dependence) variables called **principal components**.

By this way the original data set can be presented in the less number of principal components than the original variables. The first few components are based on the largest variances of the data. Therefore the information of the data is concentrated on the first few principal components, even we lost the last components we can still reconstruct the original data very well.

PCA is eigenvector-based multivariate analyses. Its operation can be thought of as revealing the internal structure of the data in a way that best explains the variance in the data.

Applications of PCA are about to reduce the dimension of the original data, face recognition and so on which we will talk about in detail below.

1. **Background Mathematics**
   1. **Statistics**

Analyze of a set in terms of the relationships between the individual points in a dataset.

**Standard Deviation:**

Mean of the sample:

=

Standard Deviation (SD) of a dataset is a measure of how spread out the data is

*s =*

**Variance:**

Variance is Similar as SD which is just the square of it. Both of them are measures of the spread of the data.

*s2* =

**Covariance:**

Variance is a measure of the deviation from the mean for points in one dimension.

Covariance is a measure of how much each of the dimensions varies from the mean with respect to each other.

Covariance is used to measure the linear relationship between 2 dimensions.

* Positive value:

--Both dimensions increase or decrease together.

* Negative value:

--One increases and the other decreases, or vice-versa

* Zero

--The two dimensions are independent

of each other

**The Covariance Matrix:**

A matrix has the covariance of any two variables in a high dimension dataset.

e.g. for 3 dimensions:

Properties:

∑;

∑ is [positive-semidefinite](http://en.wikipedia.org/wiki/Positive-semidefinite_matrix) and [symmetric](http://en.wikipedia.org/wiki/Symmetric_matrix);

;

**Correlation :**

It can refer to any departure of two or more random variables from independence, but technically it refers to any of several more specialized types of relationship between [mean values](http://en.wikipedia.org/wiki/Conditional_expectation).

It is obtained by dividing the [covariance](http://en.wikipedia.org/wiki/Covariance) of the two variables by the product of their [standard deviations](http://en.wikipedia.org/wiki/Standard_deviation).



Figure1. Datasets with different correlation coefficients

We can say correlation coefficients are the measure for the linear relationship of the data. It is the normalized covariance. A dataset with larger covariance in on one direction or axis is less correlated and has more information (entropy).

* 1. **Matrix Algebra**

Matrix A:

Matrix multiplication

Outer vector product

Inner (dot) product

Length (Eucledian norm) of a vector:

=

The angle between two n-dimesional vectors:

Orthogonal:

Determinant:

Trace:

Pseudo-inverse for a non square matrix, provided is not singular

**Eigen vectors & Eigen values**

**e.g**,

**A:** mm matrix

**v:** m1 non-zero vector

**λ:** scalar

Here the (3, 2) is an eigenvector of the square matrix **A** and 4 is an eigenvalue of **A**

The vectors for a square matrix are the ones when the product of a square matrix and the vector is still in the same direction as the original vector and with different scalars.

**Calculating:**

The roots of are the eigenvalues and for each of these eigenvalues there will be an eigenvector.

**e.g.**

Then:

**=**

**=**

**=**

**=**

We get and

From

We have

=0

=

Therefore the first eigenvector is any column vector in which the two elements have equal magnitude and opposite sign.

Similar is

Property: All eigenvectors of a symmetric matrix are perpendicular to each other, no matter how many dimensions we have.

**Exercises:**

1. What is the Covariance Matrix of
2. Calculate the eigenvectors and eignevalues of
3. [**Principal Component Analysis**](http://en.wikipedia.org/wiki/Principal_component_analysis)**(PCA)**

PCA seeks a linear combination of variables such that the maximum variance is extracted from the variables. It then removes this variance and seeks a second linear combination which explains the maximum proportion of the remaining variance, and so on. This is called the principal axis method and results in [orthogonal](http://en.wikipedia.org/wiki/Orthogonality) (uncorrelated) factors.

Often, its operation can be thought of as revealing the internal structure of the data in a way that best explains the variance in the data.

If a multivariate dataset is visualized as a set of coordinates in a high-[dimensional](http://en.wikipedia.org/wiki/Dimension_(metadata)) data space (1 axis per variable), PCA can supply the user with a lower-dimensional picture, a "shadow" of this object when viewed from its (in some sense) most informative viewpoint. This is done by using only the first few principal components so that the dimensionality of the transformed data is reduced.



1. Low redundancy



1. High redundancy

Panel(a) depicts two recordings with no redundancy, i.e. they are un-correlated and with more information.

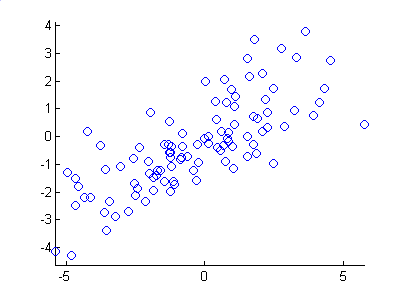
Panel(c) both recordings appear to be strongly related and linear correlated, i.e. one can be expressed in terms of the other and with less information.

**Algorithms & implement**

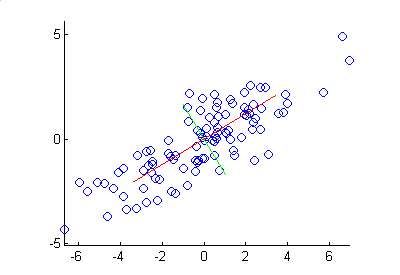
Typically we are looking for the transformation:

PX = Y

X is the original recorded data set and Y is a re-representation of that data set in the new basis matrix P. P is a matrix that transforms X into Y with the new coordinates which have the sorted variances from large to small.



 The sample dataset



The principal components of the dataset

PCA can be done by [eigenvalue decomposition](http://en.wikipedia.org/wiki/Eigendecomposition_of_a_matrix) of a data [covariance](http://en.wikipedia.org/wiki/Covariance) (or [correlation](http://en.wikipedia.org/wiki/Correlation)) matrix or [singular value decomposition](http://en.wikipedia.org/wiki/Singular_value_decomposition) of a [data matrix](http://en.wikipedia.org/wiki/Data_matrix_(multivariate_statistics)), usually after mean centering (and normalizing or using [Z-scores](http://en.wikipedia.org/wiki/Z-score)) the data matrix for each attribute.

For a original data set X, we have the covariance Matrix:

SxXXT

Where X is an m n matrix, m is data type number, i.e. the dimension, and the n is the data length of each item.

– SX is a square symmetric mm matrix.

– The diagonal terms of SX are the variance of particular measurement types.

– The off-diagonal terms of Sx are the covariance between measurement types.

We want to find the covariance matrix of Y that

1. Minimizes redundancy, unrelated between items so the information is maximized.
2. Maximizes the diagonal variances.

Where

A can be written as EDET where D is a diagonal matrix and E is a matrix of eigenvectors of A arranged as columns.

We select the matrix P to be a matrix where each

row pi is an eigenvector of XXT which is the covariance matrix.

P−1=PT since the inverse of orthonormal matrix is its transpose. The D has the goal as above.

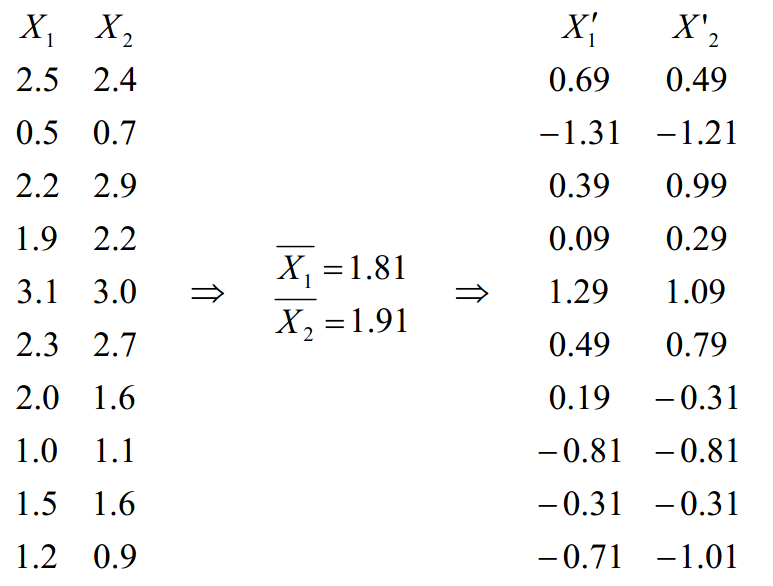
Example with steps for data PCA process and reconstruction

Step1:

Subtract the mean from each of the dimensions.

This makes the mean to zero and variance and covariance calculation easier. The variance and covariance values are not affected by the mean value.

e.g. (m=2, n=10)



Step2:

Calculate the covariance matrix

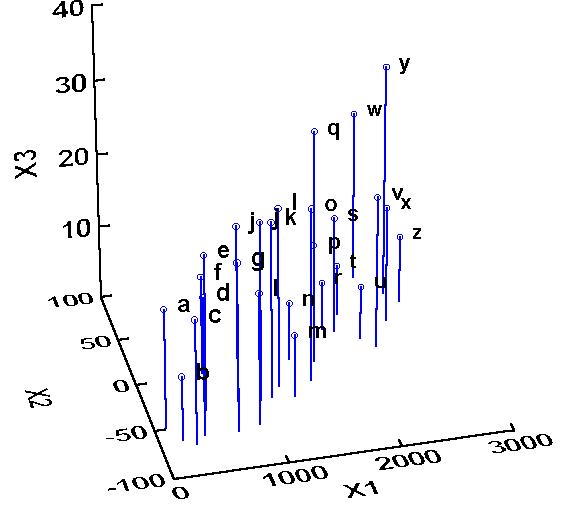
=

We can see from the covariance matrix the covariance values are large which means the items are correlated. Since it is symmetric, we will know the eigenvectors are orthogonal.

Step3:

Calculate the eigenvectors and eigenvalues of

the covariance matrix.



Sample dataset in 3 dimension space with PCA

A plot of the mean subtracted data with the eigenvectors direction lines go through the data.

We can see they are perpendicular to each other and the first one with the largest variance.

Step4:

Reduce dimensionality and form feature vector.

Order them by eigenvalue, highest to lowest. This gives the components in order of significance.

Then we can ignore the components of less significance. If a number of eigenvalues are small, we can keep most of the information we need and loss a little or noise. We can reduce the dimension of the original data.

E.g. we have data of m dimensions and we choose only the first r eigenvectors.

This is the latent in Matlab command which we usually keep the 90% of the original information.

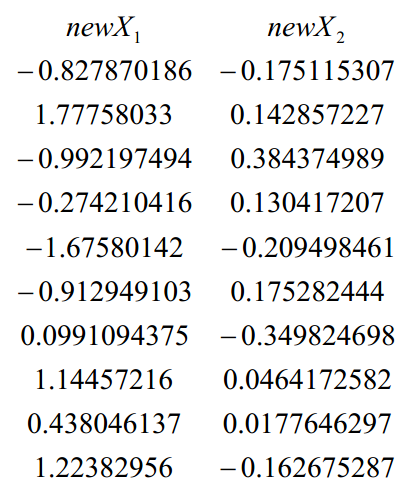
For the example before we have eigenvectors:

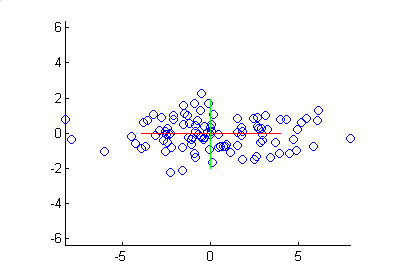
We leave out the smaller one and get:

Step5:

Derive the new data

From all the components we have new data for the example:





The principal components that are rotated

If we reduce the dimensionality, in our example let us assume that we considered only a single eigenvector.



We can see from above figure, we lost the information of the second component but keep the most significant information of the first component.

**Applications:**

**Image processing and Computer Vision**

PCA is used in computer vision. Here we give the introduction regarding the facial recognition in digital image processing.

Representation:

We see a digital image as a matrix. A square, ŒN by ŒN image can be expressed as an N2 – dimensional vector where the rows of pixels in the image are placed one after the other to form a one-dimensional image.

Pattern recognition:

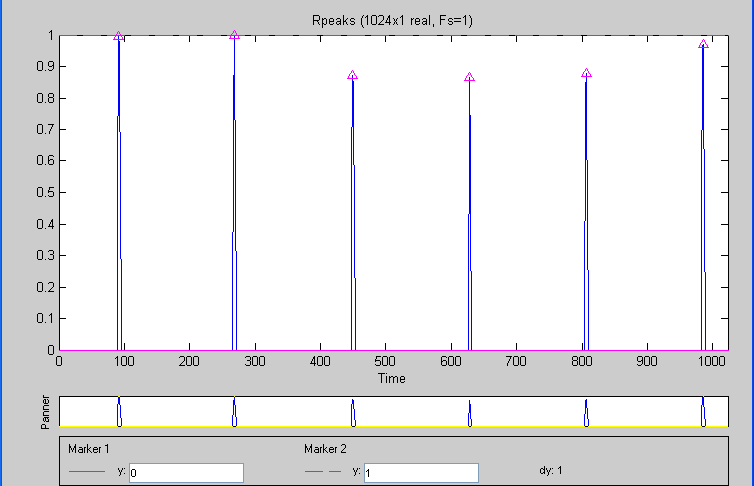
**Medical signal compression**

**12lead ECG:**





ECG signals after PCA





**Lab1**: PCA implement

**Matlab codes**:

function [signals,PC,V] = pca1(data)

% PCA1: Perform PCA using covariance.

% data - MxN matrix of input data

% (M dimensions, N trials)

% signals - MxN matrix of projected data

% PC - each column is a PC

% V - Mx1 matrix of variances

[M,N] = size(data);

% subtract off the mean for each dimension

mn = mean(data,2);

data = data - repmat(mn,1,N);

% calculate the covariance matrix

covariance = 1 / (N-1) \* data \* data’;

% find the eigenvectors and eigenvalues

http://www.mathworks.com12

[PC, V] = eig(covariance);

% extract diagonal of matrix as vector

V = diag(V);

% sort the variances in decreasing order

[junk, rindices] = sort(-1\*V);

V = V(rindices);

PC = PC(:,rindices);

% project the original data set

signals = PC’ \* data;

This second version follows section computing PCA through SVD.

function [signals,PC,V] = pca2(data)

% PCA2: Perform PCA using SVD.

% data - MxN matrix of input data

% (M dimensions, N trials)

% signals - MxN matrix of projected data

% PC - each column is a PC

% V - Mx1 matrix of variances

[M,N] = size(data);

% subtract off the mean for each dimension

mn = mean(data,2);

data = data - repmat(mn,1,N);

% construct the matrix Y

Y = data’ / sqrt(N-1);

% SVD does it all

[u,S,PC] = svd(Y);

% calculate the variances

S = diag(S);

V = S .\* S;

% project the original data

signals = PC’ \* data;

**lab2: ECG signal PCA processing**

**matlab code:**

%%pca data

for i=1:32

s=['[coeff,score,latent]=princomp(data',num2str(i),''')'];

eval(s);

datade=12;

datalen=10000;

figure

for a=1:datade

subplot(12,1,a);plot(score(:,a));

axis([0 10000 -5 5]);title(['PCA',int2str(a),' for No.',int2str(i)])

end

%%save data

temp = ['x',num2str(i)];

s=[temp,'=score(:,1);'];

eval(s);

eval(['save ',temp,' ',temp]);

%figure;

%eval(['plot(',temp,');'])

end

**References:**

[1] J. SHLENS, "TUTORIAL ON PRINCIPAL

COMPONENT ANALYSIS," 2005

<http://www.snl.salk.edu/~shlens/pub/notes/pca.pdf>

[2] Introduction to PCA,

<http://dml.cs.byu.edu/~cgc/docs/dm/Slides/>

[3] Bell, Anthony and Sejnowski, Terry. (1997) “The

Independent Components of Natural Scenes are

Edge Filters.”Vision Research37(23), 3327-3338.

[4] Bishop, Christopher. (1996)Neural Networks for

Pattern Recognition. Clarendon, Oxford, UK.

[5] Lay, David. (2000). Linear Algebra and It’s Applications.

Addison-Wesley, New York.

[6] Mitra, Partha and Pesaran, Bijan. (1999) ”Anal-

ysis of Dynamic Brain Imaging Data.” Biophysical

Journal. 76, 691-708.

[7] Will, Todd (1999) ”Introduction to the Sin-

gular Value Decomposition” Davidson College.

[www.davidson.edu/academic/math/will/svd/index.html](http://www.davidson.edu/academic/math/will/svd/index.html)

[8]“FaceRecognition: Eigenface, ElasticMatching, and

NeuralNets”, JunZhang et al. Proceedings of the

IEEE,Vol.85,No.9,September1997.

[9]

[**http://www.grasshoppernetwork.com/Technical/Share/?q=node/156**](http://www.grasshoppernetwork.com/Technical/Share/?q=node/156)