London R Practice

Rob Hayward

September 15, 2013

1 Introduction

This is a file to run through the presentation that was made at the September 2013 London R by Maarten Speekenbrink.

I am not sure how to load the appropriate packages (so this can probably be deleted in the future). However, I will do that in a chunk. The information is in the title: dependent mixed models. This is to implement micture and hidden Markov models.

2 Mixed Models

From the London R presentation. In a *mixture model* each observation is assumed to be drawn from a number of distinct subpopulation ("component distributions" according to Maarten). The distribution from which the observation is drawn is not directly observable and therefore it is represented by a *latent state*.

A mixture distribution is defined as

$$p(Y_1 = y) = \sum_{i-1}^{N} p(Y_t = y|S_t = i)P(S_t = i) \tag{1}$$

where,

- $S_t \in 1, \ldots, N$ denotes the latent state or class of observation t
- $P(S_t = i)$ denots the probability of the latent state t equals i
- $p(Y_t = y|S_t = i)$ denotes the density of observation of Y_t conditional on latent state being $S_t = i$.

3 Perth Water Example

4 Change Model

This is a model that assumes one or more discrete change points in the data. It may be the mean, trend or other parameters that may change. In this example with the S&P 500 it is the mean and the standard deviation. There is a transition matrix that defines the change points. For example, if there is one transition the matrix would be along the lines of

$$\begin{pmatrix} p_1 & 1 - p_1 \\ 0 & 1 \end{pmatrix}$$

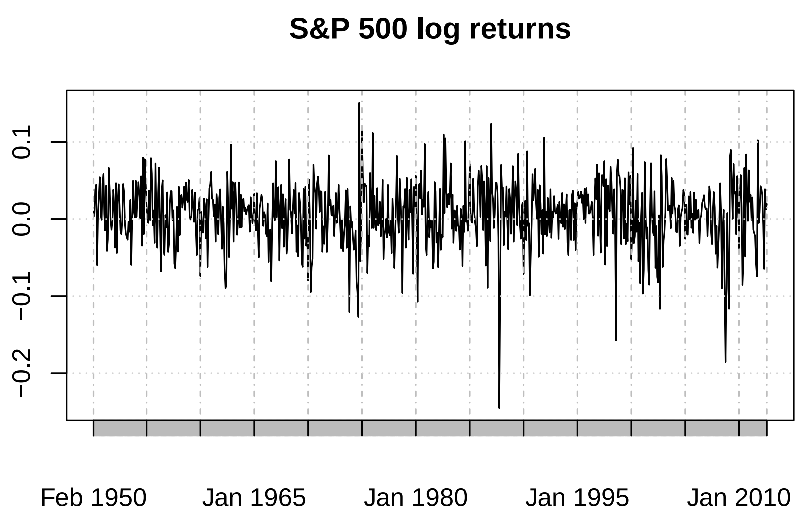
Where p_1 is the probability that the system will be in state 1. One there is a switch to state two. This matrix can be extended for more states. Need to come back and look at this if I can get the data.

5 S&P 500 Example

```
library(TTR)
# load SP500 returns
Sys.setenv(tz = "UTC")
sp500 <- getYahooData("^GSPC", start = 19500101, end = 20120909, freq = "daily")
ep <- endpoints(sp500, on = "months", k = 1)
sp500 <- sp500[ep[2:(length(ep) - 1)]]
sp500$logret <- log(sp500$Close) - lag(log(sp500$Close))
sp500 <- na.exclude(sp500)
```

Now plot the data to get an idea of what it looks like.

```
plot(sp500$logret, main = "S&P 500 log returns")
```



The aim is to identify the bull and bear markets. First set up the model (mod). This is a logistic regression ("logret") with two states. Then fit the model.

```
mod <- depmix(logret ~ 1, nstates = 2, data = sp500)
set.seed(1)
fm2 <- fit(mod, verbose = FALSE)

## iteration 88 logLik: 1348
```

The number of iterations and the log likelihood is printed. Now summarise the information.

```
depmixS4::summary(fm2)

## Initial state probabilties model
## Model of type multinomial (identity), formula: ~1
## <environment: 0x000000001202e608>
## Coefficients:
##           [,1] [,2]
## [1,] 1.735e-47    1
##
## Transition model for state (component) 1
## Model of type multinomial (identity), formula: ~1
## <environment: 0x0000000011a4ccb8>
## Coefficients:
## [1] 0.8215 0.1785
##
## Transition model for state (component) 2
## Model of type multinomial (identity), formula: ~1
## <environment: 0x0000000011a4ccb8>
## Coefficients:
## [1] 0.03914 0.96086
##
##
## Response model(s) for state 1
##
## Response model for response 1
## Model of type gaussian (identity), formula: logret ~ 1
## Coefficients:
## [1] -0.01505
## sd  0.06484
##
##
## Response model(s) for state 2
##
```

```
## Response model for response 1
## Model of type gaussian (identity), formula: logret ~ 1
## Coefficients:
## [1] 0.01045
## sd  0.03378
```