Mathematical Statistics

SF1900 Probability Theory and Statistics: ht 2019
Computer exercise 2 for
TCOMK

Introduction

This computer exercise will at assessment either be pass or fail. The students who pass the computer exercise do not have to answer exercise 12 in part I of the written exam, but will still get the point. They will also have 3 bonus points in part II of the exam. Please note that the extra points will only be awarded at the exam at the end of the course and at the first re-exam.

Please read the instructions carefully, and make sure that you understand what the MATLAB code included does. In order to pass you have to be able to present written solutions to the preparatory exercises **individually**. For the computer exercise it is allowed (and encouraged) to work in groups of **at most two** persons. Please bring a printed copy of the instructions to the assessment. A signed copy works as a certificate that you have passed.

1 Preparatory exercises

1. When a random variable X has the density function

$$f_X(x) = \frac{x}{b^2} e^{-\frac{x^2}{2b^2}}, \qquad x \geq 0.$$

   then it is said to be Rayleigh distributed. Assume that you have observed outcomes of n independent Rayleigh distributed random variables.

   - Determine the ML estimate of the parameter b.
   - Determine the LS estimate of the parameter b.

2. Derive a confidence interval for the parameter b with approximate confidence level $1 - \alpha$. Motivate when and why it is reasonable to make the approximation.
   Hint: Base the interval on the LS estimate of b.

3. Describe the idea behind linear regression. Describe how the MATLAB command `regress` can be used to obtain estimates of the parameters in the following model

$$w_k = \log(y_k) = \beta_0 + \beta_1 x_k + \varepsilon_k \tag{1}$$

Necessary files

Start by downloading the following files

- `wave_data.mat`
- `hist_density.m`
- `birth.dat`
- `birth.txt` - description of the data `birth.dat`
- `moore.dat`

from the homepage of the course. Make sure the files are downloaded to the directory you will be working in. To make sure that the files are in the right directory you can type `ls` to list the files.
You can write your commands directly at the prompt in MATLAB, but usually it is easier to work in the editor. If the editor is not open you can open it and create a new file by typing `edit lab2.m`. The code included below is written in sections. A new section is begun by typing two percent signs. The code `Ctrl+Enter` executes the commands in a section.

Problem 1 : Simulation of confidence intervals

A confidence interval with confidence level $1-\alpha$ for the (unknown) parameter μ covers the true μ with probability $1-\alpha$. The purpose of this problem is to give an understanding of the concept of confidence level by looking at simulations. The code on ye next page uses $n = 25$ independent observations from the $N(2,1)$ distribution to compute a confidence interval for the expectation with confidence level 95% (we pretend to forget that we know what the true value is). If this is repeated 100 times leaving us with 100 confidence intervals, how many do we expect to cover the true parameter?

What do the horizontal lines and the vertical line indicate? How many intervals cover the true value of μ? Does the result agree with what you expected? Run the simulations a few more times and interpret the results.

```
%% Simulation of confidence intervals
% Parameters:
n = 25; % Number of measurements
mu = 2; % Expected value
sigma = 1; % Standard deviation
alpha = 0.05;

%Simulation of n * 100 observations. (n observations for ...
    each interval and 100 intervals)
x = normrnd(mu, sigma,n,100); %n x 100 matrix of observations

%Estimation of mu by mean
xbar = mean(x); % vector containing 100 means.

%Computation of upper and lower limits
lowerl = xbar - norminv(1-alpha/2)*sigma/sqrt(n);
upperl = xbar + norminv(1-alpha/2)*sigma/sqrt(n);

%Plot all the intervals making the ones which do not cover ...
    the true value red
figure(1)
hold on
for k=1:100
    if upperl(k) < mu
        plot([lowerl(k) upperl(k)],[k k],'r')
    elseif lowerl(k) > mu
        plot([lowerl(k) upperl(k)],[k k],'r')
    else
        plot([lowerl(k) upperl(k)],[k k],'b')
    end
end
%b1 and b2 are only used to make the figure look nice.
b1 = min(xbar - norminv(1 - alpha/2)*sigma/sqrt(n));
b2 = max(xbar + norminv(1 - alpha/2)*sigma/sqrt(n));
axis([b1 b2 0 101]) % Minimizes amount of unused space in ...
    the figure

%Plot the true value
plot([mu mu],[0 101],'g')
hold off
```

What do the horizontal lines and the vertical line indicate? How many intervals cover the true value of μ? Does the result agree with what you expected? Run the simulations a few more times and interpret the results. Now change the values for μ, σ, n and α (one at a time). How does changing the parameters affect the results?

Problem 2 - Maximum likelihood/Least squares

In this exercise you should look at to different point estimates of the parameter in a Rayleigh distribution. The code below generates observations from

a Rayleigh distribution with parameter 4 and plots the estimates `my_est_ml` and `my_est_ls`. Use the two point estimates that you derived in the preparatory exercise 1.

```
%% Problem 2: Maximum likelihood/Least squares
M = 1e4;
b = 4;
x = raylrnd(b, M, 1);
hist_density(x, 40)
hold on
my_est_ml = % Write the formula for your ML estimate here
my_est_ls = % Write the formula for your LS estimate here
plot(my_est_ml, 0, 'r*')
plot(my_est_ls, 0, 'g*')
plot(b, 0, 'ro')
hold off
```

Do your estimates look good? Check what the density function looks like by plotting it along with your estimate:

```
plot(0:0.1:6, raylpdf(0:0.1:6, my_est_ml), 'r')
hold off
```

Problem 3 - Confidence interval for Rayleigh distribution

In this section you should investigate data from a Rayleigh distributed signal; you should estimate the parameter of the distribution and determine a confidence interval for the parameter. Load the data by typing `load wave_data.mat`. The file contains a signal that you can plot by typing the following

```
%% Problem 3: Confidence interval for Rayleigh distribution
load wave_data.mat
subplot(211), plot(y(1:100))
subplot(212), hist_density(y)
```

If you change `y(1:100)` to `y(1:end)`, then you can see the whole signal. Estimate the parameter based on the observations found in `wave_data.mat` in the same way that you did in the previous problem. Assign your estimate to `my_est`. Compute a confidence interval for the parameter and assign the upper and lower limits to `upper_bound` and `lower_bound`, respectively (recall the preparatory exercise 2). Now plot the confidence interval for the parameter

```
% ...
hold on     % holds the current plot
plot(lower_bound, 0, 'g*')
plot(upper_bound, 0, 'g*')
```

Check what the density function looks like by plotting it along with your estimate, just as you did in the previous problem.

```
% ...
plot(0:0.1:6, raylpdf(0:0.1:6, my_est), 'r')
hold off
```

Does it look as if though the distribution fits the data?

The Rayleigh distribution can be used to describe the fading of a radio signal. Experimental work in Manhattan has shown that the effect of the densely built city on the propagation of a radio signal can be approximated by Rayleigh fading. [1].

Problem 4 - Comparison of distributions in different populations

In this exercise you should study a data set using visual aids in MATLAB to see if any interesting observations can be made. The file `birth.dat` contains data about 747 women who gave birth to their first child in Malmö during the years 1991-1993. The file contains 26 different variables, some of which are numerical (such as the weight and length of the mother) and some of which are categorical, i.e. they can take on one of a limited, and usually fixed number of possible values (for instance "1" if the baby was planned and "2" if the baby was not planned). Use the information in `birth.txt` and the MATLAB-functions `subplot` and `hist_density.m` to generate a figure showing histograms representing the distributions of the birth weight of the child, the age of the mother, the length of the mother, and the weight of the mother, respectively.

It is of medical interest to identify risk factors which increase the probability that a child is born with too low birth weight. Low birth weight is defined as a birth weight below 2500 g, very low birth weight as a birth weight below 1500 g, and extremely low birth weight as a birth weight below 1000 g. You should now use the given data set to try to identify risk factors for low birth weight by comparing the weight distribution of children whose mothers have a certain potential risk factor to the weight distribution of children whose mothers do not have the potential risk factor.

A known risk factor for low birth weight is if the mother smokes during pregnancy. You should therefore study the difference between birth weights of children whose mothers smoked during the pregnancy, and birth

weights of children whose mothers did not smoke during pregnancy. In the file `birth.txt` you can see that column 20 of `birth.dat` contains information about smoking habits during the pregnancy. The values 1 and 2 indicate that the mother did not smoke during the pregnancy, whereas the value 3 indicates that the mother did smoke during the pregnancy. You can therefore create two vectors `x` and `y` containing birth weights of children of non-smoking and smoking mothers, respectively, using the following code

```
%% Problem 4: Distributions of given data
    load birth.dat
    x = birth(birth(:, 20) < 3, 3);
    y = birth(birth(:, 20) == 3, 3);
```

The code `birth(:, 20) < 3` returns a vector of "true" (indicated by the value 1) and "false" (indicated by the value 0), and only those rows of column 3 (which contains the birth weights in `birth`) for which the comparison is true, end up in the vector `x`. Use the function `length` or the command `whos` to find out the sizes of the vectors `x` and `y`. Use the code below to visually present the data.

```
%% Problem 4: Distributions of given data (contd.)
    subplot(2,2,1), boxplot(x),
    axis([0 2 500 5000])
    subplot(2,2,2), boxplot(y),
    axis([0 2 500 5000])
```

```
%% Problem 4: Distributions of given data (contd.)
    subplot(2,2,3:4), ksdensity(x),
    hold on
    [fy, ty] = ksdensity(y);
    plot(ty, fy, 'r')
    hold off
```

What do the plots show? What can you say about the effect a smoking mother has on the birth weight of the child?

Now choose another of the categorical variables in the data that you suspect can affect the birth weight and use the same method as above to see if there seems to be a relationship between the birth weight and your selected variable. Note that if you choose a categorical variable which can take on more than two values you have to redefine it a bit so that it can only end up in the values 0 or 1 (as for the variable representing smoking habits above).

Problem 5 - Testing for normality

Many statistical methods are based on the assumption that the data come from a normal distribution. It is therefore of interest to be able to test for normality, i.e. to determine if a data set is well-modeled by a normal distribution. Two methods to assess normality of a data set visually are implemented in the MATLAB commands `normplot` and`qqplot` which both compare the empirical quantiles of the data set with the quantiles of a normal distribution. Using one of these commands study if the variables for the birth weight of a child, the age of the mother, the length of the mother, and/or the weight of the mother can be said to be normally distributed, and if not in which way the distribution of the variable(s) deviates from the normal distribution.

The methods `normplot` and `qqplot` rely on visual assessment of a graph and are therefore subjective to some extent. There are however statistical test designed to assess normality. One such test is the Jarque-Bera test for normality which is based on a comparison of the skewness and kurtosis of the sample data to the skewness and kurtosis of a normal distribution. For a random variable X with expectation μ and standard deviation σ the skewness, γ, and kurtosis, κ, are defined as

$$\gamma = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] \quad \text{and} \quad \kappa = E\left[\left(\frac{X-\mu}{\sigma}\right)^4\right].$$

The Jarque-Bera test for normality uses the test statistic

$$JB = \frac{n}{6}\left(S^2 + \frac{1}{4}(K-3)^2\right),$$

where n is the number of observations and S and K are estimates of skewness and kurtosis. If the data comes from a normal distribution, the JB statistic asymptotically has a chi-squared distribution with two degrees of freedom. The null hypothesis is a joint hypothesis of the skewness being zero and the excess kurtosis being zero.

Use the MATLAB-function `jbtest` to determine whether or not the variables for the birth weight of the child, the age of the mother, the length of the mother and/or the weight of the mother are normally distributed according to the test at significance level 5%.

Problem 6 - Simple linear regression

Linear regression was developed in the late 18th century by the young Gauss. The method gained attention when it was used to successfully predict the orbit of the first discovered asteroid, Ceres. Linear regression is today in extensive use with applications in virtually all sciences dealing with data. The theory is treated in more detail in the course “Regression Analysis”.

In this problem you should look at the phenomenon known as Moore's law. Load the data from `moore.mat` in the same way as before. In the data `y` represents the number of transistors/unit area whereas `x` is the year. This means that if you plot `y` against `x` then what you see is a plot of the development over time of the number of transistors/unit area. It would seem as if though the number of transistors/unit area increases exponentially, so the logarithm of the number of transistors/unit area should increase linearly over time. Let us introduce the following model

$$w_i = \log(y_i) = \beta_0 + \beta_1 x_i + \varepsilon_i, \tag{2}$$

Form a matrix X with columns

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix},$$

where n is the number of observations in the data and estimate the parameter $\beta = (\beta_0, \beta_1)^T$ using MATLAB's function `regress`. Call the estimate $\hat{\beta}$. Plot your estimated model

$$\log(\hat{y}) = X\hat{\beta},$$

by comparing $\hat{y}$ to the data y. Then plot the residuals in the following way.

```
%% Problem 6: Regression
    res =  X*beta_hat - y1;
    subplot(2,1,1), normplot(res)
    subplot(2,1,2), hist(res)
```

Which distribution do they seem to come from? Use the function `regress` to determine the quantity R^2, which is a measure of the proportion of the variance in the dependent variable that is predictable from the independent variable. If you use the data from 1972 until 2019 to estimate the parameter $\hat{\beta}$, what is your prediction for the number of transistors the year 2025?

Problem 7 - Multiple linear regression

Regression can also be used when the response variable is assumed to depend on several explanatory variables, say x, w and z. Let us introduce the following multiple linear regression model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 w_i + \beta_3 z_i + \varepsilon_i.$$

Use the data in `birth.dat` to first set up a simple linear regression model for the dependence of a child's birth weight on the length of the mother.

Then set up a multiple linear regression model where you use the weight of the mother, the smoking habits of the mother, and the other categorical variable you studied in Problem 4, as explanatory variables. Note that the categorical variables should only take on the values 0 or 1. Use `regress` to determine confidence intervals for the parameters in the multiple regression model, and use the intervals to determine whether the studied variables seem to have a significant impact on the birth weight of a child. Finally, plot the residuals of the multiple regression model using `normplot` and interpret the result.

Referenser

[1] Chizhik, Dmitry and Ling, Jonathan and Wolniansky, Peter W and Valenzuela, Reinaldo A and Costa, Nelson and Huber, Kris (2003). Multiple-input-multiple-output measurements and modeling in Manhattan *Selected Areas in Communications, IEEE Journal on*, Vol **21**, p. 321-331.