Hidden Markov Models

Redmond Xia

August 20, 2020

In this notebook, we will apply Hidden Markov model to simulated data to test if it can predict market regimes.

At this stage, we will only simulate two-regime and assuming the market returns are normally distributed. These are the parameters used to predict a bull and bear market returns

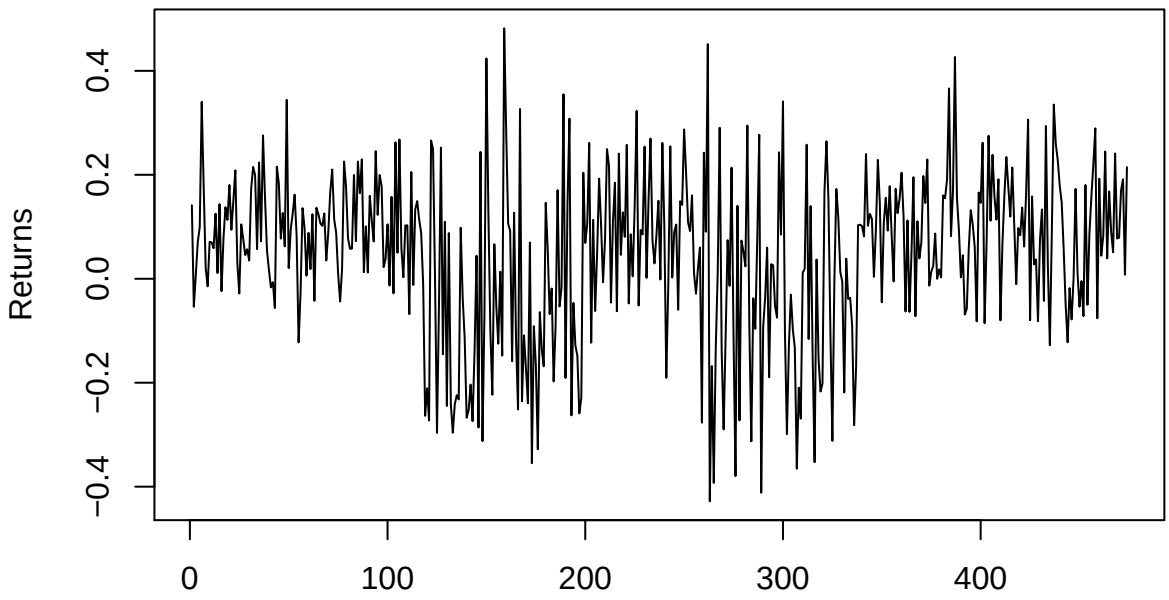
```
Nk_lower <- 50
Nk_upper <- 150
bull_mean <- 0.1
bull_var <- 0.1
bear_mean <- -0.05
bear_var <- 0.2
days <- replicate(5, sample(Nk_lower:Nk_upper, 1))
```

Below is returns for kth period are randomly drawn

```
market_bull_1 <- rnorm( days[1], bull_mean, bull_var)
market_bear_2 <- rnorm( days[2], bear_mean, bear_var)
market_bull_3 <- rnorm( days[3], bull_mean, bull_var)
market_bear_4 <- rnorm( days[4], bear_mean, bear_var)
market_bull_5 <- rnorm( days[5], bull_mean, bull_var)
```

Encoding 1 for bullish and 2 for bearish

```
true_regimes <- c( rep(1, days[1]), rep(2,days[2]), rep(1,days[3]), rep(2,days[4]), rep(1, days[5]))
returns <- c( market_bull_1, market_bear_2, market_bull_3, market_bear_4, market_bull_5)
plot(returns, type='l', xlab = '', ylab = 'Returns')
```

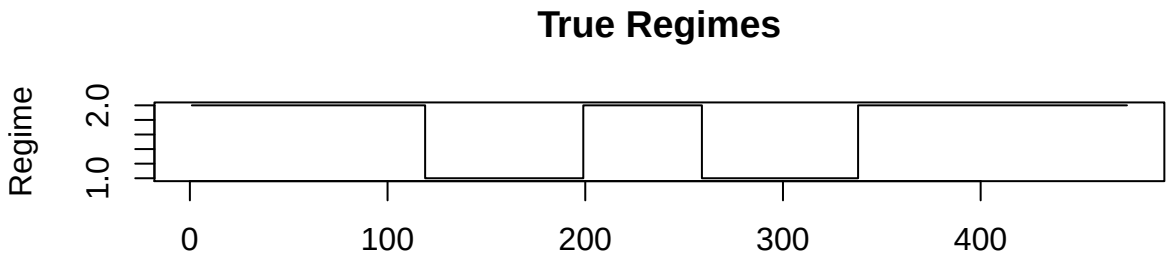


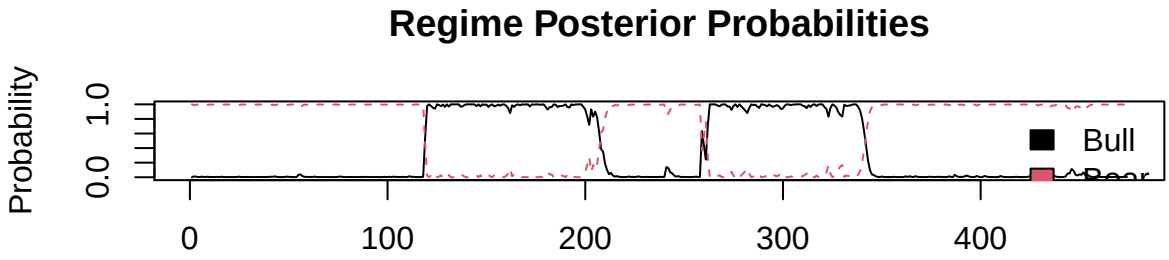

```
hmm <- depmix(returns ~ 1, family = gaussian(), nstates=2, data= data.frame(returns=returns))
hmmfit <- fit(hmm,verbose = FALSE)
```

```
## converged at iteration 24 with logLik: 289.6389
```

Outputting both the true regimes and the posterior probabilities of the regimes

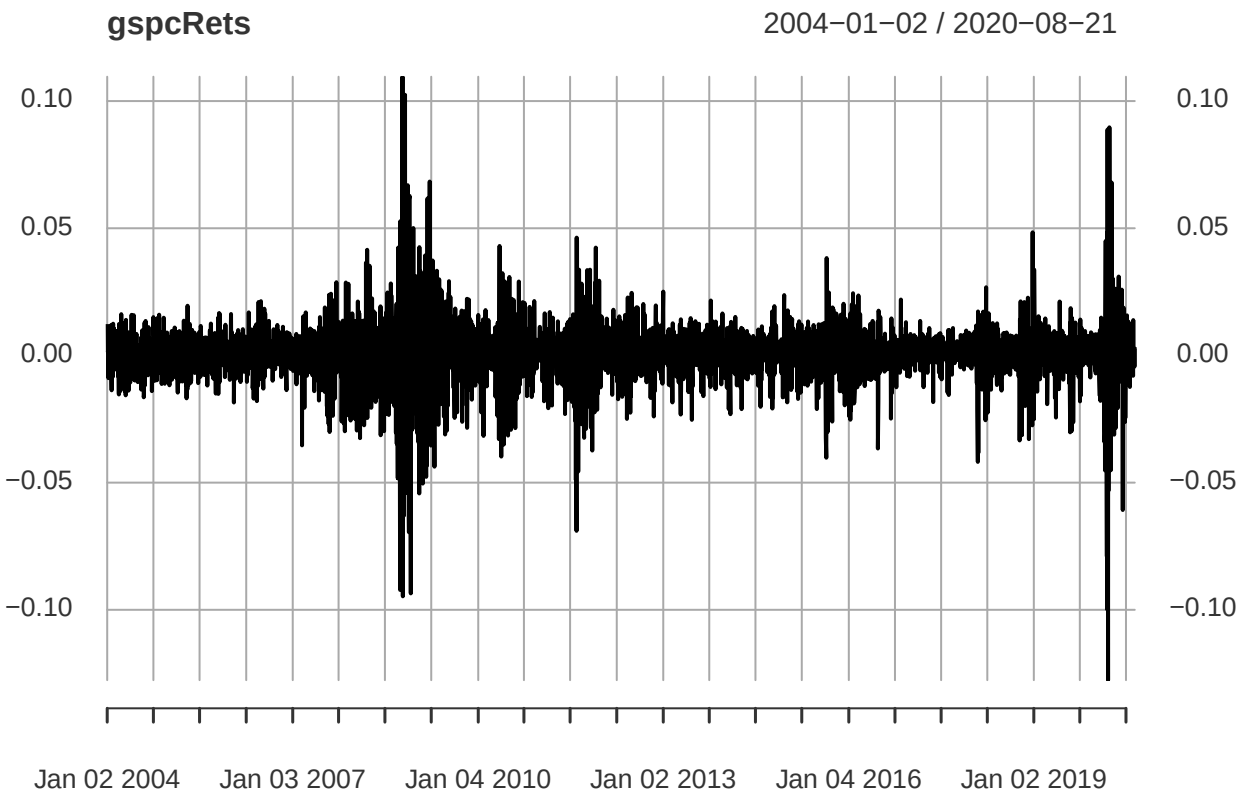
```
post_probs <- posterior(hmmfit)
layout(1:2)
plot(post_probs$state, type='s', main='True Regimes', xlab="",ylab = 'Regime')
matplot(post_probs[,-1], type="l", main='Regime Posterior Probabilities', ylab='Probability')
legend(x='topright', c('Bull','Bear'), fill=1:2, bty='n')
```





Instead of using simulated data we will now run it on sp500 returns

```
layout(1)
plot(gspcRets)
```



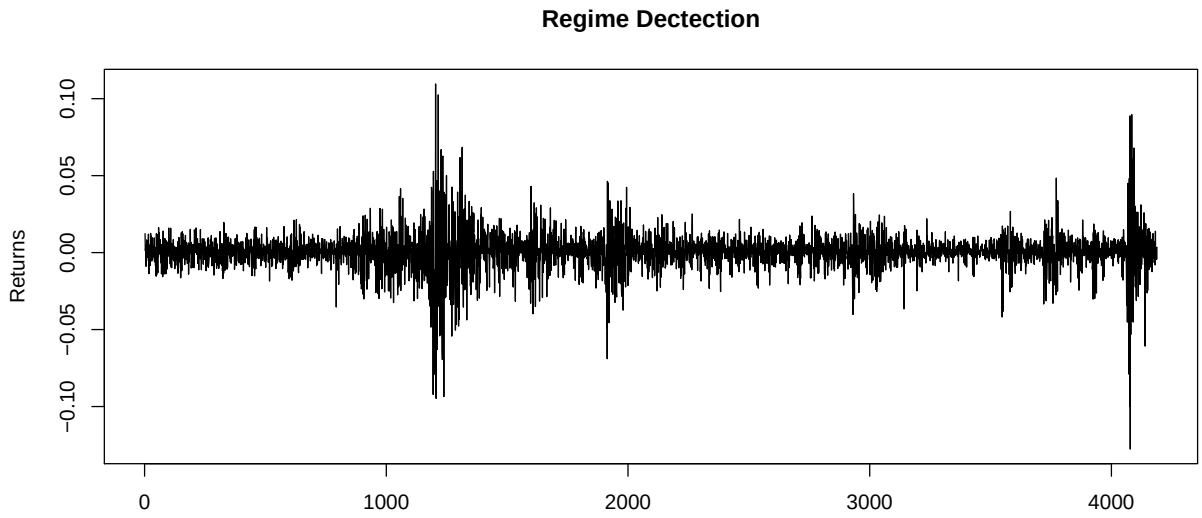
We will now proceed to fit the two state HMM with EM algorithm

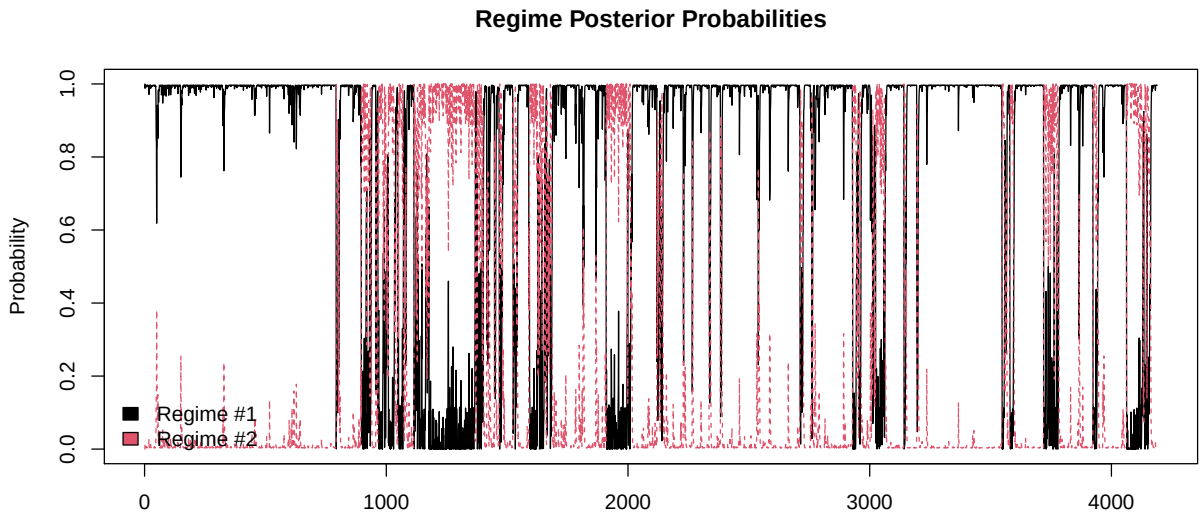
```
hmm <- depmix(returns ~ 1, family=gaussian(), nstates=2, data=data.frame(returns=returns))
hmmfit <- fit(hmm, verbose=FALSE)
```

```
## converged at iteration 35 with logLik: 13676.17
```

```
post_probs = posterior(hmmfit)

# Plotting the return stream and the posterior probabilities of the separate regimes
layout(1:2)
plot(returns, type='l', main='Regime Dectection', xlab='', ylab='Returns')
matplot(post_probs[,-1], type='l', main='Regime Posterior Probabilities', ylab='Probability')
legend(x='bottomleft', c('Regime #1','Regime #2'), fill=1:2, bty='n')
```






```
rownames(post_probs) <- index(GSPC)
```

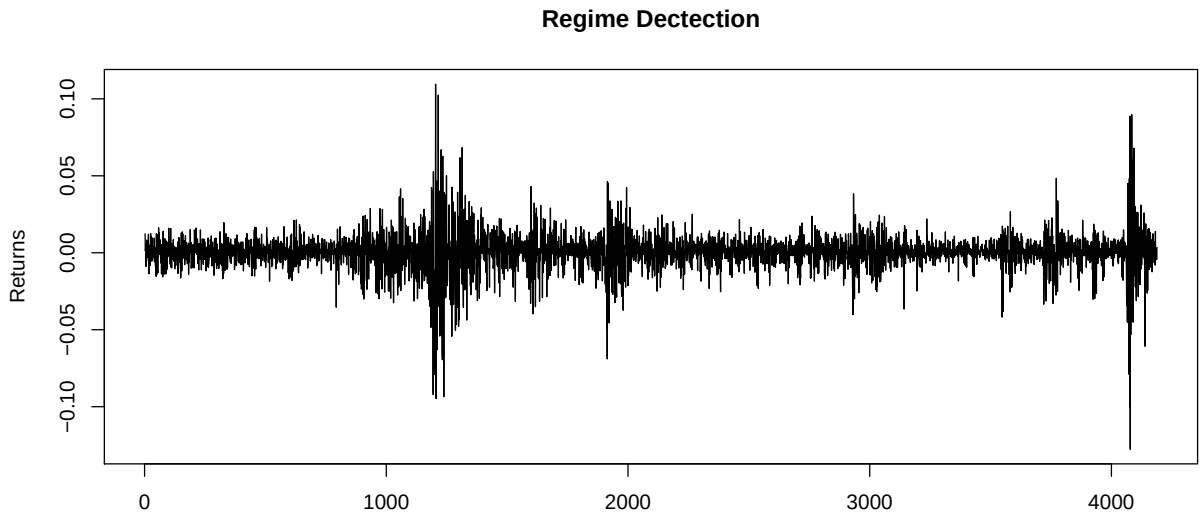
We will now switch to a 3 state regime

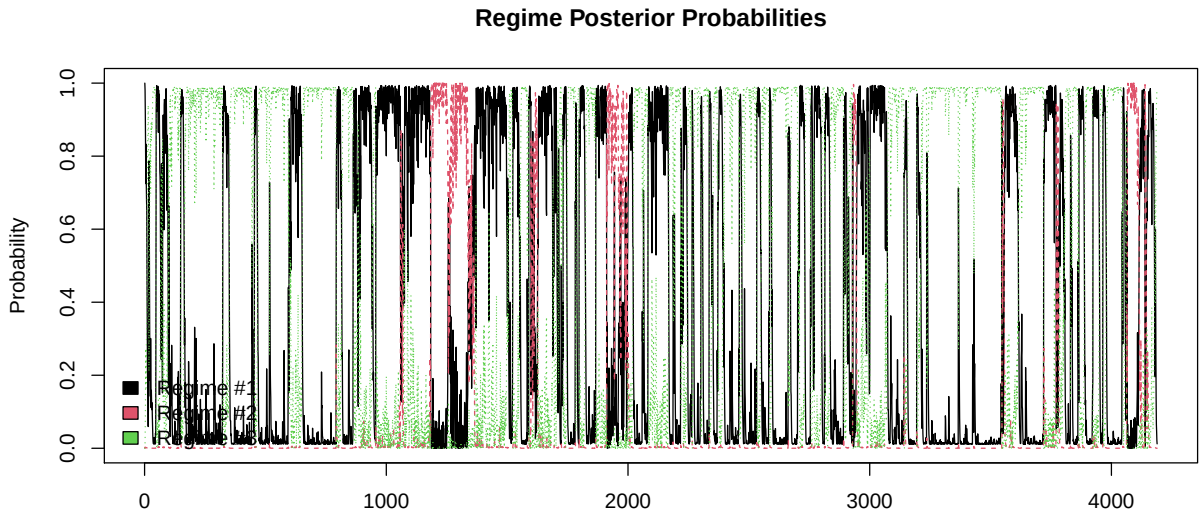
```
hmm <- depmix(returns ~ 1, family=gaussian(), nstates=3, data=data.frame(returns=returns))
hmmfit <- fit(hmm, verbose=FALSE)
```

```
## converged at iteration 112 with logLik: 13892.61
```

```
post_probs = posterior(hmmfit)

# Plotting the return stream and the posterior probabilities of the separate regimes
layout(1:2)
plot(returns, type='l', main='Regime Dectection', xlab='', ylab='Returns')
matplot(post_probs[,-1], type='l', main='Regime Posterior Probabilities', ylab='Probability')
legend(x='bottomleft', c('Regime #1','Regime #2','Regime #3'), fill=1:3, bty='n')
```






```
rownames(post_probs) <- index(GSPC)
```