Vincent Purcell

Professor Cohen

ECE 487 – Pattern Recognition

12 November 2019

Homework 7

"A Tutorial Introduction to the *Minimum Description Length* Principle"

How are MDL, the Bayesian information criterion (BIC) , and the AIC similar? How are they different?

Minimum Description Length, Bayesian Information Criterion and Akaike Information Criterion are similar because they are all based on the Maximum Likelihood function:

$$\mathcal{L}(\theta \mid x) = p_\theta(x) = P_\theta(X = x),$$

The area that the differ is their different methods of manipulating that function. The take the same approach of uses log functions to manipulate the likelihood function in order to get it into a useable form based on what is best for the input data. Those functions are as follows:

$$MDL = argmin[\,-\lg\big(P(Data|model)\big) - \lg\big(P(model)\big)]$$

$$AIC = 2k - 2\ln(\hat{L})$$

$$BIC = \ln(n)\,k - 2\ln(\hat{L})$$

How would you construct an MDL solution for problem 6.1?

The idea behind MDL and a data set like the one use in problem 6.1 is a basic curve fitting problem. The smooth curve attempts tot represent the cloud of data points and in the case of the data in problem 6.1 a hyperplane. First one would measure the closeness of the data points using the following equation,

$$D(y^n|x^n,\theta) = \textstyle\sum_i (y_i - \bar{y}_i)^2$$

Theta in the above equation represents the number of coefficients for that will be in the polynomial model. The MDL principle calls for a model, defined by parameter values, theta, that minimizes the code length of the data and that is defined by the following model:

$$L(y^n|x^n,\theta) = \log 1/f(y^n|x^n;\theta) + L(\theta)$$

What is the biggest challenge in applying MDL?

The main goal of MDL is data compression. The biggest issue arises when this data compression disagrees with the statistics behind the issue at hand. This compression will naturally lead to loss. The biggest challenge with MDL is to maximize compression while minimizing loss incurred from the model. This follows the general machine learning principal of overfitting and underfitting, as in with MDL

overfitting leads to too much loss and underfitting leads to not enough data compression to make the model worthwhile.

Textbook Problems (Computer Experiment 6.1)

Below is the published MATLAB code for Computer Experiment 6.1 from the textbook on page 401. The figure shown at the end of the document shows the original data plotted in blue and the PCA manipulation of the data plotted in red. The First principal component was found to be [-0.7054, 0.7088] which when multiplied by the original data gives the plane plotted in red. The eigenvalue for the first principal component was found to be 64.3, which means the first principal component can explain 64.3% of the variance within the data.

References

- https://homepages.cwi.nl/~pdg/ftp/mdlintro.pdf
- http://www.scholarpedia.org/article/Minimum_description_length

Contents

Vincent Purcell - HW 7 - ECE487

```
clear; clc; close all;
```

Problem 6.1

Problem 6.1 from the Text on page 401.

```
X = generate_hyper([1;1],0,10,1,1000,0);
[pc,variances]=pcacov(cov(X'))
X_0 = pc(1,:)'.*X;

%Plot subplots
figure;
subplot(2,2,[1 3]); plot(X(1,:),X(2,:),'.b'); %X
title("Data");
xlabel("X_0");
ylabel("X_1");
xlim([-10 10])
ylim([-15 15])
subplot(2,2,[2 4]); plot(X_0(1,:),X_0(2,:),'.r'); %PCA
title("PCA");
xlabel("PC1");
ylabel("PC2");
xlim([-10 10])
ylim([-15 15])
sgtitle("Principal Component Analysis");
```

Functions Received From Textbook

```
The following functions were received from the Textbook
Pattern Recognition - Theodoridis, Koutroumbas
```

Generate Hyperplane Function

Adapted from page 399 of the text

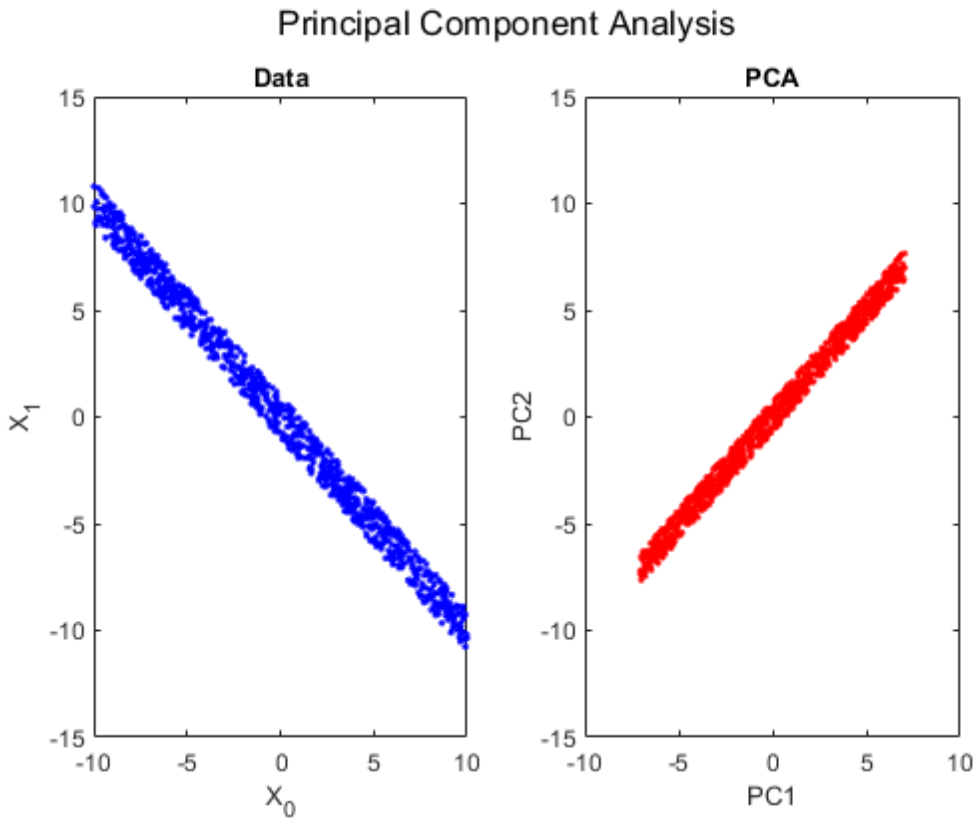
```
function X=generate_hyper(w,w0,a,e,N,sed)
    rng(sed);
    l=length(w);
    t=(rand(l-1,N)-.5)*2*a;
    t_last=-(w(1:l-1)/w(l))'*t + 2*e*(rand(1,N)-.5)-(w0/w(l));
    X=[t; t_last];
end
```

```
pc =

   -0.7054    0.7088
    0.7088    0.7054


variances =

   64.3294
    0.1624
```



Published with MATLAB® R2019b