ENSAI (Non)-homogeneous Markov chains 2019-2020, Smart Data Master

Project 3 : Markov switching models

Simulated Data

In this section simulated returns data will be generated from separate Gaussian distributions, each of which represents a "bullish" or "bearish" market regime. The bullish returns draw from a Guassian distribution with positive mean and low variance, while the bearish returns draw from a Gaussian distribution with slight negative mean but higher variance.
Five separate market regime periods will be simulated and "stitched" together in R. The subsequent stream of returns will then be utilised by a Hidden Markov Model in order to infer posterior probabilities of the regime states, given the sequence of observations.

The first task is to install the depmixS4 and quantmod libraries and then import them into R. The random seed will also be fixed in order to allow consistent replication of results :

```
> install.packages('depmixS4')
> install.packages('quantmod')
> library('depmixS4')
> library('quantmod')
> set.seed(1)
```

At this stage a two-regime market will be simulated. This is achieved by assuming market returns are normally distributed. Separate regimes will be simulated with each containing N_k days of returns. Each of the k regimes will be bullish or bearish. The goal of the Hidden Markov Model will be to identify when the regime has switched from bullish to bearish and vice versa.

In this example $k = 5$ and $N_k \in [50, 150]$. The bull market is distributed as $\mathcal{N}(0.1, 0.1)$ while the bear market is distributed as $\mathcal{N}(0.05, 0.2)$. The parameters are set via the following code :

```
>  Create the parameters for the bull and bear market returns distributions
> Nklower <- 50
> Nkupper <- 150
> bullmean <- 0.1
> bullvar <- 0.1
> bearmean <- -0.05
> bearvar <- 0.2
```

The N_k values are randomly chosen :

```
>  Create the list of durations (in days) for each regime
> days <- replicate(5, sample(Nklower :Nkupper, 1))
```

The returns for each kth period are randomly drawn :

```
>  Create the various bull and bear markets returns
> marketbull1 <- rnorm( days[1], bullmean, bullvar )
> marketbear2 <- rnorm( days[2], bearmean, bearvar )
> marketbull3 <- rnorm( days[3], bullmean, bullvar )
> marketbear4 <- rnorm( days[4], bearmean, bearvar )
> marketbull5 <- rnorm( days[5], bullmean, bullvar )
```

The R code for creating the true regime states (either 1 for bullish or 2 for bearish) and final list of returns is given by the following :

```
>  Create the list of true regime states and full returns list
> trueregimes <-c( rep(1,days[1]), rep(2,days[2]), rep(1,days[3]), rep(2,days[4]), rep(1,days[5]))
> returns <-c( marketbull1, marketbear2, marketbull3, marketbear4, marketbull5)
```

Plotting the returns shows the clear changes in mean and variance between the regime switches :

```
> plot(returns, type="l", xlab=”, ylab="Returns")
```

At this stage the Hidden Markov Model can be specified and fit using the Expectation Maximisation algorithm (see p.30 of Chapter 1). The family of distributions is specified as gaussian and the number of states is set to two (nstates = 2) :

```
> Create and fit the Hidden Markov Model
> hmm <- depmix(returns ∼ 1, family = gaussian(), nstates = 2, data=data.frame(returns=returns))
> hmmfit <- fit(hmm, verbose = FALSE)
```

Subsequent to model fitting it is possible to plot the posterior probabilities of being in a particular regime state. $post_{\text{probs}}$ contain the posterior probabilities computed by the Forward-Bacward algorithm. These are compared with the underlying true states.

```
> Output both the true regimes and the posterior probabilities of the regimes
> post_probs <- posterior(hmmfit)
> layout(1 :2)
> plot(post_probs$state, type='s', main='True Regimes', xlab='', ylab='Regime')
> matplot(post_probs[,-1], type='l', main='Regime Posterior Probabilities', ylab='Probability')
> legend(x='topright', c('Bull','Bear'), fill=1 :2, bty='n')
```

1. **Question :** Can you write the model behind this code.

Application on real financial data

In the above section it was straightforward for the Hidden Markov Model to determine regimes because they had been simulated from pre-specified set of Gaussians. As stated above the problem of Regime Detection is actually an unsupervised learning challenge since the number of states is not known a priori, nor is there any "ground truth" on which to "train" the HMM.

In this section different modelling tasks will be carried out for different number of states. The results between these models will be compared by using citerions selection model. Instead of generating the returns stream from two Gaussian distributions it will simply be downloaded using the quantmod library :

```
>  Obtain S&P500 data from 2004 onwards
> getSymbols( "^GSPC", from="2004-01-01" )
```

1. From the S&P500 time series, compute the returns.
2. Plot the returns and comment the figure.
3. As before, fit using the EM algorithm different n-state Hidden Markov Model and plot for each model the returns and posterior probabilities of each regime.
4. By using the BIC or AIC criterion choose the model that best fit your data.
5. Interpret your result. Is your estimation in accordance with financial market ? For example, does your estimation lead to recover the high volatile regime to the sub-prime crisis between 2007-2009 ?
6. Make the same analysis for the "IWM" time series [1]. This serie can be obtained as S&P500 by using getSymbol from the quantmod library.

1. IMW Financial is a financial planning firm specializing in small business owners.