CHAPTER 17

HIDDEN MARKOV CHAINS

Suppose that in some setting there is a Markov chain (MC) that is evolving with time. We are interested in knowing what state the MC is in at any particular time, however we have only imperfect information to go on. This situation is know as a *hidden* Markov chain — the MC is there, but we can't directly observe it. Here we give two examples: the first a simple example to illustrate the concept; the second a more serious exercise in estimating a stock's beta.

§1. The Concept

Let $(X_n : n \geq 0)$ be a two state MC (call the states 0 and 1) with PTM $\mathbf{P} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, and suppose $X_0 \sim (\alpha_0, \beta_0) = (0.5,\, 0.5)$. (The initial distribution of the MC is virtually irrelevant.) Here, of course, $a + b = c + d = \alpha_0 + \beta_0 = 1$. Suppose I can observe the realization of this MC but you can't. At each step of the MC I tell you what state it is in, however, I occasionally lie to you. In fact, I lie to you each step with probability p. This produces a sequence $(Y_n : n \geq 1)$ where

$$Y_n = \begin{cases} X_n \text{ (the truth)} & \text{with probability } 1-p \\ 1 - X_n \text{ (a lie)} & \text{with probability } p, \end{cases}$$

where the choice (truth or lie) is independent of $X_0, \ldots, X_n$ and independent of previous truth/lie choices. Here (X_n) is the hidden MC and (Y_n) is the imperfect information you have about its state. Initially we assume that you know the MC's PTM and what p is.

For a realization $(y_1, \ldots, y_N)$ of the Y process and any $n \leq N$, let

$$D_n = \{Y_1 = y_1, \ldots, Y_n = y_n\}, \tag{1}$$

and put $P_n[E] = P[E|D_n]$ for any event E. (When $n = 0$, so there is not yet any history for the Y process, we take $D_0 = \Omega$, the whole sample space, so $P[D_0] = 1$ and $P_0[E] = P[E]$.) Suppose you have observed the Y process up to period $n-1$ and, through your Bayesian prowess, have concluded that $P_{n-1}[X_{n-1} = 0] = \alpha_{n-1}$ and $P_{n-1}[X_{n-1} = 1] = \beta_{n-1}$. I then observe the next step of the MC, X_n, and report to you Y_n. If I report to you that $Y_n = 0$ you will update as follows:

$$\begin{aligned} \alpha_n &= P_{n-1}[X_n = 0 | Y_n = 0] \\ &= P_{n-1}[Y_n = 0 | X_n = 0] \cdot \frac{P_{n-1}[X_n = 0]}{P_{n-1}[Y_n = 0]}. \end{aligned} \tag{2}$$

Now

$$P_{n-1}[Y_n = 0 | X_n = 0] \;=\; P[\text{I tell the truth at step } n] \;=\; 1 - p.$$

Also, the P_{n-1} distribution of X_{n-1} is $(\alpha_{n-1}, \beta_{n-1})$, so the P_{n-1} distribution of X_n is (p_0, p_1) where

$$\begin{aligned} (p_0, p_1) &= \big(P_{n-1}[X_n = 0],\; P_{n-1}[X_n = 1]\big) \\ &= (\alpha_{n-1}\; \beta_{n-1}) \begin{pmatrix} a & b \\ c & d \end{pmatrix} \;=\; (\alpha_{n-1}a + \beta_{n-1}c,\; \alpha_{n-1}b + \beta_{n-1}d), \end{aligned}$$

so $p_0 = P_{n-1}[X_n = 0] = \alpha_{n-1}a + \beta_{n-1}c$ and $p_1 = 1 - p_0$. Additionally,

$$\begin{aligned} P_{n-1}[Y_n = 0] &= P_{n-1}[Y_n = 0 | X_n = 0]\, P_{n-1}[X_n = 0] \\ &\quad + P_{n-1}[Y_n = 0 | X_n = 1]\, P_{n-1}[X_n = 1] \\ &= (1 - p)p_0 + pp_1. \end{aligned} \tag{3}$$

Setting all of this into (2) gives that

$$\alpha_n \;=\; (1 - p) \cdot \frac{p_0}{(1 - p)p_0 + pp_1},$$

and, of course $\beta_n = 1 - \alpha_n$.

This, recall, is if I report to you that $Y_n = 0$. The reader will verify that, if I report that $Y_n = 1$, you will update as follows:

$$\alpha_n \;=\; p \cdot \frac{p_0}{pp_0 + (1 - p)p_1} \quad \text{and} \quad \beta_n \;=\; 1 - \alpha_n.$$

This hidden MC is implemented in the code `HiddenMC.cpp`. The program asks the user to specify the transition matrix and the probability, p, that I lie to you about the current state. It then simulates 50,000 steps of both the X and Y processes, reporting to the screen for the first 20 steps the current assessment of X_n's distribution (α_n, β_n) given the data $Y_1, \ldots, Y_n$, and reporting to the output file `YProcess.txt` the Y_n for $0 \le n \le 50{,}000$.

Some interesting cases are when $p = 0$ or $p = 1$, in which case you have perfect information (if I *always* lie you know the truth!). Also interesting is the case $p = 0.5$, in which case the reported Y_n contains no information — regardless of X_n I report that $Y_n = 0$ or 1, each with probability 0.5. In this case the value of (α_n, β_n) converges to the MC's invariant distribution. This is because $P_n[X_n = 0] = P[X_n = 0]$ and $P_n[X_n = 1] = P[X_n = 1]$ and the vector $\big(P[X_n = 0],\, P[X_n = 1]\big)$ converges to the invariant distribution as n gets large.

When the Parameters Are Not Known. The above analysis assumes that you know the PTM, $\mathbf{P}$ (determined by a and c), and the probability, p, that I lie. Here we relax these assumptions and suppose that these parameters are unknown. Using the data, you can use the maximum likelihood technique to estimate the parameters a, c, and p. Specifically, if trial values of a, c, and p are specified you may calculate the corresponding probability of observing your particular realization of the Y process $(y_1, \ldots, y_N)$. Suppose, for $n \leq N$, you have calculated $P[D_{n-1}]$, as defined in (1). You then calculate

$$\begin{aligned} P[D_n] &= P[D_{n-1} \cap \{Y_n = y_n\}] \\ &= P[D_{n-1}] \cdot P[Y_n = y_n | D_{n-1}] \\ &= P[D_{n-1}] \cdot P_{n-1}[Y_n = y_n]. \end{aligned}$$

Now (3) tells you how to calculate $P_{n-1}[Y_n = 0]$, and $P_{n-1}[Y_n = 1] = 1 - P_{n-1}[Y_n = 0]$. Start with $P[D_0] = 1$ and proceed forward in this manner until you have calculated $P[D_N]$, which is what you want. For this particular choice of a, c, and p, let $L(a, c, p) = P[D_N]$ denote the probability of observing this data with these parameters. Then loop through a grid of values $0 < a < 1$, $0 < c < 1$ and $0 < p < 0.5$ (see the Exercise 1 below for why we do not consider $0.5 < p < 1$), and find the triple (a, c, p) that maximizes L. This is implemented in `MLEHiddenMC.cpp`, where the file `YProcess.txt` is read in and the MLE values of a, c, and p are reported.

We note that multiplying successive numbers that are between 0 and 1 will rapidly lead to underflow on the computer. For this reason we calculate $\ln L$ instead and use that

$$\ln P[D_n] = \ln P[D_{n-1}] + \ln P_{n-1}[Y_n = y_n]$$

and choose the parameter values that maximize $\ln P[D_N]$.

Exercise 1. If we include $0.5 < p < 1$ in the search, there be two solutions. Suppose $\mathbf{P} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ with lying probability p maximizes L. Then $\widetilde{\mathbf{P}} = \begin{pmatrix} d & c \\ b & a \end{pmatrix}$ and $\widetilde{p} = 1 - p$ also maximizes L. Explain why!

§2. A Brief Regression Digression

Here we briefly review some elementary facts from OLS regression. This is useful both below and in another *Gibbs sampling* application coming in a subsequent chapter. Suppose $x_1, \ldots, x_n$ are non-random, and a sequence of random variables $(Y_t : 1 \leq t \leq n)$ is modeled as $Y_t = \beta x_t + \epsilon_t$, where the (ϵ_t) are iid Normal $(0, \sigma_\epsilon^2)$ and β and σ_ϵ are unknown parameters. With observed values $y_1, \ldots, y_n$ we may estimate β by choosing b to minimize the sum of squared errors:

$$SOS = \sum_t (y_t - bx_t)^2.$$

Now $\frac{\partial}{\partial b} SOS = -2\sum_t (y_t - bx_t)x_t$, which is zero when $\sum_t x_t y_t = b\sum_t x_t^2$, or

$$b \;=\; \widehat{\beta} \;=\; \frac{\sum_t x_t y_t}{\sum_t x_t^2}.$$

Since the ϵ_ts are independent and normally distributed, $\widehat{\beta}$ is also the maximum likelihood estimate for β. To see this, note that if $\beta = b$ the density function for Y_t is $f_t(y) = \frac{1}{\sqrt{2\pi}\sigma_\epsilon} e^{-(y-bx_t)^2/2\sigma_\epsilon^2}$ and by independence the likelihood of the observed data as a function of b is

$$\begin{aligned} L(b) \;&=\; \prod_t f_t(y_t) \;=\; \prod_t \frac{1}{\sqrt{2\pi}\sigma_\epsilon} e^{-(y-bx_t)^2/2\sigma_\epsilon^2} \\ &=\; \left(\frac{1}{\sqrt{2\pi}\sigma_\epsilon}\right)^n \exp\left[-\sum_t (y-bx_t)^2 \Big/ 2\sigma_\epsilon^2\right]. \end{aligned}$$

MLE chooses the value of the parameter that maximizes the likelihood function. Here, regardless of the value of σ_ϵ, this happens when the (negative) exponent on the right is maximized, i.e., when $\sum_t (y - bx_t)^2$ is minimized. Hence $b = \widehat{\beta}$ is the MLE estimate of β.

Now $\widehat{\beta}$ defined above is a realization of a random variable (which we'll also call $\widehat{\beta}$) that corresponds to our realization $y_1, \ldots, y_n$ of the Y_ts and is defined by

$$\widehat{\beta} \;=\; \frac{1}{\sum_t x_t^2} \sum_t x_t Y_t.$$

Recalling that $Y_t = \beta x_t + \epsilon_t$ we see that the Y_ts are independent and normal with $EY_t = \beta x_t + E\epsilon_t = \beta x_t$ and $\operatorname{Var} Y_t = \operatorname{Var} \epsilon_t = \sigma_\epsilon^2$. Hence

$$E\widehat{\beta} \;=\; \frac{1}{\sum_t x_t^2} \sum_t x_t E Y_t \;=\; \frac{1}{\sum_t x_t^2} \sum_t x_t \beta x_t \;=\; \beta,$$

and

$$\operatorname{Var} \widehat{\beta} \;=\; \left(\frac{1}{\sum_t x_t^2}\right)^2 \sum_t x_t^2 \operatorname{Var} Y_t \;=\; \left(\frac{1}{\sum_t x_t^2}\right)^2 \sum_t x_t^2 \sigma_\epsilon^2 \;=\; \frac{\sigma_\epsilon^2}{\sum_t x_t^2}.$$

Finally, observe that $\widehat{\beta}$ is normally distributed since it is a linear combination of independent normals. That is, $\widehat{\beta} \sim \text{Normal}\,\big(\beta, \sigma_\epsilon^2 / \sum_t x_t^2\big)$.

§3. Estimating a Stock's Beta

Suppose $\big((m_t, r_t) : 1 \le t \le N\big)$ is a time series of daily observations of the market's return, m_t, as represented, say, by the S&P 500, and an individual stock return r_t over the same day. We view the market time series (m_t) as given, i.e. as non-stochastic, and we view the stock time series (r_t) as a particular realization of a stochastic process (R_t). A common model for the (R_t) process is

$$R_t - r_f \;=\; \beta \cdot (m_t - r_f) + \epsilon_t,$$

where r_f is the one-day risk-free rate and the ϵ_ts are iid Normal$\,(0, \sigma_\epsilon^2)$ for some variance σ_ϵ^2. Now $R_t - r_f$ and $m_t - r_f$ are the stock's and the market's respective *excess* return over the risk-free rate. In the example that we shall consider, the risk-free rate r_f (the risk-free rate really should be time dependent as well) is extremely small. As a convenience we take it to be 0, so our model is $R_t \;=\; \beta m_t + \epsilon_t$.

With the time series data (m_t, r_t) in hand, we can estimate β using the standard regression techniques described above, yielding

$$\beta \;\approx\; \frac{\sum_t r_t m_t}{\sum_t m_t^2}.$$

A Specific Example. We have implemented this in `BetaRegression.cpp`. Here the data is daily for the five year period January 2012 through December 2016 for JP Morgan (the r_t) and the S&P 500 (the m_t) — there are 1258 daily returns for each. The above calculations yield the estimates $\beta \approx 1.302$ and $\sigma_\epsilon \approx 0.974$. Our estimate for σ_ϵ is the sample standard deviation of the residuals as computed with the estimated β:

$$\sigma_\epsilon \;\approx\; \sqrt{\sum_{t=1}^{1258} (r_t - 1.302 m_t)^2 \Big/ 1257} \;=\; 0.974.$$

Using Bayesian Inference. With a bit of trickery we may obtain the same result for β using Bayesian inference. To see this let Normal$\,(\mu_0, \sigma_0^2)$ denote a prior distribution for β, where these hyperparameters will be chosen shortly. We know that we may update to get a posterior distribution that reflects the data $(r_t : 1 \le t \le 1258)$ by updating sequentially: first for r_1, then for r_2, etc. To this end, suppose we have updated the distribution to reflect the data through time $t-1$, $\{R_1 = r_1, \ldots, R_{t-1} = r_{t-1}\}$, and that, given this data, we have deduced via Bayesian inference that $\beta \sim$ Normal$\,(\mu, \sigma^2)$. Initially we have this for $t = 1$, where there is no data and $\mu = \mu_0$ and $\sigma^2 = \sigma_0^2$ — the hyperparameters. Now let

$$X_t \;=\; \frac{R_t}{m_t} \;=\; \beta + \frac{\epsilon_t}{m_t},$$

so we have

$$\beta \sim \text{Normal}\,(\mu, \sigma^2) \quad \text{and then} \quad X_t \sim \text{Normal}\left(\beta, \frac{\sigma_\epsilon^2}{m_t^2}\right),$$

which is exactly the setting of Example 1 in Chapter 15. The realized value of X_t is $\frac{r_t}{m_t}$ so, using (4) in that Example 1 (with the substitutions $\beta \to \mu$, $\mu \to \mu_0$, $\sigma^2 \to \sigma_0^2$, $\frac{r_t}{m_t} \to x$, and $\frac{\sigma_\epsilon^2}{m_t^2} \to \sigma^2$) we update the distribution of β to reflect this new piece of data as follows:

$$\begin{aligned} \text{updated } \mu &= \frac{\frac{\sigma_\epsilon^2}{m_t^2}\mu + \sigma^2 \frac{r_t}{m_t}}{\frac{\sigma_\epsilon^2}{m_t^2} + \sigma^2} = \frac{\sigma_\epsilon^2 \mu + \sigma^2 r_t m_t}{\sigma_\epsilon^2 + m_t^2 \sigma^2}; \quad \text{and} \\ \text{updated } \sigma^2 &= \frac{\frac{\sigma_\epsilon^2}{m_t^2}\sigma^2}{\frac{\sigma_\epsilon^2}{m_t^2} + \sigma^2} = \frac{\sigma_\epsilon^2 \sigma^2}{\sigma_\epsilon^2 + m_t^2 \sigma^2}. \end{aligned} \tag{4}$$

We update sequentially in this fashion through $t = 1258$. Since the normal density function peaks at its mean, we find that the MAP estimator for β given the data is the value of μ at $t = 1258$.

We know that MAP = MLE when the prior distribution is uniform on the parameter's range. We cannot choose μ_0 and σ_0^2 so that Normal (μ_0, σ_0^2) is uniform. However, if σ_0 is extremely large then the Normal (μ_0, σ_0^2) density function is extremely flat, imitating a uniform distribution (the trickery!). In `BetaBayes.cpp` the above updating is carried out with a prior distribution for β that is Normal $(0, 100^2)$. The $t = 1258$ value of μ, 1.302 (the MAP estimate), agrees with the regression estimate (the MLE estimate) to at least three decimal places. `BetaBayes.cpp` also reports that the $t = 1258$ value of σ^2 is 0.034^2.

§4. The Full Model

So far there is no hidden Markov chain; we are about to introduce one. In the full model, β is time-dependent and we assume it undergoes the stochastic process

$$\begin{aligned} \beta_t &= \beta_{t-1} + \delta_t, \quad \text{and then} \\ R_t &= \beta_t m_t + \epsilon_t, \end{aligned}$$

where (δ_t) and (ϵ_t) are independent with each $\delta_t \sim$ Normal $(0, \sigma_\delta^2)$ and each $\epsilon_t \sim$ Normal $(0, \sigma_\epsilon^2)$ for some parameters σ_δ and σ_ϵ (to be estimated below). Here (β_t) is the hidden Markov chain and the data (R_t) is the imperfect information that we can actually observe. We note that this model with $\sigma_\delta = 0$ (so $\beta_t = \beta_{t-1}$) is the model discussed above where β is constant.

We assume that $\beta_0 \sim$ Normal (μ_0, σ_0^2) for some hyperparameters μ_0 and σ_0^2. Assume for the time being that we know the values of the parameters σ_δ and σ_ϵ. Assume we have deduced from the data through time $t-1$ that $\beta_{t-1} \sim$ Normal (μ, σ^2). Then, from the dynamics of the β process we will have $\beta_t \sim$ Normal $(\mu, \sigma^2 + \sigma_\delta^2)$. As above, we let

$$X_t = \frac{R_t}{m_t} = \beta_t + \frac{\epsilon_t}{m_t},$$

so

$$\beta_t \sim \text{Normal}\,(\mu, \sigma^2 + \sigma_\delta^2) \quad \text{and then} \quad X_t \sim \text{Normal}\left(\beta_t, \frac{\sigma_\epsilon^2}{m_t^2}\right).$$

Following (4) and using that the realized value for X_t is $\frac{r_t}{m_t}$, we update β_t's parameters as

$$\begin{aligned}\text{updated } \mu &= \frac{\sigma_\epsilon^2\mu + (\sigma^2 + \sigma_\delta^2) r_t m_t}{\sigma_\epsilon^2 + m_t^2(\sigma^2 + \sigma_\delta^2)}, \quad \text{and} \\ \text{updated } \sigma^2 &= \frac{\sigma_\epsilon^2(\sigma^2 + \sigma_\delta^2)}{\sigma_\epsilon^2 + m_t^2(\sigma^2 + \sigma_\delta^2)}.\end{aligned}$$

As above, we update sequentially in this fashion through $t = 1258$. At $t = 1258$, μ is the MAP estimator for JP Morgan's beta at December 2016.

Specifying the Parameters. We take the hyperparameters μ_0 and σ_0^2 to be 1.302 and 0.034^2, respectively. These are the $t = 1258$ parameter values for β in the constant beta case. We do not attempt to justify this choice but note only that the hyperparameters are of little relevance when there is an abundance of data.

The specification of σ_δ and σ_ϵ is of greater importance. As with the two state Markov chain, we use the maximum likelihood technique. With a trial choice of the two parameters in hand suppose at time $t-1$ we deduce that:

$$\begin{aligned} \beta_{t-1} &\sim \text{Normal}\,(\mu, \sigma^2); \quad \text{so} \\ \beta_t = \beta_{t-1} + \delta_t &\sim \text{Normal}\,(\mu, \sigma^2 + \sigma_\delta^2); \quad \text{and} \\ \beta_t m_t &\sim \text{Normal}\,(\mu m_t, (\sigma^2 + \sigma_\delta^2) m_t^2). \end{aligned}$$

Recalling that the independent $\epsilon_t \sim \text{Normal}\,(0, \sigma_\epsilon^2)$ we see that

$$\begin{aligned} R_t = \beta_t m_t + \epsilon_t &\sim \text{Normal}\,\left(\mu m_t, (\sigma^2 + \sigma_\delta^2) m_t^2 + \sigma_\epsilon^2\right) \\ &\sim \text{Normal}\,(\mu m_t, v), \end{aligned}$$

where $v = (\sigma^2 + \sigma_\delta^2) m_t^2 + \sigma_\epsilon^2$. Conditioned on the data through day $t-1$, the density function for R_t is therefore $f_t(r) = \frac{1}{\sqrt{2\pi v}} e^{-(r - \mu m_t)^2/2v}$. We evaluate this at the observed r_t and then compute

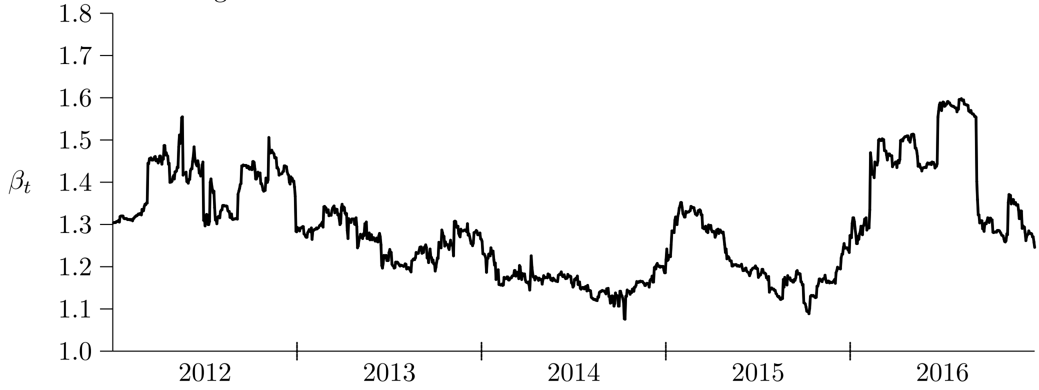
$$L(\sigma_\delta, \sigma_\epsilon) = \prod_{t=1}^{1258} f_t(r_t),$$

giving the likelihood of observing this data with these parameter values. (As usual we actually work with the logarithm of this to avoid underflow.)

We repeat this exercise over a grid of values for $(\sigma_\delta, \sigma_\epsilon)$ and numerically observe where $L(\sigma_\delta, \sigma_\epsilon)$ is maximized. This is implemented in `MLEParameters.cpp`, where to three decimal places the optimal values are found to be $\sigma_\delta = 0.015$ and $\sigma_\epsilon = 0.968$.

Beta Over Time. With these parameter values we calculate the parameters μ and σ^2 in `BetaFull.cpp` for the distribution of β_t for $t = 1$ to $t = 1258$. Recall that μ at time t is the MAP estimator for β_t. This is plotted in Figure 1 below.

Figure 1. Full model estimates of JPM's beta over time.

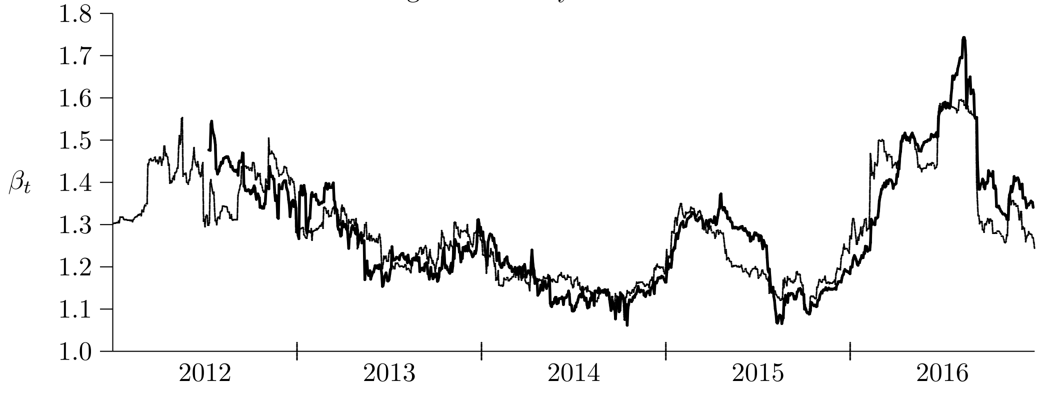


A Sanity Check. As a sanity check on all of this we estimate β_t using an entirely different approach. With a moving window of 130 days (that works best!) we calculate the maximum likelihood estimate for β_t using the data at the 130 consecutive times $Q_t = \{s : t - 130 \le s < t\}$:

$$\beta_t \approx \sum_{s \in Q_t} r_s m_s \bigg/ \sum_{s \in Q_t} m_s^2 .$$

This is implemented in `BetaWtdAve.cpp`; the heavier plot below is this moving average approach which is shown below starting at day 131 due to the Q_t window. Clearly we do not expect complete agreement with these two distinct mythologies, but the degree of coincidence is striking!

Figure 2. Sanity check.



Accompanying Code

The Two-State Model. `HiddenMC.cpp` generates a realization of the two-state hidden Markov chain of §1. The user is asked to input the parameters a and c of the PTM $\mathbf{P}$ as well as the probability, p, that I lie to you. It prints to the screen information about the first 20 steps of this process and reports 50,000 steps of the Y process to the output file `YProcess.txt`. The program `MLEHiddenMC.cpp` reads in the data from `YProcess.txt` and uses MLE to estimate the three parameters a, b, and p. The smaller p is, the better it does at guessing the true values. Please experiment.

JP Morgan's Beta. `BetaRegression.cpp` uses regression to determine the *constant* beta for JPM's stock price over the period January 2012 to December 2016. `BetaBayes.cpp` also calculates a constant beta for JPM but uses Bayesian inference rather than regression. `MLEParameters.cpp` uses maximum likelihood estimation to arrive at values for the parameters σ_δ and σ_ϵ in the full beta model. It employs a grid search to maximize the likelihood function $L(\sigma_\delta, \sigma_\epsilon)$, as discussed in §4. `BetaFull.cpp` implements the hidden Markov chain model as displayed in Figure 1 and `BetaWtdAve.cpp` implements the "sanity check" discussed above and plotted in Figure 2.