Pattern recognition system and whitening

This lesson will draw information from the previous exercises. We will build a simple pattern recognition system (see introduction chapter in the book) and study whitening. The system begins with whitening of the data. Then we divide the data into training and validation sets (check the pdf for the 6th exercise). We don't need test set as it is included into forward search (SFS) algorithm that applies leave-one-out method. The classification of the data is done with knn (4th exercise).

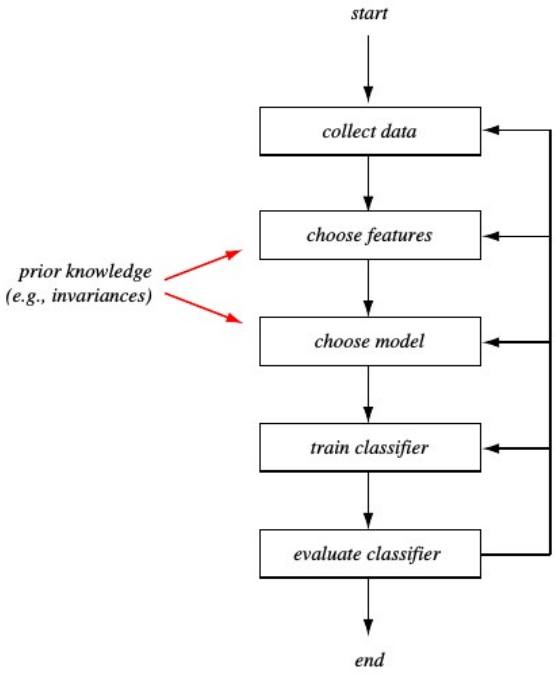


FIGURE 1.8. The design of a pattern recognition system involves a design cycle similar to the one shown here. Data must be collected, both to train and to test the system. The characteristics of the data impact both the choice of appropriate discriminating features and the choice of models for the different categories. The training process uses some or all of the data to determine the system parameters. The results of evaluation may call for repetition of various steps in this process in order to obtain satisfactory results. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Data standardization

The range of the features may vary drastically. Same data could contain data in years or dollars, from cm to astronomical units. Before anything can be discerned from the data, it needs to be preprocessed. One step of preprocessing is data normalization that can mean feature scaling as in

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

in which the data is scaled to range of [0,1]. Normalization is not always advised: If there are big outliers, then normalizing will scale the data to a very small interval. If some data parameters are known or can be calculated then data standardization is a good choice. This works especially for data that is normally distributed.

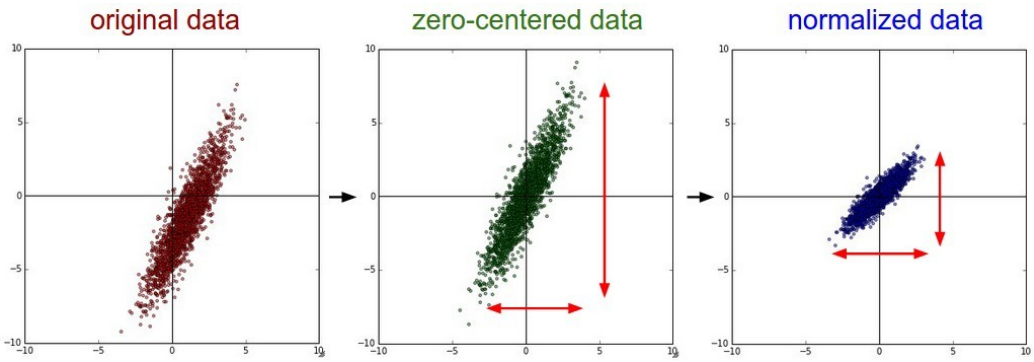
$$x_{new} = \frac{x - \mu}{\sigma}$$

The standardization sets variances to 1 but leaves correlations intact. However, this does not guarantee a range for the input values and both mean and standard deviation are sensitive to outliers.

Let's look at standardization with matrices. Let's assume we have a d-dimensional feature vector $x=(x_1,\ldots,x_d)^T$ with mean $E(x)=\mu=(\mu_1,\ldots,\mu_n)^T$ and positive definite $d\times d$ covariance matrix $var(x)=\sum$. The standardization of x can be written as

$$z=V^{-1/2}x$$

where the matrix $V=diag(\sigma_1^2,\ldots,\sigma_d^2)$ contains the variances $var(x_i)=\sigma_i^2$. It can be convenient to combine this standardization operation with mean-centering (the mean has been subtracted across all observations) of x or z to get $E(z)=0$ but it is not always necessary.



Common data preprocessing pipeline. Left: Original 2-dimensional input data. Middle: The data is zero-centered by subtracting the mean in each dimension. The data cloud is now centered around the origin. Right: Each dimension is additionally scaled by its standard deviation. The red lines indicate the extent of the data - they are of unequal length in the middle, but of equal length on the right.

Data whitening

Whitening is a more generalized version of standardization. In whitening, x is multiplied with a $d\times d$ whitening matrix W so that

$$z=Wx$$

and $cov(z)=I$. The aim is to make the input less redundant, so that in the training data the features are less correlated with each other and the features all have the same variance. When data have an identity covariance, all dimensions are statistically independent, and the variance of the data along each of the dimensions is equal to one.

Whitening is called such because it changes the input vector into a white noise vector. It is also sometimes called 'sphering' as it converts an arbitrary multivariate normal distribution into a spherical one, i.e., one having a covariance matrix proportional to the identity matrix I.

One way for determining W is with eigenvalue decomposition.

Let's say we have some data matrix X composed of K dimensions and n observations. Let's also assume that the rows of X have been mean-centered. The covariance $\sum$ of each of the dimensions with respect to the other is

$$\sum = Cov(X) = E[XX^T]$$

Where the covariance $E[XX^T]$ can be estimated from the data matrix

$$E[XX^T] = XX^T/n$$

We can write the matrix as the product of two simpler matrices E and D using a procedure known as eigenvalue decomposition

$$\sum = EDE^{-1}$$

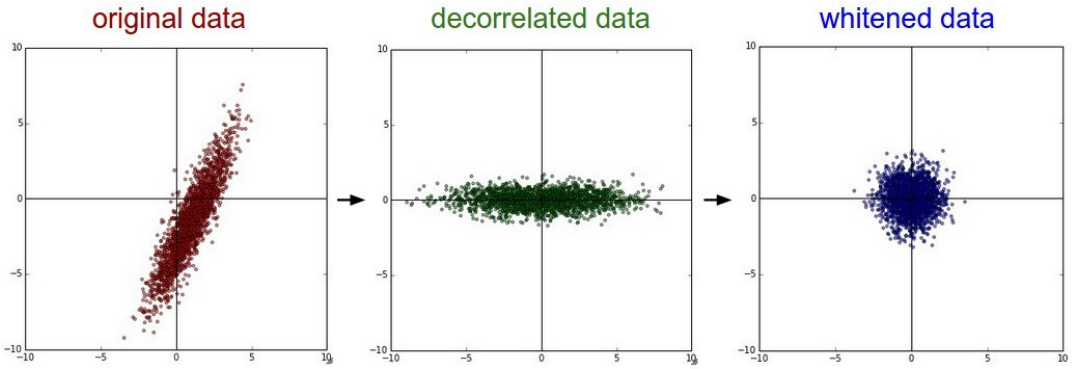
The matrix E is an K×K-sized matrix, where each column is an eigenvector of $\sum$, and D is a diagonal matrix whose diagonal elements D_{ii} are eigenvalues that correspond to the eigenvectors of the i-th column of E.

If EDE^T is the eigenvalue decomposition of the covariance matrix $\sum$ then whitening matrix

$$W_E = D^{-1/2}E^T.$$

The eigenvalue decomposition is not the only decomposition that can be used as W is not unique. E.g. SVD is often used instead as it is numerically more reliable.

(This explanation jumps over quite a bit of math. A cool link containing more information https://theclevermachine.wordpress.com/2013/03/30/the-statistical-whitening-transform/)



Whitening. Left: Original 2-dimensional input data. Middle: After performing PCA. The data is centered at zero and then rotated into the eigenbasis of the data covariance matrix. This decorrelates the data (the covariance matrix becomes diagonal). Right: Each dimension is additionally scaled by the eigenvalues, transforming the data covariance matrix into the identity matrix. Geometrically, this corresponds to stretching and squeezing the data into an isotropic Gaussian blob. (Images and image text borrowed from http://cs231n.github.io/neural-networks-2/)