

Lab 2: Filters, Autocorrelation, Tests for randomness

Due: Sep 23.

1 Introduction

In this lab you will explore the MA and AR filters, and examine their auto-correlation functions. You will also compare various tests for randomness. In class we talked about using the acf of a residual process, rather than the acf of the process, to determine whether we've selected a model of adequate complexity. A model may be adequate if the residual process is unpredictable or random. Finally, in the last section of the lab you will implement a forecasting algorithm, and examine the partial auto-correlation function (pacf) of an AR process.

2 White noise

Simulate two white noise processes with a) normally distributed observations, and b) exponentially distributed observations. Adjust the simulated data such that the mean and variance equal 0 and 1 for both processes. Plot the simulated data and the autocorrelation functions. What do you observe? How many observations exceed ± 2 for each set of simulations, how many would you expect to exceed these limits?

Simulate the product series $x_t = e_t e_{t-1}$. Plot the simulated data and acf. Also display a qq-plot of the x_t process (`qqnorm`, `qqline`). Does the acf plot test for independence? Plot the acf of the squared product series, and the acf of the squared e_t process. Discuss.

3 MA

Simulate a normal white noise processes with mean 0, but different variances. Use these noise processes to simulate an MA(2) process using the filter command.

```
w1<-rnorm(502)
a<-c(1,-.5,.2)
ma2<-filter(w1,filter=a,side=1)
ma2<-ma2[3:502]
```

Look at the help file for `filter`. Change the sign of the second filter coefficient to .5. Plot the acf of the simulated MA(2) process (`help(acf)`, `type=`). Simulate a new MA(2) process using the second noise process. Plot the acf and the autocovariance function (`acvf`). Confirm the theoretical acf and `acvf` of an MA(2) using the plots. Try other coefficient values, and higher order MA processes. Discuss.

Simulate an MA(q) process. This time, use a noise process that is not stationary. For example, you can simulate a noise process with increasing variance. Is the resulting MA process stationary? Plot the MA process and its acf. Does the acf plot test for stationarity?

4 AR

Use the `filter` command to simulate an AR(2) process. First simulate 1500 observations from a standard normal, this is your noise process `w`. Simulate an AR(2) process as follows;

```
a<-c(a1,a2)
```

```
ar2<-filter(w, filter=a, side=1, method='recursive')
```

The AR process requires an initialization, so you should use only the last 250 values for further study. Plot the AR(2) process (last 250 values) and its acf. What do you see? Now simulate an AR(1) process with small and large, positive and negative coefficient values. What do you see? Simulate more AR(2) processes and discuss for which choices for `a1` and `a2` the simulated process is causal and stationary.

5 Estimating AR-coefficients by regression

A simple way of estimating the coefficients of an AR process is through regression. Simulate an AR(1) process and make sure you keep a segment of data (500 data points) that does not depend on the starting values. Regress the process X_t on X_{t-1} using the function `lm()`. Plot the process, acf, residuals and residual acf. Comment on the fit (R^2 , t-test for coefficients etc). Simulate an AR(2) process and repeat the exercise. What happens if you fit an AR(2) model to the AR(1) data? What happens if you fit an AR(1) model to the AR(2) data?

6 The acf and Tests for Randomness

If your model is adequate, the residuals should appear random, white noise (uncorrelated). An even stronger condition is that the residuals should be independent. The acf allows you to see if the linear dependency structure has been removed from the process using your model - that is the correlation structure. The residuals can still have non-linear dependencies.

In the text and in lecture we have discussed the asymptotic confidence bands for the estimated acf. For iid observations $\rho(h) \sim N(0, 1/n)$, for large n . The confidence band $\pm 1.96/\sqrt{n}$ is *pointwise*. You should only use the iid confidence band as a guideline, remember it's asymptotic. Beware of data snooping. Simulate a white noise process and plot the acf with confidence bands for $n = 25, 50, 500$. Compare with the AR(1), $n = 25, 50, 500$.

6.1 The Portmanteau test

A test that is more directed at simultaneous testing of $\rho(h)$, for multiple h , is the Portmanteau test. The test statistic $Q = n * \sum_{j=1}^P \rho^2(j)$ is asymptotically χ_P^2 distributed if the observations are iid. If the process is correlated Q will be inflated. The choice of P is not easily made. Small P means you ignore large lag correlations, which may be important. Large P leads to a loss of power in the test.

Simulate and MA(1) and MA(2) process. Use $n = 25, 50, 500$ and perform the Portmanteau test with two or more choices of P . Compare with a white noise process (same n). Comment on the power, and the size of n . Give the p-values of the tests.

```
rho<-acf(x)$acf
```

rho now contains the $\rho(h)$ function.

```
Q<-n*sum(rho[2:(P+1)]^2)
```

(Note that the sum starts at 2 and ends at P+1, since index 1 correspond to $\rho(0)$).

6.2 The turning point test

Implement the turning point test (see the book, p. 36). Check the AR(1), the residuals from your regression fit, and a white noise process using this test. Conclusions? What about the size of the sample n , how does n affect the outcome of the test?

6.3 A simulation study

Write a program that generates 500 Gaussian white noise processes of length $n = 250$. Calculate Q and the turning point test statistic T for each data set. Check the empirical distribution of Q and T against the asymptotic distribution (χ^2 , Gaussian). Do the same for $n = 50$.

7 Durbin-Levinson and the pacf

The best linear predictor of X_{n+m} based on X_1^n is given by $\phi_{n1}X_n + \phi_{n2}X_{n-1} + \dots + \phi_{nm}X_1$ (section 2.5.1 in the book), where the coefficients ϕ are given by (assuming a mean zero process)

$$\Gamma_n \phi_n = \gamma_n^{(m)}$$

Simulate an AR(2) process with coefficients of your own choice, and $n = 500$. Estimate the autocovariance function γ (`acf()`, `type=`, make sure you extract the `acvf`, not the `acf`). Use the first 10 covariances to construct the matrix Γ ;

```
Gammamatrix<-matrix(0,10,10)
```

```
Gammamatrix[1,]<-c(covf[1:10])
```

```
Gammamatrix[2,]<-c(covf[2],covf[1:9])
```

etc

Solve for the best linear predictor coefficients (`solve()`) for $m = 1$ (one-step prediction) and $m = 2$. Compared with the theoretical result using the true coefficients.

7.1 Durbin-Levinson

It is not always feasible to solve for ϕ in the direct manner above, this required the inversion of a matrix which can be quite large. The Durbin-Levinson algorithm (2.5.1 in the book) allows for the iterative estimation of ϕ . In addition, we estimate the pacf as an intermediate step. The Durbin-Levinson algorithm is useful for deciding the order of an AR process. (How?).

The `acf()` function uses the D-L algorithm to compute the pacf. Simulate an AR(2) process and plot the pacf.

Now implement the D-L algorithm yourself (by hand). For which h is ϕ_{hh} small compared with $1/\sqrt{n}$? Show your work (i.e. the derived values for ϕ_{hh}) up to order 3. What is the reduction in MSE at each step of the algorithm, as stipulated by theory and observed from real data?