## 1. Dimensionality Reduction

Motivation I：Data Compression

Motivation II: Visulization

Principal Component Analysis Problem Formulation

Reduce from 2-dimension to 1-dimension : Find a directon (a vector u(i)) onto which to project the data so as to minimize the projection error

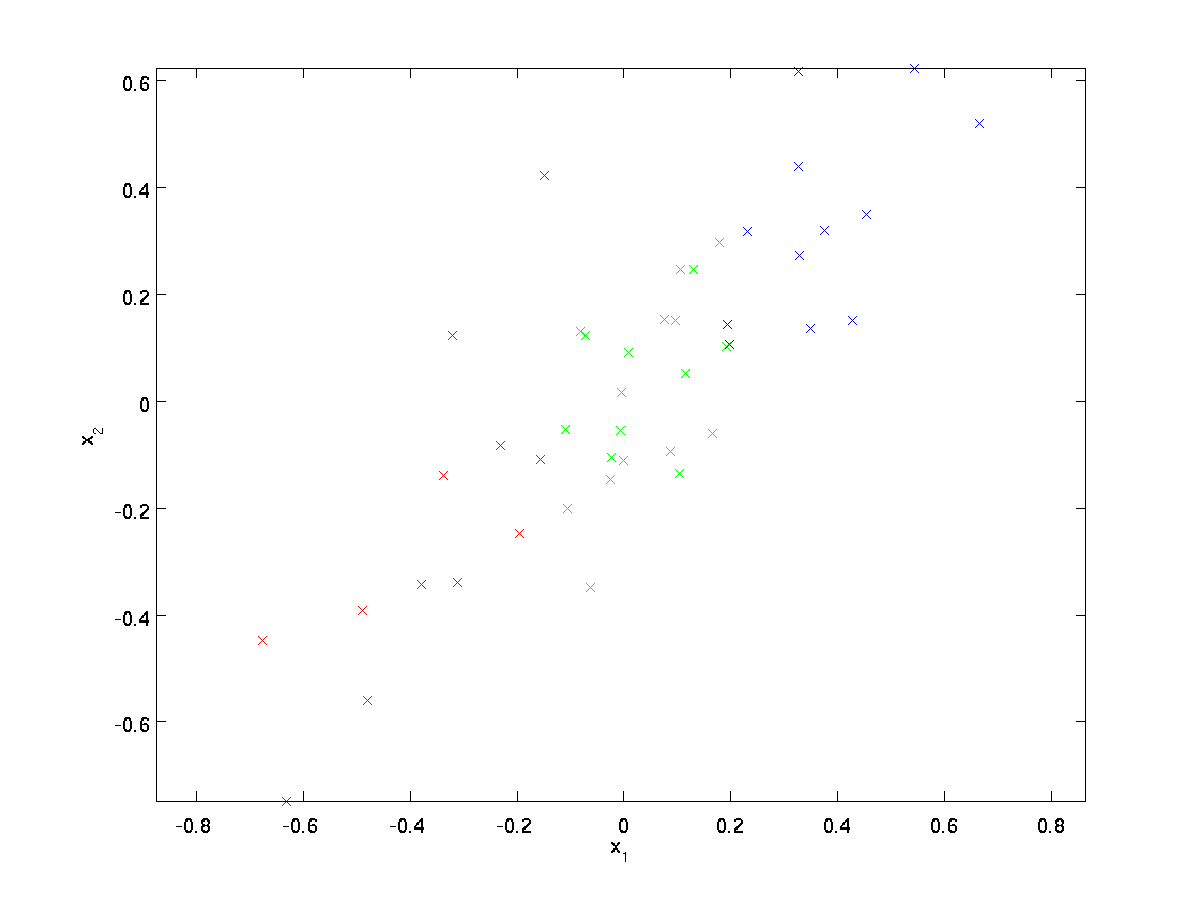
Reduce from n-dimension to k-dimension: Find k vectors u(1),u(2),...,u(k) onto which to project the data, so as to minimize the projection error.

PCA is not linear regression.

PCA aims to find the directions of maximum variance in high-dimensional data and projects it onto a new subspace with equal or fewer dimensions than the original one.

**2. Example and Mathematical Background**

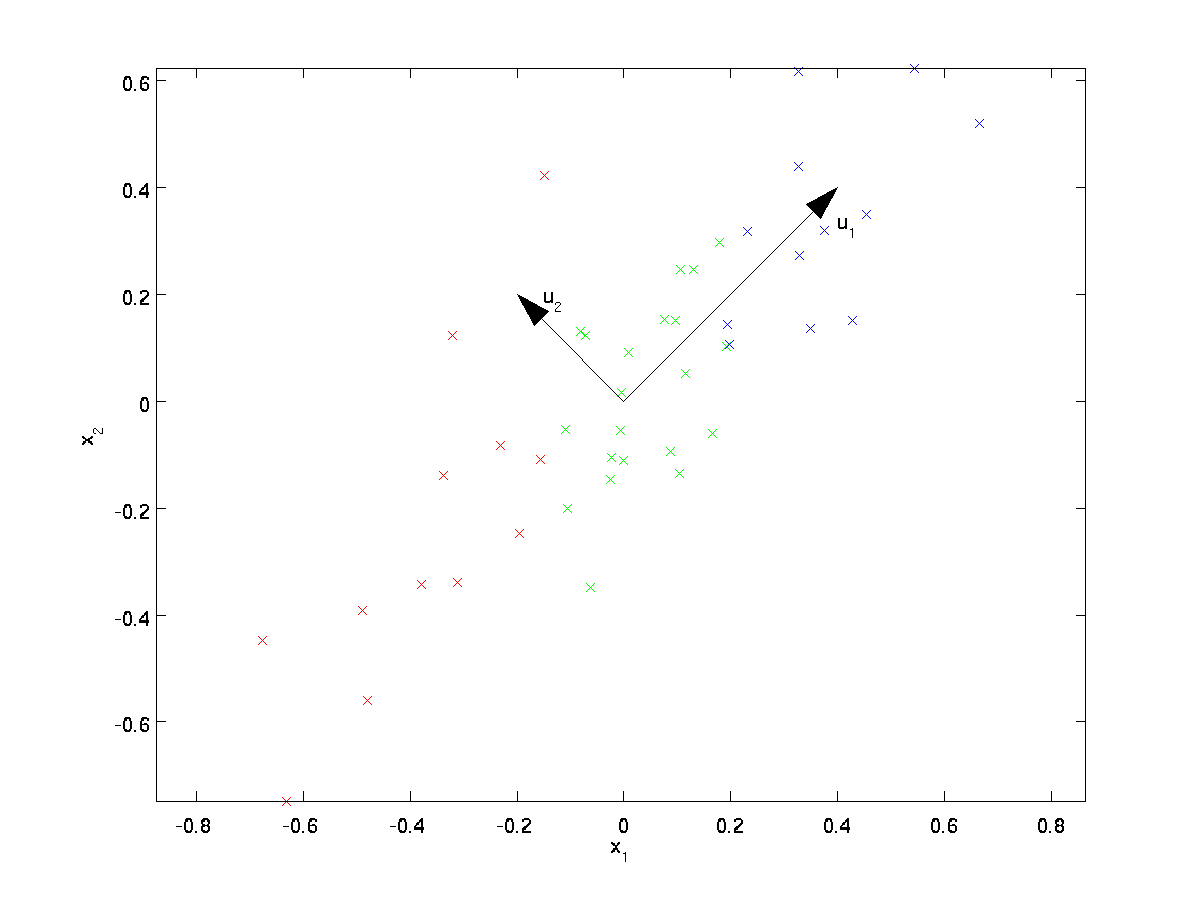
For our running example, we will use a dataset  with n=2 dimensional inputs, so that . Suppose we want to reduce the data from 2 dimensions to 1. (In practice, we might want to reduce data from 256 to 50 dimensions, say; but using lower dimensional data in our example allows us to visualize the algorithms better.) Here is our dataset:



This data has already been pre-processed so that each of the features x1 and x2 have about the same mean (zero) and variance.

For the purpose of illustration, we have also colored each of the points one of three colors, depending on their x1 value; these colors are not used by the algorithm, and are for illustration only.

PCA will find a lower-dimensional subspace onto which to project our data.  
From visually examining the data, it appears that u1 is the principal direction of variation of the data, and u2 the secondary direction of variation:



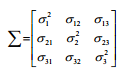
I.e., the data varies much more in the direction u1 than u2. To more formally find the directions u1 and u2, we first compute the matrix Σ as follows:



If x has zero mean, then Σ is exactly the covariance matrix of x. (The symbol ”Σ”, pronounced “Sigma”, is the standard notation for denoting the covariance matrix. Unfortunately it looks just like the summation symbol, as in ; but these are two different things.). A positive covariance between two features indicates that the features increase or decrease together, whereas a negative covariance indicates that the features vary in opposite directions

It can then be shown that u1—the principal direction of variation of the data—is the top (principal) eigenvector of Σ, and u2 is the second eigenvector.

For example, a covariance matrix of three features can then be written as (note that ∑ stands for the Greek letter *sigma*, which is not to be confused with the *sum* symbol):



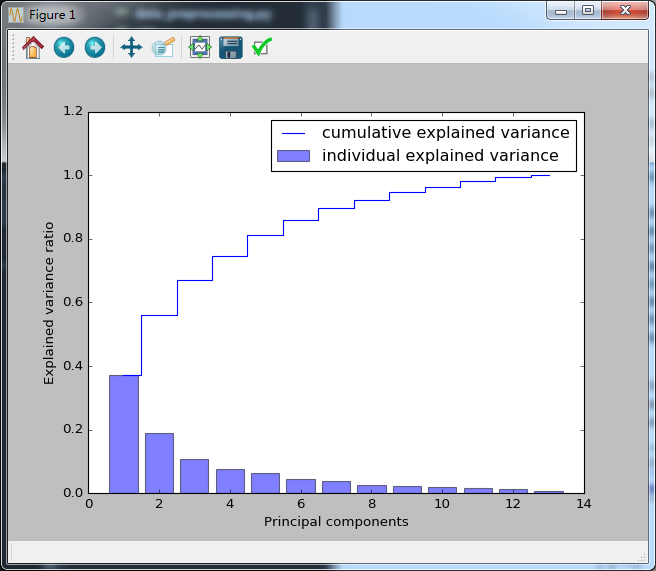
The eigenvectors of the covariance matrix represent the principal components (the directions of maximum variance), whereas the corresponding eigenvalues will define their magnitude.

eigenvalue ***v*** satisfies the following condition:



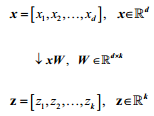
Here, λ is a scalar: the eigenvalue.

**Variance explained ratio**: 



Whereas a random forest uses the class membership information to compute the node impurities, variance measures the spread of values along a feature axis.

If we use PCA for dimensionality reduction, we construct a d x k-dimensional transformation matrix W that allows us to map a sample vector x onto a new k -dimensional feature subspace that has fewer dimensions than the original d -dimensional feature space:



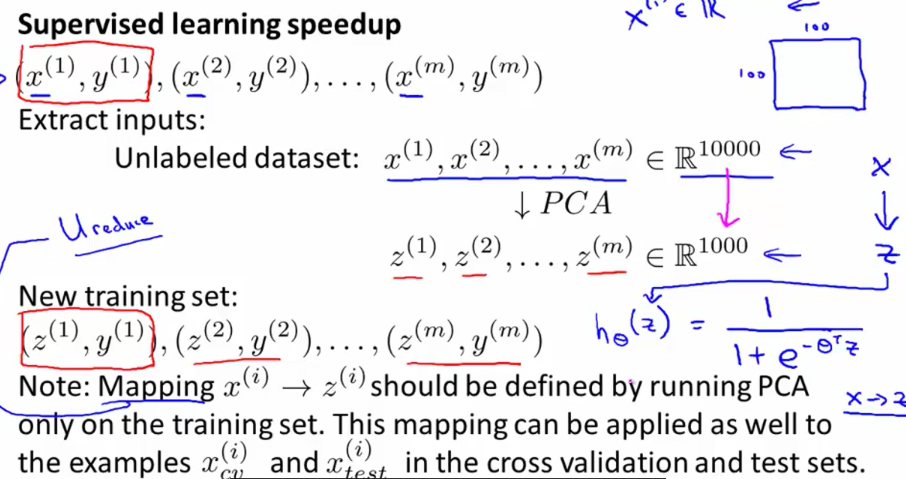
Note that the PCA directions are highly sensitive to data scaling, and we need to standardize the features *prior* to PCA if the features were measured on different scales and we want to assign equal importance to all features.

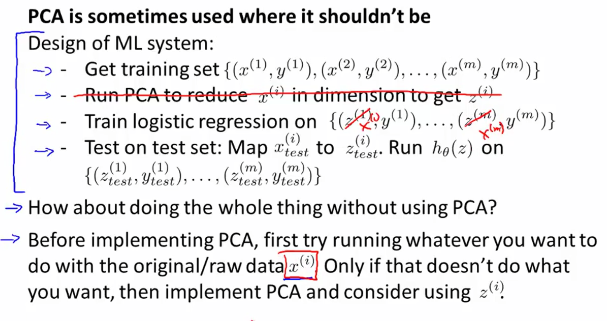
Simple steps:

1. standardize the d-dimensional dataset
2. Construct the covariance matrix
3. Decompose the covariance matrix into its eigenvectors and eigenvalues
4. Select k eigenvectors that correspond to the k largest eigenvalues, where k is the dimensionality of the new feature subspace (k<=d)
5. Construct a projection matrix W from the top k eigenvectors
6. Transform the d-dimensional input dataset X using the projection matrix W to obtain the new k-dimensional feature subspace.

## 3.Summary:

Advice for applying PCA





Reference:

1. Machine Learning, Ng Video
2. Python machine learning
3. <http://ufldl.stanford.edu/tutorial/StarterCode/>