BIOS 560R: Advanced Statistical Computing

Fall 2012 Homework 3

Due 10/18/2012 at 4pm before the class

Hidden Markov model (HMM) is useful for modeling financial time series data such as the stock prices. In this homework we will practice using HMM to model the daily price of a hypothetic stock.

Define the closing price for the stock at day t is $x_t, t = 0, \ldots, T$, and the "log returns" as $r_t = log(x_t/x_{t-1}), t = 1, \ldots, T$. We can assume the market for day t belongs to one of the 3 states: "bullish", "bearish" and "flat" , which means that the prices are going up, going down or fluctuating. Denote the state of day t by Z_t, the daily log return can be modeled by following 3-state HMM with Normal emission probability:

$$
\begin{aligned}
&Pr(Z_1 = k) = \pi_k \\
&Pr(Z_{t+1} = l|Z_t = k) = P_{kl} \\
&r_t|Z_t = k \sim N(\mu_k, \sigma_k^2), k = 1, 2, 3
\end{aligned}
$$

Obtain the simulated price data from the class website, then answer following questions:

1. Why is it important to use log returns instead of the daily price in the HMM? Can we formulate a HMM using the prices? **Hint**: consider the assumptions of a homogeneous Markov chain, then check whether the data satisfy these assumptions.

2. The parameters for a 3-state HMM are $(\pi_k, P_{kl}, \mu_k, \sigma_k)$. Forward-backward algorithm with Baum-Welch can be applied to estimate these parameters iteratively. At iteration i,

   (a) Given current values of the parameters, write down the expressions of forward and backward probabilities.

   (b) Give the forward and backward probabilities, write down the procedures for estimating model parameters.

3. Implement the forward and backward algorithm with Baum-Welch in a programming language of your choice, and report the estimates of model parameters.

4. Assume we want to predict tomorrow's stock price based on our HMM results. What's the marginal distribution of x_{T+1} given $x_1, \ldots, x_T$? What are its mean and variance? **Hint**: if you cannot derive the closed form solution, use a simulation to obtain these values.

Computational tip: When the HMM chain is long, the computation of forward/backward matrices has to be done in logarithm scale, otherwise they will become negative infinity very quickly. However to evaluate the sum in log-scale is tricky, for example, $\log(e^a + e^b)$ will become negative infinity when a or b are negative number with large absolute values. Use the following trick to deal with the scenario:

$$log(e^a + e^b) = log(e^a(1 + e^{b-a})) = a + log(1 + e^{b-a})$$

It equals b when $b >> a$, equals a when $b << a$. When the values of b and a are close, the computation is numerically stable. Following is an R implementation of the algorithm, which works for two vectors:

```
Raddlog <- function (a, b)
{
    result <- rep(0, length(a))
    idx1 <- a > b + 200
    result[idx1] <- a[idx1]
    idx2 <- b > a + 200
    result[idx2] <- b[idx2]
    idx0 <- !(idx1 | idx2)
    result[idx0] <- a[idx0] + log1p(exp(b[idx0] - a[idx0]))
    result
}
```

A simple test (with a and b being -1000) shows that they work well, whereas directly summing up then taking log doesn't work:

```
> Raddlog(-1000, -1000)
[1] -999.3069
> log(exp(-1000)+exp(-1000))
[1] -Inf
```

Disclaimer: the stock price is simulated and this homework is for training purpose only. Please do NOT attempt to apply this to real world trading. The modeling of actual financial data is far more complicated.