Parameter estimation and linear time-series analysis

This computer exercise gives an introduction to time-series analysis and estimation and model choice within the Maximum Likelihood framework. The exercise was designed with Matlab in mind, but you are allowed to use the programming language or package of your choice (**R**, `Python`, Julia etc.).

1 Preparations for the exercise

Read chapter 4 in [1] and this instruction for the computer exercise. Also review relevant exercises in chapter 4.

Before the computer exercise some of the questions below will be posed. All of the posed question must be answered correctly in order to pass the computer exercise. Anyone that has not been preparing properly will be asked to leave the computer exercise.

2 Catalogue of questions

You should be able to answer the following questions before the computer exercise.

1. Calculate the auto-correlation function for an AR(1) process and an MA(1) process respectively.
2. Describe how one can estimate the parameters in an AR(1) process using Maximum Likelihood. How are the parameters estimated in an MA(1) process?
3. Write down the likelihood function for the following model:

$$y_i = \beta_0 + \beta_1 x_i + \sigma\varepsilon_i, \tag{1}$$

   where ε_i is iid $N(0,1)$ or $t(\nu)$.
4. Compute the estimates and the covariance of the estimates for the linear model with Gaussian noise, when using OLS and ML methods

3 Maximum Likelihood estimation

Maximum Likelihood (ML) estimation is the preferred technique if the distribution of the innovations in the model is known and the risk for misspecification is small. The basic idea is to choose the parameter values that make the observed data the most probable. The Likelihood function is given by:

$$L(\theta) = p(x_1, \ldots, x_n|\theta) = p(x_1|\theta) \prod_{i=2}^{n} p(x_i|x_{i-1}, x_{i-2} \ldots; \theta)$$

where $p(x_i|x_{i-1}, x_{i-2} \ldots; \theta)$ is the conditional distribution of x_i given all previous values of x. Choose the estimate as

$$\widehat{\theta} = \arg\max_{\theta \in \Theta} L(\theta).$$

This is difficult to optimize, as it contains a product of terms. Instead, we solve the equivalent problem

$$\widehat{\theta} = \arg\max_{\theta \in \Theta} \log L(\theta) = \arg\max_{\theta \in \Theta} \sum_{i=2}^{n} \log p(x_i|x_{i-1}, x_{i-2} \ldots; \theta)$$

Under fairly generous regularity conditions the following holds:

$$\sqrt{N}(\hat{\theta} - \theta) \xrightarrow{d} \mathcal{N}\left(0, I_F^{-1}(, \theta)\right),$$

where $I_{0,\theta}$ is Fisher's Information matrix $I_F(, \theta) = -\mathbf{E}[\frac{\partial^2}{\partial \theta^2} \log L(X, \theta)]$ This can be approximated by the empirical Hessian $[\frac{1}{N} \frac{\partial^2}{\partial \theta^2} \sum_{i=1}^{N} \log L(X_i, \theta)|_{\theta=\widehat{\theta}}]$ or the covariance of the score functions.

3.1 Tests

A very common technique that is used to determine model order and parameter significance is the Likelihood Ratio (LR) test. The test is used to compare two different models in which the parameter space of one model (the so-called restricted model) is a subspace to the parameter space of another model (the so-called unrestricted model). The tests are based on the following null-hypothesis:

$$H_0 : \theta = \theta_0$$
$$H_A : \theta \neq \theta_0.$$

The test statistic is the defined as:

$$\Lambda = \frac{\sup_{H_0} L(\theta)}{\sup_{H_0 \cup H_A} L(\theta)}.$$

Intuitively this means that if H_0 isn't true the denominator will be much larger than the numerator. One can show that if the null-hypothesis is true, it holds asymptotically that

$$-2 \log \Lambda \in \chi^2(k),$$

where k is the number of fixed parameters in the smaller model. Please note that the test can be written as

$$-2 \left(\sup_{H_0} \log L(\theta) - \sup_{H_0 \bigcup H_A} \log L(\theta) \right).$$

The test can be used when the likelihood function is known. There exists more powerful tests for some models, e.g. the F-test, but these tests are asymptotically equivalent.

4 Computer exercises

The computer exercise consists of two parts; an introduction to Maximum Likelihood estimation applied to linear regression and an introduction to time-series analysis. Please note that you are expected to write all programs yourself and *not* use any existing toolboxes for statistical modeling other than to check your results.

4.1 Linear regression

To get familiar with Maximum Likelihood estimation in a controlled environment we will apply the technique to a simple example, namely regression. Load the data in to Matlab by writing:

```
>> load regdataN.mat
```

```
>> load regdatat.mat
```

where `regdataN.mat` has Gaussian additive noise and `regdatat.mat` has Student t-distributed additive noise, effectively contaminated by outliers.

Assignment: Estimate the linear model, $y_i = \beta_0 + \beta_1 x_i + \sigma\varepsilon_i$, and construct a 95% confidence interval for the parameters for both datasets. The estimator can be calculated in closed form, but it is the purpose of this computer exercise to get familiar with numerical minimization of functions. That means that you *should* estimate the parameters by applying numerical optimization to the log-likelihood function. Compare your results to the results from LS-estimation, see for instance `mldivide` and `regress`.

Assignment: Test the linear model against a quadratic model using LR-test. It is sufficient to do this for one of the datasets.

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \sigma\varepsilon_i. \tag{2}$$

Present and comment on the result.

4.2 Time series analysis

We will limit ourselves to AR(p) processes to avoid computational difficulties. If you took the course on time series analysis this section won't bring anything new to the table. Please note that you are expected to write the routines needed for model fitting yourself. You will fit a model for a simulated interest rate process. We can not observe the interest rate directly but only the price of bonds. Therefore we must calculate the interest rate from the bond price as $p(t,T) = e^{-r(t)(T-t)}$. Start with loading:

```
>> load prisdata.mat
```

and transforming the data. Is the data stationary?

Assignment: Fit a suitable AR(p) process to data and study the distribution and autocorrelation of the residuals.

Hint: Remember that the expected value of an AR(p) process is zero.

Modeling help

The section below doesn't contain any extra assignments and is simply an aid those you who didn't take the course in time series analysis.

The original data series isn't stationary; it is clear from the expected value but even more so from the time decaying variance. A stationary series can be constructed through the transformation below.

```
>> r=-log(p(1:end-1))./(T-t(1:end-1));
>> plot(r)
```

Note that you cannot reconstruct the series on the day of maturity since the price is 1 (any other price would have given an arbitrage opportunity). The interest rate process you have reconstructed has constant expected value and variance and hence we can consider fitting a suitable model. This will be done by first finding a suitable number of lags in the AR(p) process by observing the autocorrelation process.

```
>> rm=r-mean(r);
>> sacfplot(rm,30);
```

There is significant dependence on the first lag. Therefore, fit an AR(1) process and study its residuals. If you have written your routines correctly you will find that there is still a significant dependence among the residuals. *Test higher order processes and compare the models with the LR-test.*

You can cheat by using Matlabs built-in routines for time series models. Compare your own results with Matlabs by writing:

```
>> th1=arx(rm,1);
>> present(th1)
>> th2=arx(rm,2);
>> present(th2)
```

5 Feedback

Comments and ideas relating to the computer exercise are always welcome. Send them to Magnus wiktorsson, `magnus.wiktorsson@matstat.lu.se`

6 MATLAB–routines

`arx` Fits an ARX process to data.

`present` Presents the results from a model fitting

`resid` Calculates and test the residuals from a model.

`fminunc` Numerical minimisation of a multidimensional function. The routine is based on quasi-Newton (BFGS) and it can return the minimizing parameter value, the minimal function value and the Hessian.

`MLmax` Custom written Quasi-Newton with line search optimisation algorithm based on the BHHH algorithm specifically developed for maximum likelihood estimation. It maximises the log-likelihood function by estimating the Fisher's Information matrix from the sample covariance of the score function, which ensures that the matrix is positive definite. The log-likelihood function must be returned as a column vector in order to be able to compute the average over the samples $I_F \approx \frac{1}{N}\sum_{n=1}^{N}(\nabla_\theta \log p)_n(\nabla_\theta \log p)_n^T$.

```
>> [xout,logL,CovM]=MLmax(@lnL,x0,indata)
```

References

[1] Lindström, E., Madsen, H., Nielsen, J. N., (2015) *Statistics for Finance*, Chapman and Hall/CRC, ISBN 9781482228991.