Probability Theory, ST701A, advanced level, 7.5 ECTS credits

Computer exercise 1

Assessment

This assignment is a compulsory part of the course. At the end of the session each group's/individual results will be reviewed as **pass** or **fail**. Observe that in order to be able to finish the assignment in the specified time, you are supposed to read the whole assignment and to perform the preparatory exercises **before** you attend the computer session.

1 Exploring Probability Distributions

In this exercise, you will first be given an introduction to Matlab, which is an integrated technical computing environment. Matlab will be used in the computer exercises throughout this course. We then proceed by exploring the concepts of probability distributions by means of numerical examples in Matlab.

Special instruction for ST701A. To make files and data available on your computer enter the following commands:

- Go to `System start, Statistical programs`, and choose Matlab.
- Go to `Current directory`, and click on the top-right corner, □. You get a dialog window, `Select a directory`.
- Choose `Den här datorn` and then `Inluppgifter på Studentserver statistik (M:)`
- Go to `casberlab` directory.

If you are doing the exercise at another place, all necessary files are downloadable from `Studentserver statistik (M:)`.

1.1 Preparatory exercises

As a preparation the laboration you need to read the instructions for the computer exercise, Sections 2.1, 3.2-3.3 4.5-4.6 in the course book by Casella, Berger (CB) and present answers to the following questions

1. Make sure you understand what probability mass and density functions are and how they are related to the distribution function.
2. Define the *empirical* cumulative distribution function for a sample $x_1, ..., x_n$ from a random variable (r.v.) X.
3. Write down definitions of covariance and correlation.
4. Give the definition of independent random variables. Write down the definitions of the conditional and marginal pmf and pdf's, conditional expectation and variance.

5. Determine the distribution of $Y = \sigma \cdot Z + \mu$ if $Z \in \mathcal{N}(0,1)$ and μ and $\sigma > 0$ are any constants.

6. Let (X, Y) be a two-dimensional normally distributed random variable with $\mu_X = 1$, $\mu_Y = 2$, $\sigma_X = 1$, $\sigma_Y = 0.5$ and $\rho = 0.3$. Determine the distribution of X given that $Y = 1$. (*Hint*: Recall the Exercise 4.45 from CB and explanations on p. 177).

7. Define α -quantile of a distribution.

8. Check proof of Thm 2.1.10 and make sure that you understand what is probability integral transform (see e.g Example 5.6.3) and how the inverse transform sampling can be used to generate observations from a population with continuous cdf $F_X(\cdot)$.

9. Recall the Exercise 4.26 from CB and explain what is meant by the censored random variables.

1.2 Matlab - the first steps

Skip this subsection if you are familiar with Matlab. Matlab allows the user to combine numeric computation with advanced graphics and visualisation. Short commands can be executed interactively, but for more complicated problems, it is also possible to perform programming, defining own functions ets. In addition to Matlab, several so-called toolboxes exist for specific applications, like signal processing, control theory, finite-element methods. In the computer exercises, we will make use of, among others, the commercial Statistics Toolbox. More information about Matlab is found at `http://www.mathwork.com/`.

1.2.1 Use of matrices and vectors in Matlab

Matlab can be used as an advanced calculator; the most common functions are predefined. At the Matlab prompt (»), you can for example calculate $\sin(\pi/4) + \cos^2(\pi/3) + \sqrt{3.5^3 + 4} + e^5$ by typing

```
>> sin(pi/4)*(cos(pi/3))^2+sqrt(3.5^3+4)+exp(5)
```

and the result appears on the screen. When you want to find out more about predefined functions in Matlab, the help-command `help` is useful. It is a good rule to make use of it during the exercise, even if not explicitly stated in the text! First, write `help help`. As an example, write `help log` to find out which base Matlab is using as default in the logarithm function. Matlab is shorthand for **Mat**rix **lab**oratory, and use of vectors and matrices is characteristic for Matlab. All data are stored in vectors or matrices. (With a vector, we mean a row or column matrix.) The matrix

$$M = \begin{pmatrix} 5 & 0 \\ 7 & 3 \end{pmatrix}$$

is entered in the following way:

```
>> M=[5 0; 7 3]
```

An example of a vector is given by

```
>> v=[0 1 2 3 4 5 6];
```

(A semicolon after a written statement prevents the echo on the screen, and may be useful if long vectors are entered.) We show now how to build up vectors in a simple way: the `v` can also be defined by typing

```
>> v=0:1:6
```

The command `length` or (`size`) determines the size of the vector (or a matrix):

```
>> vLength=length(v)
>> MSize=size(M)
```

It goes without saying that accessing elements in a vector is a very important step. Assume that you want the value of the second element in the vector, as well as the values of the last three elements. The solution is given by

```
>> v(2), v(4:6)
```

or, jointly,

```
>> v([2 4:6])
```

Observe that statements can follow consecutively, separated by commas or semicolons. Elements in a vector can be sorted in increasing order by the command `sort`.

```
>> u=[8 -3 2.5]; uSort=sort(u)
```

1.2.2 Managing data and variables

We have defined a number of variables, and a list of current variables is given by writing `who`. The command `whos` is similar, but also returns the size of the variables. Try yourself these commands; do you recognize the names of the variables? One can remove all variables by typing `clear`. Type `help clear` to find out how to only remove specific variables.
If a worksheet in Excel is saved on disk as a tab separated file, one can import it to the Matlab workspace: For example, a data sheet, stored as Data1.txt can be read into Matlab by typing `load Data1.txt-ascii` at the Matlab prompt. The data in `Data1.txt` will then be loaded, and afterwards, in Matlab's workspace, the data is referred to as `Data1` , i.e. the file name without its extension. This works only on condition that, firstly, the file `Data1.txt` contains only numerical values, and, secondly, all rows have the same number of elements, and, thirdly, the decimal sign is a full stop (not a comma). If you run Matlab 7.9.0, you can use the much more general and convenient command `xlsread`; type `help fileformats` for more information. If you want, you can open Excel, enter up some data, save it (see that you get appropriate file format), and try to load it into the workspace of Matlab.

1.2.3 Data visualisation and graphics

Here we will investigate how to make simple plots in Matlab. The main goal is to learn how to make a plot of a function $f(x)$, i.e $x \to f(x)$. As a first example, let us consider $y = f(x) = \cos(x), 0 < x < 4\pi$. To make a plot we need first to define vectors x and y. Let's do it as follows

```
>> x=[0:0.05:4*pi]; y=cos(x);
```

Use command `length` which we explore previously, to determine the length of x and y. Two vectors of the same length can be plotted against each other, i.e. x vs y using the following command

```
>> plot(x,y)
```

A graphical window appears and the correspondent figure is referred to as `figure1`. It is possible have a number of graphical windows accessible simultaneously. The command `plot` can be given several options, for example colour. Try to type

```
>> plot(x,y,'g')
```

Another option is the plot symbol. In order to plot the values marked as stars, we just recall the vectors x and y are composed of a number of discrete points, and type

```
>> plot(x,y,'*')
```

It is possible to combine plot options. In `help plot` you can find out what the following command will perform and check it yourself:

```
>> plot(x,y,'--rs')
```

The `axis` command may be used to display interesting regions in a figure; try for example

```
>> axis([0 7 -2.1 2.1])
```

A plot is most often easier to study if a *grid* is inserted: try to find out how to use the command `grid` and then apply a grid to the current plot.
The current figure can deleted by the command `clf`. An empty window will remain on the screen after this operation. If you also wish the window to disappear, then use the command `close` instead.

2 Summarizing data: Methods based on empirical distributions

In this part, we will use numerical examples in Matlab to approach the concept *probability* and *distribution* with the goal to complement theoretical probabilistic reasoning by an intuitive understanding.

2.1 Exploring data. Empirical cumulative distribution functions and probability plots

In order to illustrate various distributional concepts we will use the artificial data which are simulated from a number of probability distributions. This in fact is opposite to the real world situation, where no labels or explanations of the distributional properties are found. However, knowing the origin of the data makes it easier to perform data analyses. Note that simulated data are also extensively used in real life research in order to investigate properties and performance of statistical and data analysis methods.
To generate a random data set of 100 values, type

```
>> data=randn(1,100);
```

What is the distribution of your random sample (use `help randn`)? Determine the density function of this distribution.

Answer:

A good rule, whenever a new set of data is received: try to plot it in some type of graph! Use for example the plot command: `plot(data,'.-')`. Another way of presenting the data is to plot the *sorted* data: `plot(sort(data),1:length(data),'-.')`. From the data set which we generate above, choose a relatively high number, say $x = 1.7$. It could be interesting to specify the percentage of data which have values less than or equal to this number. When the size of the generated sample increases, we may interpret the percentage as the probability of observing a value less than x. The percentage is calculated in the following way:

```
>> x=1.7; percent=sum(data<=x)/length(data)
```

Check that you understand the command and try some other values of x. How do you expect the percentage to change if you decrease/increase the value of x? Compare your results with the sorted data graph.

Answer:

The inverse procedure, that is, specify the value of x which corresponds to a given percentage/probability is also quite important. This is referred to as specifying the *quantiles* of a distribution. We will investigate this in more details later on.

The similar plot can be obtained by the routine `ecdf` which computes **e**mpirical **c**umulative **d**istribution **f**unction (ecdf) of the data. This function returns two vectors, the values chosen, collected in x and corresponding percentages collected in `percent`. To see the code one can use the command `type ecdf`. The procedure simply creates a step function with a jump in cumulative probability, P, of $1/n$ at each data point, x. The result of applying `ecdf` to the set of data `data` is depicted in a new figure after performing the following steps:

```
>> [percent,x]=ecdf(data);
>> figure(2);
>> subplot(2,1,1)
>> plot(x,percent,'.')
>> grid on
>> subplot(2,1,2)
>> ecdfhist(percent,x,10)
>> colormap([0.5 0.5 0.5])
```

The upper panel of figure 2 must be similar to the figure 1 obtained before and shows how the values in data set are distributed data presentation we, for each value of $x = x_0$, can specify a percentage of values in the sample with values less than x_0. The lower panel of the figure 2 represents **e**mpirical **c**umulative **d**istribution unction **hist**ogram. This procedure computes the bar heights in the empirical cdf, and normalizes them so that the area of the histogram is equal to 1, unlike the `hist` procedure where bars heights represent bin counts. Different number of bins (in this case 10) can be specified in teh last agrument of the function.

2.2 Large samples. Distribution function of a random variable

We now investigate a larger set of data, let say 2500 observations coming from the same distribution as the previous data. We simulate data and plot them in a new figure:

```
>> data=randn(1,2500);
>> [percent,x]=empcdf(data);
>> figure(3);
>> plot(x,percent,'.')
>> grid on
```

Given a large number of observations, the resulting empirical distribution approaches the true distribution function. In our example, the data are generated from the normal distribution, i.e. $X \in \mathcal{N}(0,1)$. It is interesting to plot the theoretical distribution function, specified by `normcdf` together with the empirical one. To get both distributions in the same figure perform the following steps:

```
>> figure(3);
>> hold on
>> plot(x,normcdf(x),'r')
>> hold off
```

Interpret the figure. What are on the x and y axes? How the number of observations effects the graph properties? Try to use `ecdfhist` and superimpose the theoretical density function on the normalized histogram by the following commands.

```
>> figure(4);
>> ecdfhist(percent,x,10)
>> colormap([0.5 0.5 0.5])
>> hold on
>> plot(x,normpdf(x), 'r-','LineWidth',2)
>> hold off
```

Answer:

2.3 Quantiles and Q-Q plots

The concept of quantile is quite important in both probability and statistics and can be defined in different ways. Here we will use the following definition: If X is a continuous random variable with a strictly increasing distribution function, $F_X(\cdot)$, the α-quantile of the distribution is defined to be the solution of the equation

$$P(X \leq x_\alpha) = 1 - \alpha, \quad \text{or} \quad P(X \geq x_\alpha) = \alpha.$$

Special cases are $\alpha = 0.5$ which corresponds to the *median* of $F_X(\cdot)$ and $\alpha = 3/4$ and $\alpha = 1/4$, which corresponds to the lower and upper quartiles of $F_X(\cdot)$. Use your graph of the empirical distribution from the exercise above and definition of quantile, try to estimate different quantiles. Start by $\alpha = 0.5$ and estimate the median of x_α. Proceed by the following steps

```
>> x = [normrnd(4,1,1,100) normrnd(6,0.5,1,200)];
>> q = [0:0.25:1]
>> y = quantile(x,q)
```

Compare your results with the exact values given by

```
>> norminv(1-alpha)
```

Answer:

Quantile-Quantile (Q-Q) plots are very useful graphical tools for comparing distribution functions. The purpose of the quantile-quantile plot is to determine whether the sample in x is drawn from a specific distribution (i.e one can qualitatively assess the fit of the data to a theoretical distribution), or whether the samples in x and y come from the same distribution. If the samples do come from the same distribution (same shape), even if one distribution is shifted and re-scaled from the other (different location and scale parameters), the plot will be linear. Suppose that $F_X(\cdot)$ is $\mathcal{N}(0,1)$ and $G_Y(\cdot)$ is $\mathcal{N}(1,1)$. Simulate 1000 observations from these distributions and sketch a Q-Q plot using

```
>> qqplot(X,Y)
```

Repeat for $G(\cdot)$ being $\mathcal{N}(1,4)$. Make also a Q-Q plot when $F(\cdot)$ and $G(\cdot)$ are exponential distribution with parameters $\lambda = 1$ and $\lambda = 2$, respectively. Explain your results.

2.4 Some other probability distributions

Some commonly used distributions and distribution functions have their own names. They are not only functional expressions in a mathematical sense but turn out to be appropriate analytical tools for modelling various real life random phenomena. Furthermore, many named distribution are special cases of the more common distributions. Many of the distributions (and relationships between them) are listed in CB and almost all of them are implemented in the Statistics Toolbox.

You are already familiar with normal distribution. Another important distribution which you meet throughout the course is the *exponential distribution* whose pdf is

$$f(x|\mu) = \frac{1}{\mu}e^{-x/\mu}, \quad 0 < x < \infty$$

The exponential distribution is a special case of the gamma distribution (obtained by setting $\alpha = 1$) in

$$f(x|\alpha, \beta) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}}x^{\alpha-1}e^{-x/\beta}, \quad 0 < x < \infty, \alpha > 0, \beta > 0,$$

where $\Gamma(\cdot)$ is the Gamma function.
The exponential distribution is especially important in modeling events that occur randomly over time. Its main application area is in studies of lifetimes. The exponential distribution can be used to model the length of life of an object, for example the lifetime of a light bulb or a patient given a particular treatment. The exponential distribution has an interesting property, known as the memorylessproperty. For $X \sim Exp(\mu)$ and for $s > t \geq 0$, it is the case that

$$P(X > s|X > t) = \frac{P(X > s)}{P(X > t)} = P(X > s - t) = e^{-(s-t)/\mu}.$$

See Section 3.3, CB. This means that for exponentially distributed lifetimes, the probability that an object will survive an extra unit of time is independent of the current age of the object. To exemplify this special property we perform the following steps.

```
>> x = 5:5:60;
>> xpd = x+0.1;
>> deltaF = (expcdf(xpd,40)-expcdf(x,40))./(1-expcdf(x,40))
```

Answer:

Explain commands and results.
To further investigate properties of exponential distribution we recall the Exercise 4.26 from CB where the *censored* random variables X and D are considered. We generate random failure (life) times as $X \sim Exp(10)$ and random censoring times as $D \sim Exp(20)$ by

```
>> x = exprnd(10,50,1);
>> d = exprnd(20,50,1);
```

Now as in the Exercise 4.26 we assume that instead of directly observing X and D we observe the random variable $Z = \min(X, D)$, i.e minimum of these times

```
>> t = min(x,d);
>> censored = (x>d);
```

By the last command we control for whether the subject failed. The resulting vector `censored` has the same size as x and its elements are 1 for observations that are right-censored and 0 for observations that are observed exactly. Now we can construct and plot empirical cumulative distribution function which takes into account censoring, and compare the empirical cdf with the known true cdf. We also superimpose a plot of the known population distribution function.

```
>> [F,y] = ecdf(t,'censoring',censored);
>> stairs(y,F,'LineWidth',2)
>> hold on
>> xx = 0:.1:max(t);
>> yy = 1-exp(-xx/10);
>> plot(xx,yy,'g-','LineWidth',2)
```

By the last steps we superimpose a plot of the known population distribution function. Notice that unlike default `ecdf` has an extra argument, `censoring`, which indicates that not all observations are observed exactly. Try to vary parameters of the distributions of X and D and explain your resulting graphs.
Optional: Observe that the exponential distribution is clearly not a best model for biological survival/life times. If it would be used as a model for the time until death of a biological object, it would imply that the probability of the object death did not depend on its age. This is a consequence of the memorylessproperty of the exponential distribution. In real life, however, one can expect that the probability that a 16-years-old will live at least 10 more years is not the same as the probability that an 80-years-old will live at least 10 more years. The more flexible distribution that can capture and model changing in the death rate (failure rate) is the *Weibull distribution* given by the following probability density

$$f(x|\gamma,\beta) = \frac{\gamma}{\beta}x^{\gamma-1}e^{-x^{\gamma}/\beta}, \quad 0 < y < \infty, \gamma > 0, \beta > 0.$$

A constant failure rate corresponds to the case $\gamma = 1$ which gives an exponential distribution as a special case of Weibull. An increasing failure rate means that units are more likely to fail/die as time goes on, and corresponds to the case of $\gamma > 1$. Use the previous analysis of the exponential distribution and `help wblrnd` command to investigate how to model full and censored data from the Weibull distribution. Try to use the simulated data to construct the empirical cdf and fit the distribution.

2.5 Conditional distributions

The goal of this part of the laboration to give you understanding the concept of the *conditional probability distribution*. Conditional distributions and especially conditional moments (expectation and variance) are important concepts in statistical prediction analyses. In this part of the laboration you will need two routines `normal2d` and `condnormal`. You can download it from `Studentserver statistik (M:), casberlab`. We focus on two-dimensional normally distributed random variables (X, Y) and explore the role of correlation coefficient as a measure of dependence between X and Y. The density function of (X, Y) with bivariate normal parameters μ_X, μ_Y, σ_X^2, σ_Y^2 and $\rho = \rho(X,Y) = \frac{Cov(X,Y)}{\sigma_X\sigma_Y}$ is

$$f_{X,Y}(x,y) = \mathcal{C}\exp\Big\{-\frac{1}{2(1-\rho^2)}\Big[\Big(\frac{x-\mu_X}{\sigma_X}\Big)^2 + \Big(\frac{y-\mu_Y}{\sigma_Y}\Big)^2 - 2\rho\Big(\frac{x-\mu_X}{\sigma_X}\Big)\Big(\frac{y-\mu_Y}{\sigma_Y}\Big)\Big]\Big\},$$

where the normalising constant is

$$\mathcal{C} = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}}.$$

By determining the conditional density $f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$ you can see that the conditional distribution of X given $Y = y$ is univariate normal with parameters

$$\mathrm{E}(X|Y=y) = \mu_X + \rho\frac{\sigma_X}{\sigma_Y}(y-\mu_Y),$$

$$\mathrm{Var}(X|Y=y) = \sigma_X^2(1-\rho^2).$$

Observe that the conditional mean consists of μ_X plus a correction term which is a linear function of y whereas the conditional variance depends on ρ only and decreases with increasing $|\rho|$. Later on we will relate these facts to the elliptic countours if the joint density of (X,Y). Analogous expressions can be obtained for the multivariate normal distribution, i.e for the conditional distribution of $X_1|X_2$ where $X = [X_1; X_2]' \in \mathcal{N}_n(\mu, \Sigma)$ with

$$\mu = [\mu_1; \mu_2]' \quad \text{and} \quad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$$

where X_1 and X_2 are two sub-vectors of respective dimensions p and q with $p+q=n$. Note that $\Sigma = \Sigma^T$, and $\Sigma_{21} = \Sigma_{21}^T$.
Now we investigate graphically how the conditional distribution, expectation and variance of X change when varying different parameters in the expressions above. In other words, how our information about X changes after observing that $Y=y$? To visualise effect of variation of Y we use `normal2d` och `condnormal` which return figures representing the involved distributions/densities. The command
`normal2d`$(\mu_X, \mu_Y, \sigma_X, \sigma_Y, \rho)$ produces a graph of the bivariate density function, its countur plot and marginal densities of both X and Y. The command `condnormal`$(\mu_X, \mu_Y, \sigma_X, \sigma_Y, \rho, ' y', y_0)$ generates a graph of the conditional density functions for X given $Y=y_0$. Try different distributions and investigate how the conditional expectation and variance are effected by small vs large values of ρ, σ_X and σ_Y. What do you observe when $\rho=0$ or when $\rho=0.9$?

Answer:

Use both `normal2d` and `condnormal` and `hold on` in order to investigate how varying of ρ and σ_Y effects the conditional density.

Answer:

2.6 Constructing dependent bivariate distributions

In this part, we first investigate a simulation technique for the dependent normal random variables which is derived from the starting point of bivariate normality of (X,Y). It is straightforward to show (see e.g. CB, p 177) that the conditional distribution of $X|Y=y$ and $Y|X=x$ are also normal, i.e.

$$Y|X=x \sim \mathcal{N}\Big(\mu_Y + \rho\frac{\sigma_Y}{\sigma_X}(x-\mu_X), \sigma_Y^2(1-\rho^2)\Big).$$

We have used this fact in the previous section. Thus, in order to simulate n pairs of observations from a bivariate normal distribution with parameters μ_X, μ_Y, σ_X, σ_Y and ρ we can first simulate X as for example

```
>>  n = 900; muX=0; muY=0; sigmaX=1; sigmaY=1; rho=0;
>>  x=normrnd(muX,sigmaX^2, n,1);
```

and then use the condition distribution of $Y|X = x$ to simulate Y. Perform these steps and plot the resulting observations by `plot(x,y.'*')`. Was the result as expected? Try to vary ρ and investigate what happens with the graph if the correlation is near -1 or 1. Explain it analytically, i.e. try to verify that the joint distribution of (X, Y) becomes more concentrated about the line $y = \mu_Y + \rho\frac{\sigma_Y}{\sigma_X}(x - \mu_X)$ when ρ is approaching -1 or 1.

Answer:

Now we turn to another approach for modeling bivariate distributions. Using the theory of univariate random number generation we recall that by applying the inverse cumulative distribution function of any distribution $F(\cdot)$ to a $U(0, 1)$ random variable, we get a random variable whose distribution is exactly $F(\cdot)$. This technique is known as the *Inversion method*. Similar two-step transformation can be applied to each variable of a standard bivariate normal, creating dependent random variables with arbitrary marginal distributions. Because the transformation works on each component separately, the two resulting random variables need not even have the same marginal distributions. For example, we can generate random vectors from a bivariate distribution with $\chi^2(3)$ and $t(6)$ marginals in the following way. Start from

```
>>  n = 1000; rho = 0.8;
>>  Z = mvnrnd([0 0], [1 rho; rho 1], n);
>>  U = normcdf(Z);  X = [chi2inv(U(:,1),3) tinv(U(:,2),6)];
```

Check what `mvnrnd` does and investigate functions `chi2inv` and `tinv` using `help`. Proceed by the following steps

```
>> [n1,ctr1] = hist(X(:,1),20);
>> [n2,ctr2] = hist(X(:,2),20);
>> subplot(2,2,2); plot(X(:,1),X(:,2),'.'); axis([0 12 -8 8]); h1 = gca;
>> title('1000 Simulated dependent Chi2 and t values');
>> xlabel('X1 ~ Chi2(3)'); ylabel('X2 ~ t(6)');
>> subplot(2,2,4); bar(ctr1,-n1,1); axis([0 12 -max(n1)*1.1 0]); axis('off'); h2 = gca;
>> subplot(2,2,1); barh(ctr2,-n2,1); axis([-max(n2)*1.1 0 -8 8]); axis('off'); h3 = gca;
>> set(h1,'Position',[0.35 0.35 0.55 0.55]);
>> set(h2,'Position',[.35 .1 .55 .15]);
>> set(h3,'Position',[.1 .35 .15 .55]);
>> colormap([.7 .7 1]); grid
```

and explain the resulting figures.

Answer: