Estimating animal spirits

Grzegorz Andruszkiewicz

Imperial College London
g.andruszkiewicz09@imperial.ac.uk

(Work in collaboration with Mark H.A. Davis and Sébastien Lleo)

Animal spirits

Following behavioural economics we assume that the dynamics of the market are affected by the 'confidence' of market participants.

Already John Maynard Keynes said:

> [A] large proportion of our positive activities depend on spontaneous optimism rather than on a mathematical expectation, whether moral or hedonistic or economic. Most, probably, of our decisions to do something positive, the full consequences of which will be drawn out over many days to come, can only be taken as a result of animal spirits—of a spontaneous urge to action rather than inaction, and not as the outcome of a weighted average of quantitative benefits multiplied by quantitative probabilities.

Property boom of 1991-2007

The banks lent huge amounts of money to property developers based on—in retrospect—gross over-estimates of property values, and/or ignoring risks. The banks collapsed in the 2008/9 credit crunch.

In the context of analysing the property prices and latest crisis:

- How can we measure animal spirits?
- How can we leverage it for testing?

Measuring animal spirits

There are two varieties of confidence indices, the *consumer confidence index* and the *purchasing managers' index* (PMI).

Both are based on surveys, and represent respectively the propensity of consumers to go out and spend, and the propensity of businesses to invest.

In the United States, consumer confidence is measured by the Conference Board and by the University of Michigan. The main difference between the two surveys is in the time horizon: while the Conference Board polls households on their expectations over the next six months, the University of Michigan looks at expectations over the coming year.

On the other hand, purchasing manager expectations are assessed regionally: the Chicago PMI is widely regarded as the most representative of nationwide sentiment.

These indices are quite noisy though, and it is not immediately clear how they are related to animal spirits.

Simplest approach to model animal spirits

The simplest way to model agents' attitude is to introduce:

- finite number of states
- with intuitive interpretation.

The simplest dynamic model is

- finite state Markov Chain

Hidden Markov Models (HMMs)

This family of models assumes two processes:

- Observable process O_t, e.g. log-returns of a property price index
- Hidden (not observable directly) *state process* q_t

The distribution of these processes in discrete time is given by:

- The hidden state process is a finite state Markov chain with transition matrix A
- Distribution parameters of O_t are functions of the state q_t
- The model is time homogeneous.

For simplicity of exposition we assume:

- O_t are Gaussian given q_t, and are conditionally independent of the values O_s and q_s for $s \neq t$
- q_t takes values in positive integers

These type of models became very popular in signal processing filed in the seventies.

In finance they are widely used to model regimes and regime-switching behaviour.

Maximum likelihood

The model is parameterized by $\lambda = (\pi, A, \theta)$ where:

- π is the initial distribution of the hidden state
- A is the transition matrix
- θ consists of distribution parameters of O_t for every hidden state

To estimate the parameters of the model we shall adopt the maximum likelihood method, so we find parameter values $\hat{\lambda}$ that maximise the likelihood function of the observed series:

$$\mathcal{L}(\lambda; O) = \mathbb{P}\left[O|\,\lambda\right] = \sum_{q} \mathbb{P}\left[O|\,q, \lambda\right] \mathbb{P}\left[q|\,\lambda\right] \tag{1}$$

where we sum over all possible realizations of the hidden state sequence. Note that we can express this as an expectation w.r.t the hidden sequence:

$$\mathbb{P}\left[O|\,\lambda\right] = \sum_{q} \mathbb{P}\left[O|\,q, \lambda\right] \mathbb{P}\left[q|\,\lambda\right] = \mathbb{E}\left[\mathbb{P}\left[O|\,Q, \lambda\right]\right] \tag{2}$$

Direct maximisation is in this case rather difficult.

EM algorithm

The algorithm starts from some user-defined initial guess for the parameter values and then improves them iteratively. A single iteration consists of two steps:

Expectation First, given the parameter values from previous step λ^{i-1} we calculate the quantity:

$$\begin{aligned} Q(\lambda, \lambda^{i-1}) &= \mathbb{E}\left[\log \mathbb{P}\left[O, Q|\, \lambda\right]|\, O, \lambda^{i-1}\right] \\ &= \sum_{q} \log \mathbb{P}\left[O, q|\, \lambda\right] \mathbb{P}\left[O, q|\, \lambda^{i-1}\right] \end{aligned} \tag{3}$$

as a function of λ.

Maximisation Next, we find $\lambda^{i} = \arg\max_{\lambda} Q(\lambda, \lambda^{i-1})$, which will be used in the next iteration.

Dempster et.al (1977) proved that the algorithm converges to a local maximum and the parameter values that maximise (3) also maximise (1), hence the algorithm gives the required estimate.

Baum-Welch algorithm

We can write the probabilities $\mathbb{P}\left[O, q\middle|\, \lambda\right]$ as:

$$\mathbb{P}\left[O, q\middle|\, \lambda\right] = \pi_{q_1} b_{q_1}(O_1) \prod_{t=2}^{T} a_{q_{t-1}q_t} b_{q_t}(O_t), \tag{4}$$

where $a_{q_{t-1}q_t}$ is the transition probability from the state q_{t-1} to q_t and $b_{q_t}(O_t) = f(O_t; q_t)$ is the density of the observable process given that the system is in the state q_t. The Q function (3) becomes:

$$\begin{aligned} Q((\pi, A, \theta), \lambda) = &\sum_{q \in \mathcal{Q}} \log \pi_{q_1} \mathbb{P}\left[O, q\middle|\, \lambda\right] \\ &+ \sum_{q \in \mathcal{Q}} \sum_{t=2}^{T} \log a_{q_{t-1}q_t} \mathbb{P}\left[O, q\middle|\, \lambda\right] \\ &+ \sum_{q \in \mathcal{Q}} \sum_{t=1}^{T} \log b_{q_t}(O_t) \mathbb{P}\left[O, q\middle|\, \lambda\right], \end{aligned} \tag{5}$$

The key observation is to notice that for all three terms above, for every t we sum over the space of all paths of the hidden state process, whereas we are only concerned with the value at the given and directly preceding time.

When we consolidate these terms, we end up with a much more computation-friendly formula:

$$\begin{aligned} Q((\pi, A, \theta), \lambda) = &\sum_{i=1}^{N} \mathbb{P}\left[O, q_1 = i \middle| \lambda\right] \log \pi_i \\ &+ \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=2}^{T} \mathbb{P}\left[O, q_{t-1} = i, q_t = j \middle| \lambda\right] \log a_{ij} \\ &+ \sum_{i=1}^{N} \sum_{t=1}^{T} \mathbb{P}\left[O, q_t = i \middle| \lambda\right] \log b_i(O_t), \end{aligned} \tag{6}$$

where the only dependence on θ is in the b_i functions.

Because of the separability of Q with respect to different parameters, it is easy to calculate the maximum analytically using the constraints:

$$\begin{aligned} \sum_{j=1}^{N} \pi_j &= 1 \\ \sum_{j=1}^{N} a_{ij} &= 1, \quad \text{for every } i \end{aligned} \tag{7}$$

and the normal density function for b. The maximised values of parameters are given by:

$$\begin{aligned} \pi_i^* &= \frac{\mathbb{P}\left[O, q_1 = i \middle| \lambda\right]}{\mathbb{P}\left[O \middle| \lambda\right]} \\ a_{ij}^* &= \frac{\sum_{t=2}^{T} \mathbb{P}\left[O, q_{t-1} = i, q_t = j \middle| \lambda\right]}{\sum_{t=2}^{T} \mathbb{P}\left[O, q_{t-1} = i \middle| \lambda\right]} \\ m^*(i) &= \frac{\sum_{t=1}^{T} \mathbb{P}\left[O, q_t = i \middle| \lambda\right] O_t}{\sum_{t=1}^{T} \mathbb{P}\left[O, q_t = i \middle| \lambda\right]} \\ var^*(i) &= \frac{\sum_{t=1}^{T} \mathbb{P}\left[O, q_t = i \middle| \lambda\right] (O_t - m^*(i))^2}{\sum_{t=1}^{T} \mathbb{P}\left[O, q_t = i \middle| \lambda\right]} \end{aligned} \tag{8}$$

Forward-Backward procedure

To be able to use the equations above, we need an efficient way to calculate the quantities $\mathbb{P}\left[O|\,\lambda\right]$, $\mathbb{P}\left[O, q_t = i|\,\lambda\right]$ and $\mathbb{P}\left[O, q_{t-1} = i, q_t = j|\,\lambda\right]$. To achieve this, we will use the Forward-Backward procedure. First define the forward variables:

$$\alpha_t(i) = \mathbb{P}\left[O_1 O_2 \ldots O_t, Q_t = i|\,\lambda\right], \tag{9}$$

which can be calculated dynamically with starting condition:

$$\alpha_1(j) = \pi_j b_j(O_1) \tag{10}$$

and the induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}) \tag{11}$$

for every state j.

Analogously we define the backward variables:

$$\beta_t(i) = \mathbb{P}\left[O_{t+1}O_{t+2}\dots O_T \middle| Q_t = i, \lambda\right]. \tag{12}$$

The final value is defined arbitrarily by:

$$\beta_T(i) = 1, \tag{13}$$

and the backward induction is given by:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j). \tag{14}$$

We can use these precomputed values to efficiently calculate the quantities of interest. First, because we might end up in any state at the final time T, we have:

$$\mathbb{P}\left[O \middle| \lambda\right] = \sum_{i=1}^{N} \alpha_T(i). \tag{15}$$

By looking at the probabilistic definitions of α and β and using the Bayes formula we get:

$$\begin{aligned}\gamma_t(i) &= \mathbb{P}\left[O\middle|\, q_t = i, \lambda\right] \\ &= \frac{\mathbb{P}\left[O_1 \dots O_t, q_t = i\middle|\, \lambda\right] \mathbb{P}\left[O_{t+1} \dots O_T\middle|\, O_1 \dots O_t, q_t = i, \lambda\right]}{\mathbb{P}\left[O\middle|\, \lambda\right]} \\ &= \frac{\mathbb{P}\left[O_1 \dots O_t, q_t = i\middle|\, \lambda\right] \mathbb{P}\left[O_{t+1} \dots O_T\middle|\, q_t = i, \lambda\right]}{\sum_{j=1}^{N} \mathbb{P}\left[O, q_t = j\middle|\, \lambda\right]} \\ &= \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}\end{aligned} \tag{16}$$

where in the third line we used the fact that given q_t the observable values after time t, $O_{t+1} \dots O_T$ are independent of the earlier observations $O_1 \dots O_t$. Analogously we have the formula for the last missing ingredient:

$$\begin{aligned}\xi_t(i,j) &= \mathbb{P}\left[q_t = i, q_{t+1} = j\middle|\, O, \lambda\right] \\ &= \frac{\mathbb{P}\left[O, q_t = i, q_{t+1} = j\middle|\, \lambda\right]}{\mathbb{P}\left[O\middle|\, \lambda\right]} \\ &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\mathbb{P}\left[O\middle|\, \lambda\right]}.\end{aligned} \tag{17}$$

It is convenient to express the estimates (8) in terms of γ and ξ:

$$\begin{aligned} \pi_i^* &= \gamma_1(i) \\ a_{ij}^* &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \\ m^*(i) &= \frac{\sum_{t=1}^{T} \gamma_t(i) O_t}{\sum_{t=1}^{T} \gamma_t(i)} \\ var^*(i) &= \frac{\sum_{t=1}^{T} \gamma_t(i)(O_t - m^*(i))^2}{\sum_{t=1}^{T} \gamma_t(i)} \end{aligned} \tag{18}$$

Scaling

Unfortunately the induction described above is not numerically stable, because the terms α_t go to zero exponentially fast as t increases and get out of the scope of double-precision numbers used in computer systems. To avoid this we will introduce scaled versions of these coefficients $\hat{\alpha}_t(i) = \frac{\alpha_t(i)}{\sum_{j=1}^{N} \alpha_t(j)}$:

$$\begin{aligned} \bar{\alpha}_1(i) &= \alpha_1(i) \\ n_t &= \sum_{j=1}^{N} \bar{\alpha}_t(j) \\ \hat{\alpha}_t(i) &= \frac{\bar{\alpha}_t(i)}{n_t} \\ \bar{\alpha}_{t+1}(j) &= \left[\sum_{i=1}^{N} \hat{\alpha}_t(i) a_{ij} \right] b_j(O_{t+1}) \end{aligned} \tag{19}$$

It is easy to notice that the scaled values are given by:

$$\hat{\alpha}_t(i) = \frac{\alpha_t(i)}{N_t}, \tag{20}$$

where

$$N_t = \prod_{\tau=1}^{t} n_t \tag{21}$$

By using the equation above for $\hat{\alpha}_t(i)$ and the definition for induction (19) we can write:

$$\hat{\alpha}_t(i) = \frac{\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji} b_i(O_t)/N_t}{\sum_{k=1}^{N}\sum_{j=1}^{N} \alpha_{t-1}(j) a_{jk} b_k(O_t)/N_t} = \frac{\alpha_t(i)}{\sum_{k=1}^{N} \alpha_t(k)}, \tag{22}$$

which shows that our induction is indeed producing scaled variables as indicated above.

We use the same normalisation factors to calculate scaled backward variables:

$$\hat{\beta}_T(i) = \frac{1}{n_T}$$
$$\hat{\beta}_t(i) = \frac{1}{n_t} \sum_{j=1}^{N} a_{ij} \hat{\beta}_{t+1}(j) b_j(O_{t+1}) \tag{23}$$

And analogously we get:

$$\hat{\beta}_t(i) = \frac{\beta_t(i)}{M_t}, \tag{24}$$

where

$$M_t = \prod_{\tau=t}^{T} n_t \tag{25}$$

These can be now directly used to calculate the needed probabilities. First because $\hat{\alpha}$ are normalized, the identity holds:

$$\sum_{i=1}^{N} \hat{\alpha}_T(i) = \frac{\sum_{i=1}^{N} \alpha_T(i)}{N_T} = 1, \tag{26}$$

thus by (15) we get $\mathbb{P}\left[O|\,\lambda\right] = N_T$. As already mentioned, this quantity is outside of the range of double numbers, but we can efficiently calculate the logarithm (the log-likelihood function):

$$\log \mathbb{P}\left[O|\,\lambda\right] = \sum_{t=1}^{T} \log n_t \tag{27}$$

To calculate ξ_t we will substitute α_t for $\hat{\alpha}_t N_t$ and β_t for $\hat{\beta}_t M_t$ in (17):

$$\begin{aligned}
\xi_t(i,j) &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\mathbb{P}\left[O|\,\lambda\right]} \\
&= \frac{\hat{\alpha}_t(i) a_{ij} b_j(O_{t+1}) \hat{\beta}_{t+1}(j) N_t M_{t+1}}{\mathbb{P}\left[O|\,\lambda\right]} \\
&= \frac{\hat{\alpha}_t(i) a_{ij} b_j(O_{t+1}) \hat{\beta}_{t+1}(j) N_T}{\mathbb{P}\left[O|\,\lambda\right]} \\
&= \hat{\alpha}_t(i) a_{ij} b_j(O_{t+1}) \hat{\beta}_{t+1}(j),
\end{aligned} \tag{28}$$

where we used the fact that $N_t M_{t+1} = N_T$ and that $\mathbb{P}\left[O|\,\lambda\right] = N_T$. The γ coefficients may be computed analogously:

$$\begin{aligned}
\gamma_t(i) &= \frac{\alpha_t(i) \beta_t(i)}{\mathbb{P}\left[O|\,\lambda\right]} \\
&= \frac{\hat{\alpha}_t(i) \hat{\beta}_t(i) N_t M_t}{\mathbb{P}\left[O|\,\lambda\right]} \\
&= \hat{\alpha}_t(i) \hat{\beta}_t(i) n_t,
\end{aligned} \tag{29}$$

because $N_t M_t = n_t N_T = n_t \mathbb{P}\left[O|\,\lambda\right]$. We can plug these values to calculate the parameter estimates according to (18), just as in the unscaled case.

Viterbi algorithm - the most likely path of the hidden process

First define

$$\delta_t(i) = \max_{q_1 q_2 \cdots q_{t-1}} \log \mathbb{P}\left[q_1 q_2 \ldots q_{t-1}, q_t = i, O_1 O_2 \ldots O_t | \, \lambda\right], \tag{30}$$

that is, $\delta_t(i)$ denotes the highest log-likelihood along a single path up to time t that ends in the state i. The maximum log-likelihood of both hidden and observable paths is given by:

$$P^* = \max_{1 \leq i \leq N} [\delta_t(i)] \tag{31}$$

δ may be efficiently computed by the following induction:

$$\begin{aligned}\delta_1(i) &= \log \pi_i + \log b_i(O_1) \\ \delta_t(j) &= \max_{1\leq i\leq N}[\delta_{t-1}(i) + \log a_{ij}] + \log b_j(O_t).\end{aligned} \tag{32}$$

We also need another set of variables ψ to track which previous state was chosen in the maximisation above, at every time and state j:

$$\begin{aligned}\psi_1(i) &= 0 \\ \psi_t(j) &= \arg\max_{1\leq i\leq N}[\delta_{t-1}(i) + \log a_{ij}].\end{aligned} \tag{33}$$

Having calculated these numbers, the last element of the most likely hidden path is given by:

$$q_T^* = \arg\max_{1\leq i\leq N}[\delta_t(i)]. \tag{34}$$

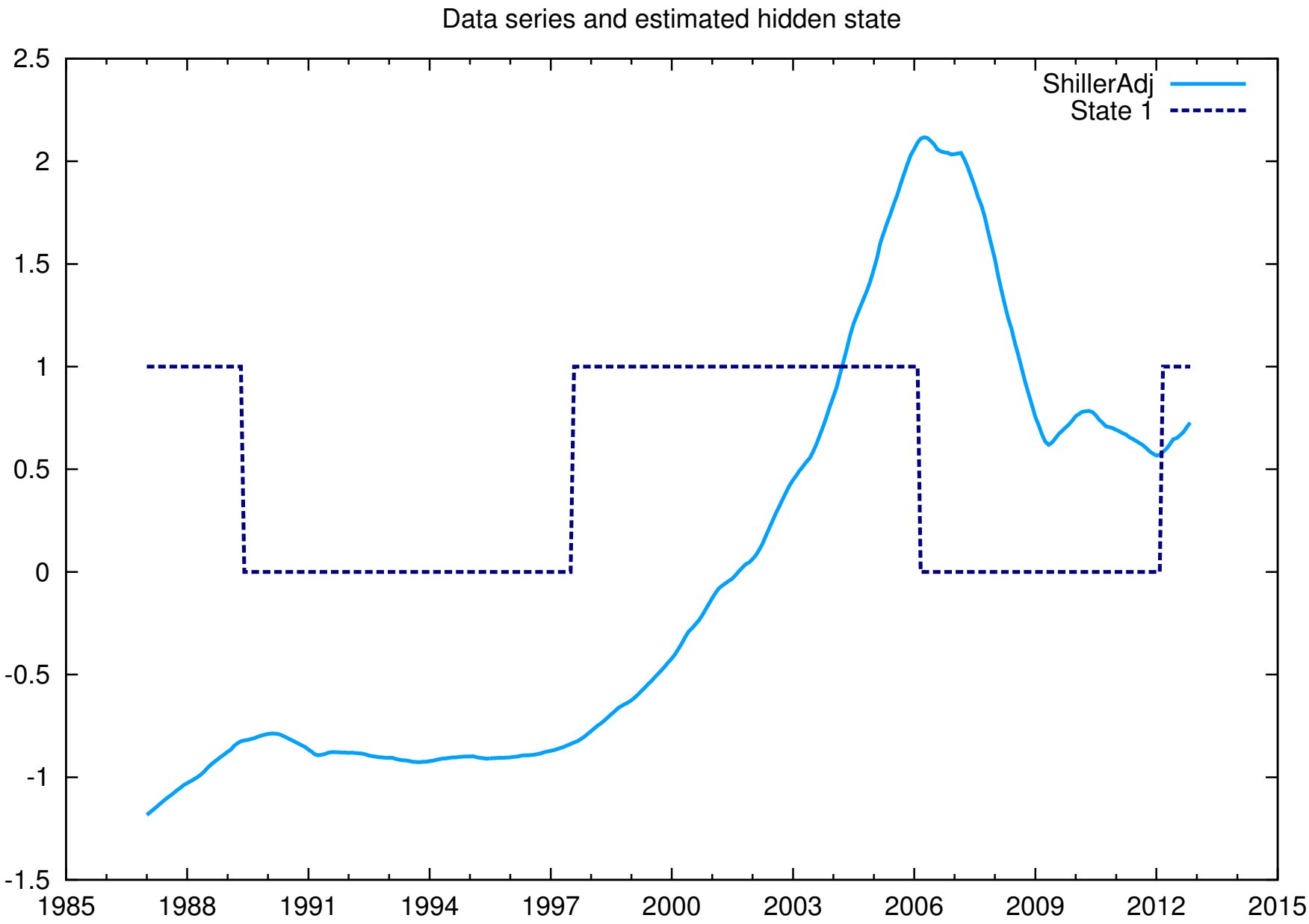
To calculate the most likely hidden state at earlier times we use the backtracking method:

$$q_t^* = \psi_{t+1}(q_{t+1}^*). \tag{35}$$

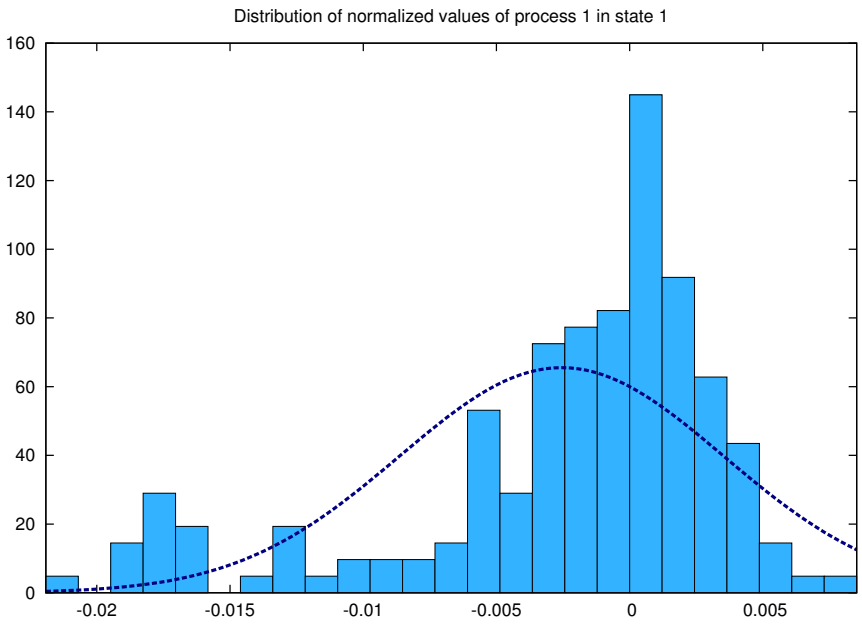
Continuous-time extension

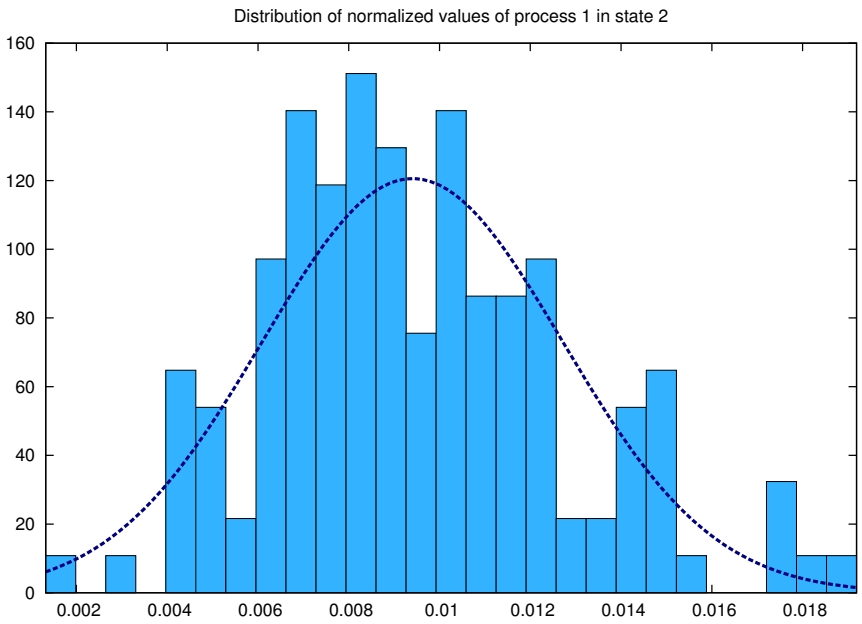
Most models in mathematical-finance are in continuous time, mainly for easier tractability. However, real world data is only available at discrete time points (monthly in case of the Shiller index) and hence the statistics literature deals with discrete time series.

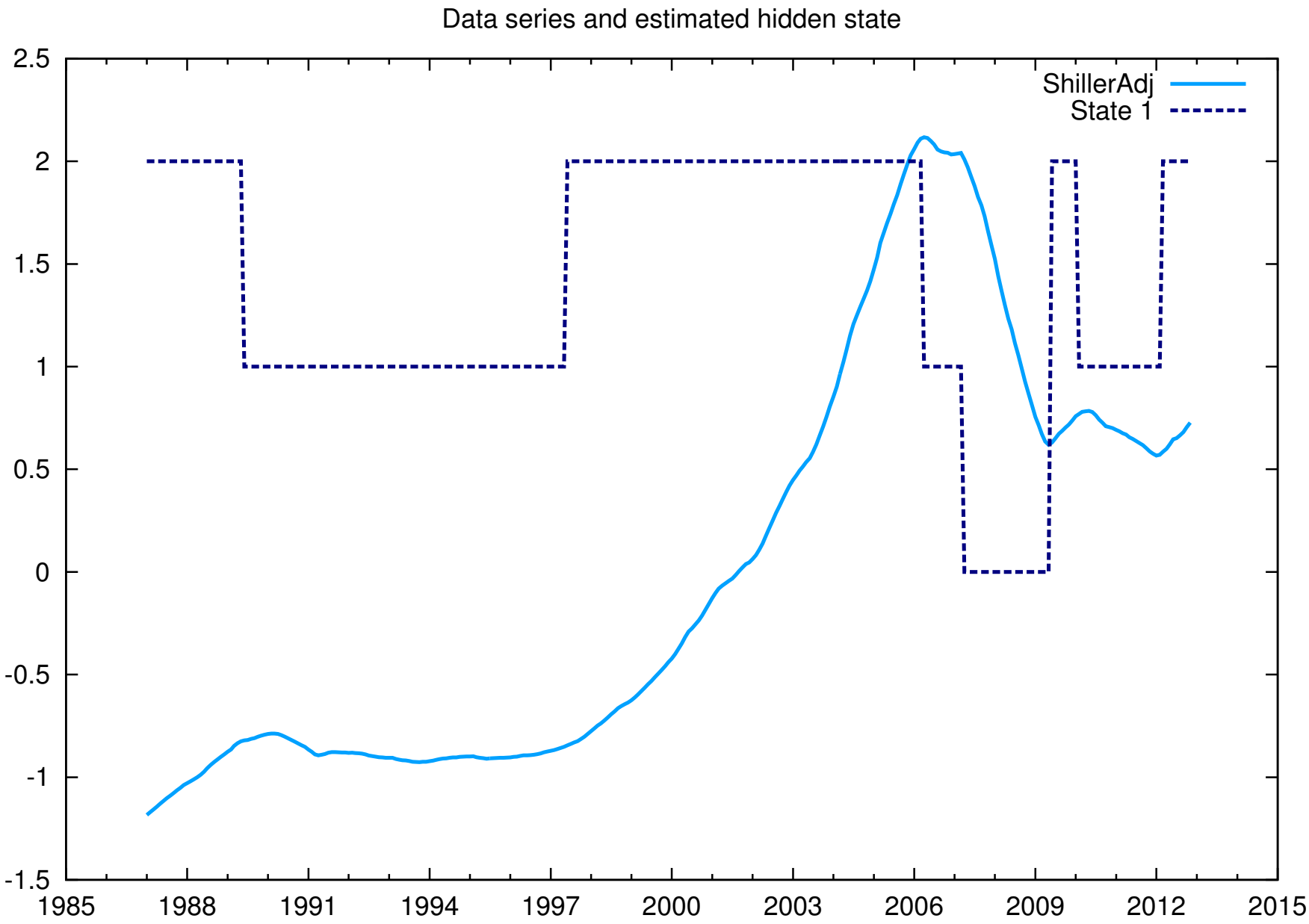
To be able to bridge these two fields easily and without introducing additional error terms, we only work with price processes that can be discretized exactly. The state process is assumed to only change value on the observation dates, so the continuous-time version has right-continuous piecewise-constant paths.

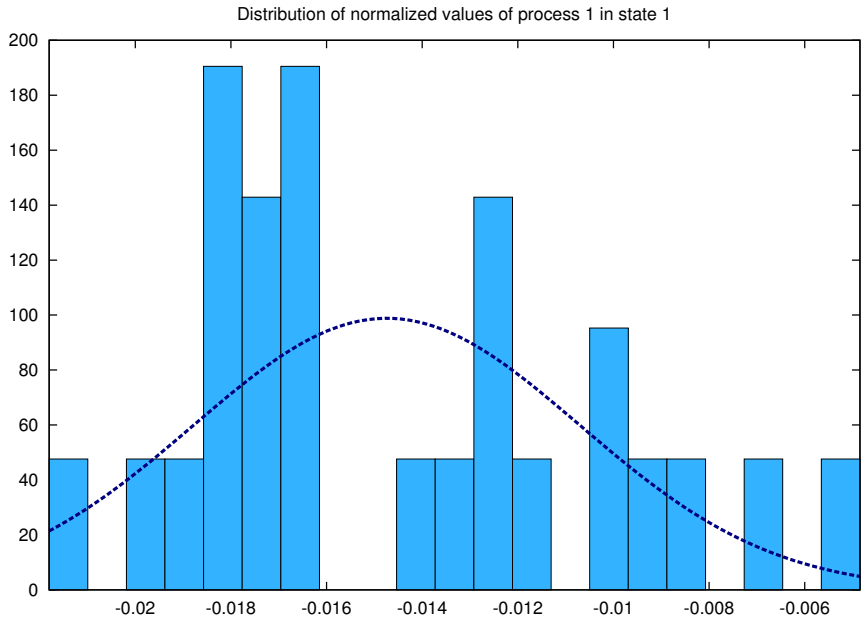

Data series and estimated hidden state
ShillerAdj
State 1
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
1985
1988
1991
1994
1997
2000
2003
2006
2009
2012
2015

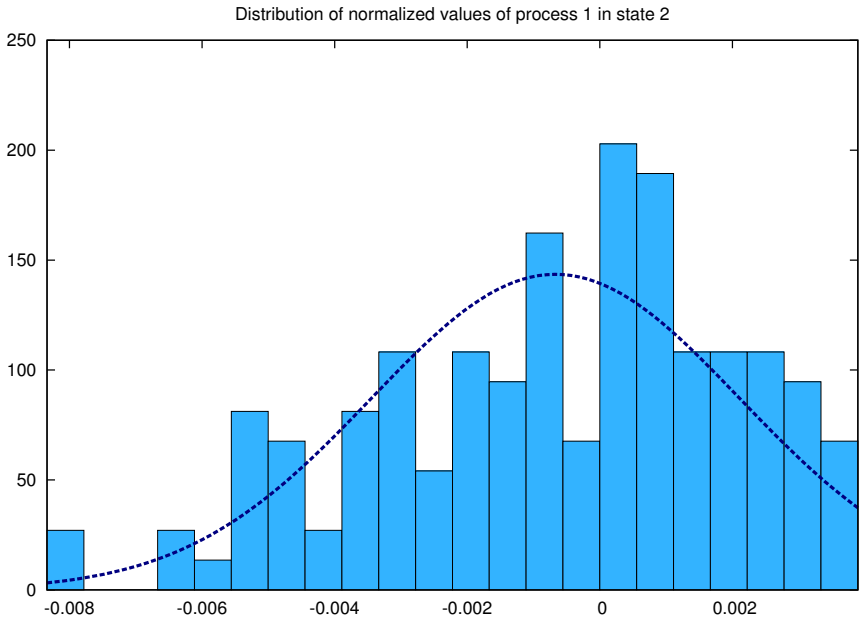
Histograms of realized observations in different states:

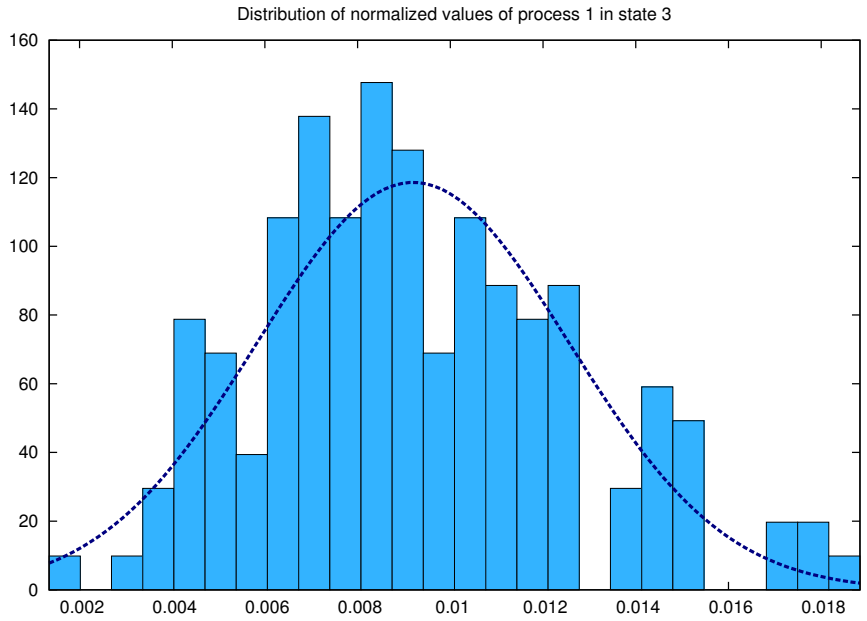


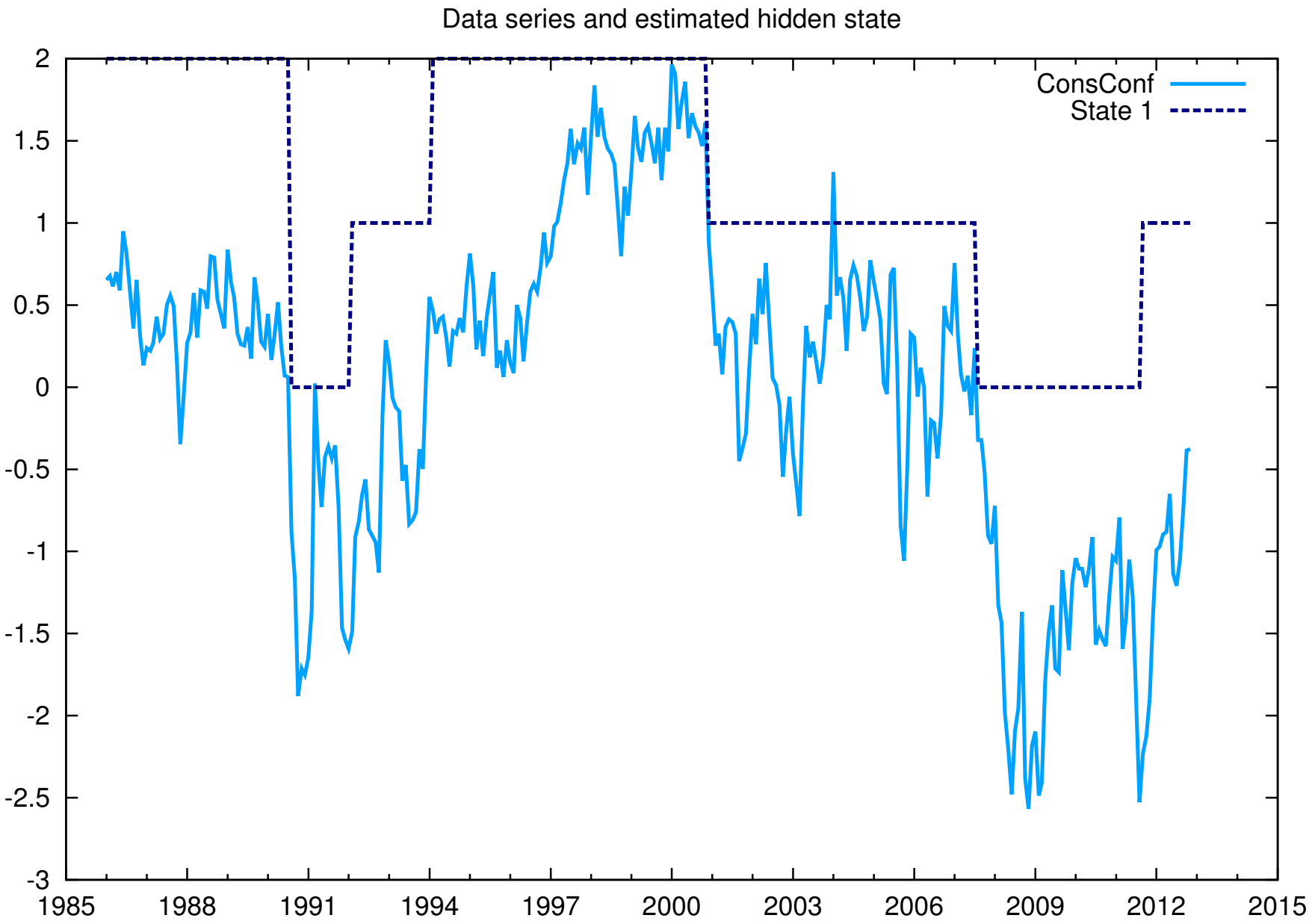


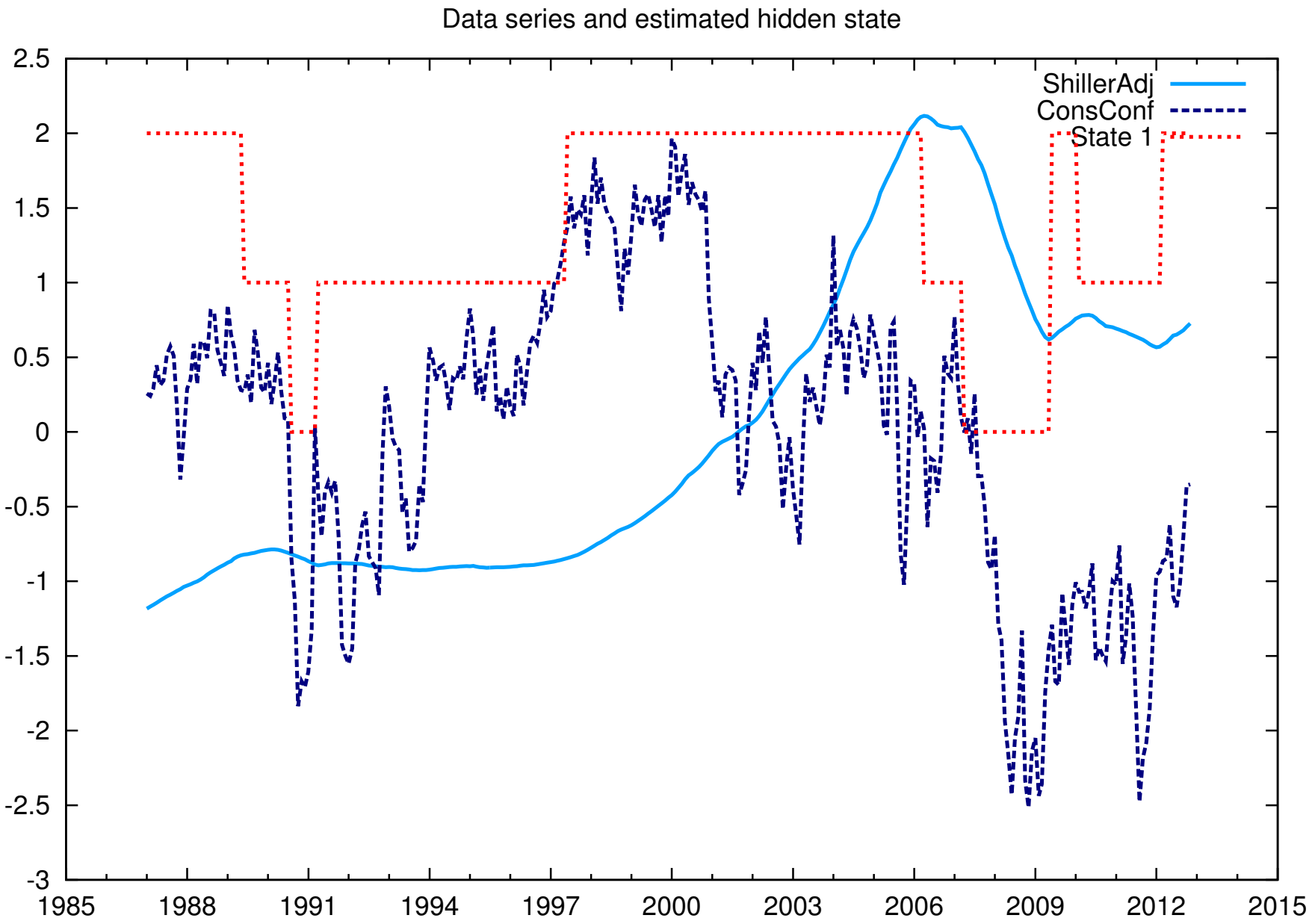

Data series and estimated hidden state
ShillerAdj
State 1
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
1985
1988
1991
1994
1997
2000
2003
2006
2009
2012
2015


Distribution of normalized values of process 1 in state 1
200
180
160
140
120
100
80
60
40
20
0
-0.02
-0.018
-0.016
-0.014
-0.012
-0.01
-0.008
-0.006


Distribution of normalized values of process 1 in state 2
250
200
150
100
50
0
-0.008
-0.006
-0.004
-0.002
0
0.002


Distribution of normalized values of process 1 in state 3
160
140
120
100
80
60
40
20
0
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0.018


Data series and estimated hidden state
ConsConf
State 1
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
1985
1988
1991
1994
1997
2000
2003
2006
2009
2012
2015


Data series and estimated hidden state
ShillerAdj
ConsConf
State 1
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
1985
1988
1991
1994
1997
2000
2003
2006
2009
2012
2015

Probability forecasts

Traditionally everyone used *point* forecasts, e.g. *tomorrow the temperature will be 20 degrees.*

We prefer to use probability forecasts, e.g. *tomorrow, the temperature is distributed by –given distribution–.*

Point forecasts (the expected value) contain not enough information for many purposes, e.g.

- Calculating probability of temperature falling below certain level
- Analysing variability of temperature
- Performing any optimisation over all possible values of temperature, etc.

Moreover, it might be the case that the temperature of 20 degrees will never happen. Imagine a bi-modal distribution with peaks at 15 degrees (if the wind is from the north) and at 25 degrees (otherwise), with zero mass at 20 degrees.

One-period forecasts with HMMs

Given the history of the process up to time T, the distribution of the observable at time $T+1$ is given by:

$$\begin{aligned}\mathbb{P}\left[O_{T+1}=o\middle|\, O_1\dots O_T\right] &= \frac{\mathbb{P}\left[O_1\dots O_T, O_{T+1}=o\right]}{\mathbb{P}\left[O_1\dots O_T\right]} \\ &= \frac{\sum_{i=1}^{N}\alpha_T(i)\sum_{j=1}^{N}a_{ij}b_j(o)}{\sum_{i=1}^{N}\alpha_T(i)},\end{aligned} \tag{36}$$

where $\alpha_T(i)$ is defined as $\alpha_t(i)=\mathbb{P}\left[O_1O_2\dots O_t, Q_t=i\middle|\,\lambda\right]$.

It is easy to notice that the forecast distribution is a mixture of the base distributions (Gaussian in our case):

$$\mathbb{P}\left[O_{T+1}=o\middle|\, O_1\dots O_T\right]=\sum_{j=1}^{N}\psi_T(j)b_j(o), \tag{37}$$

with

$$\psi_T(j)=\frac{\sum_{i=1}^{N}\alpha_T(i)a_{ij}}{\sum_{i=1}^{N}\alpha_T(i)} \tag{38}$$

The forecast distribution of the hidden state is:

$$\begin{aligned}\mathbb{P}\left[Q_{T+1}=q|\, O_1\dots O_T\right] &= \frac{\mathbb{P}\left[O_1\dots O_T, Q_{T+1}=q\right]}{\mathbb{P}\left[O_1\dots O_T\right]} \\ &= \frac{\sum_{i=1}^{N}\alpha_T(i)a_{iq}}{\sum_{i=1}^{N}\alpha_T(i)}.\end{aligned} \tag{39}$$

Note that in all the formulas above we can substitute α_T by the scaled version $\hat{\alpha}_T$, because the normalisation factors cancel out.

Verification of forecast distribution

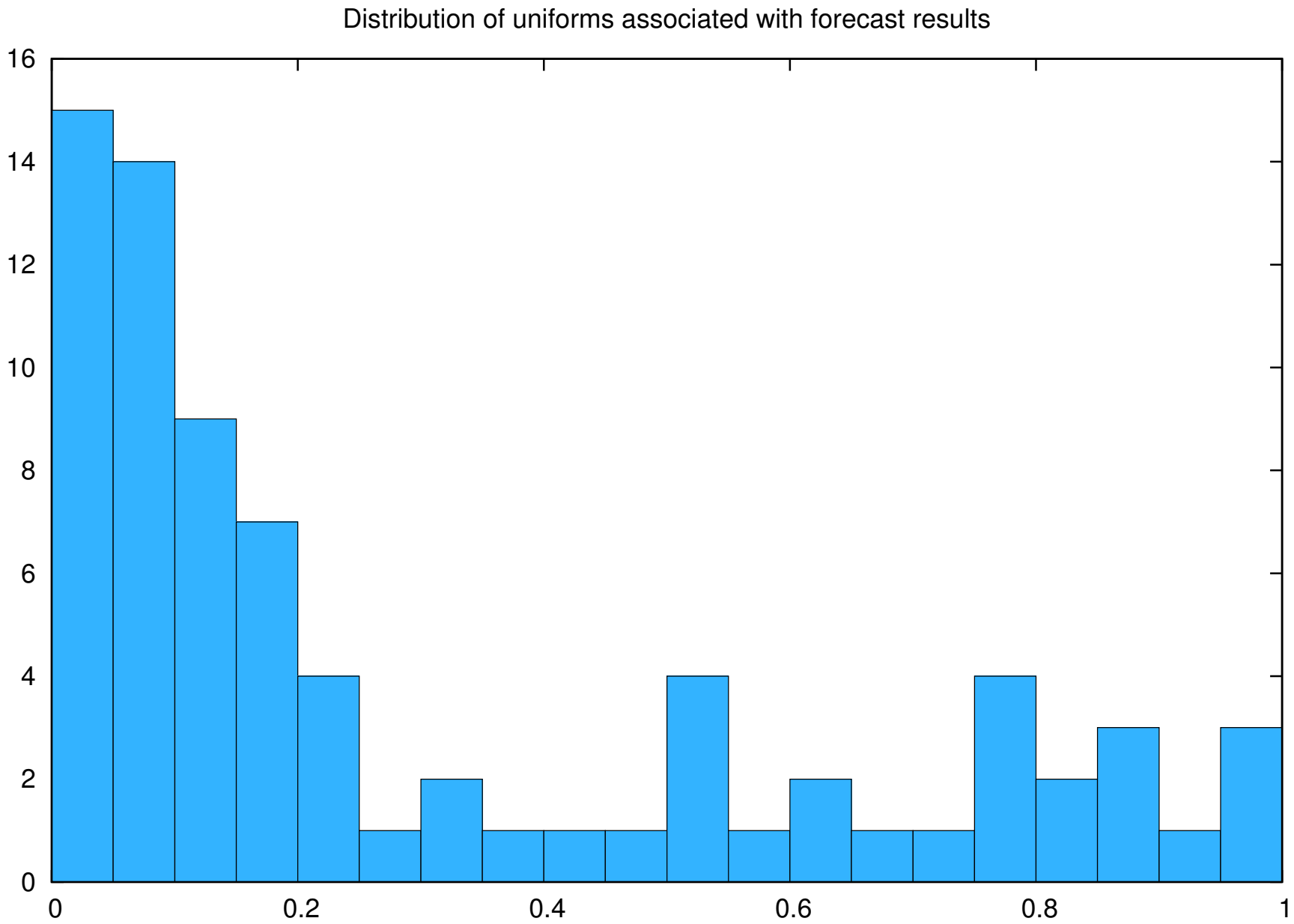
Proposition 1. *Assume that at each point in time $T_0 \leq t \leq T$ we are given a distribution forecast for the value of process X:*

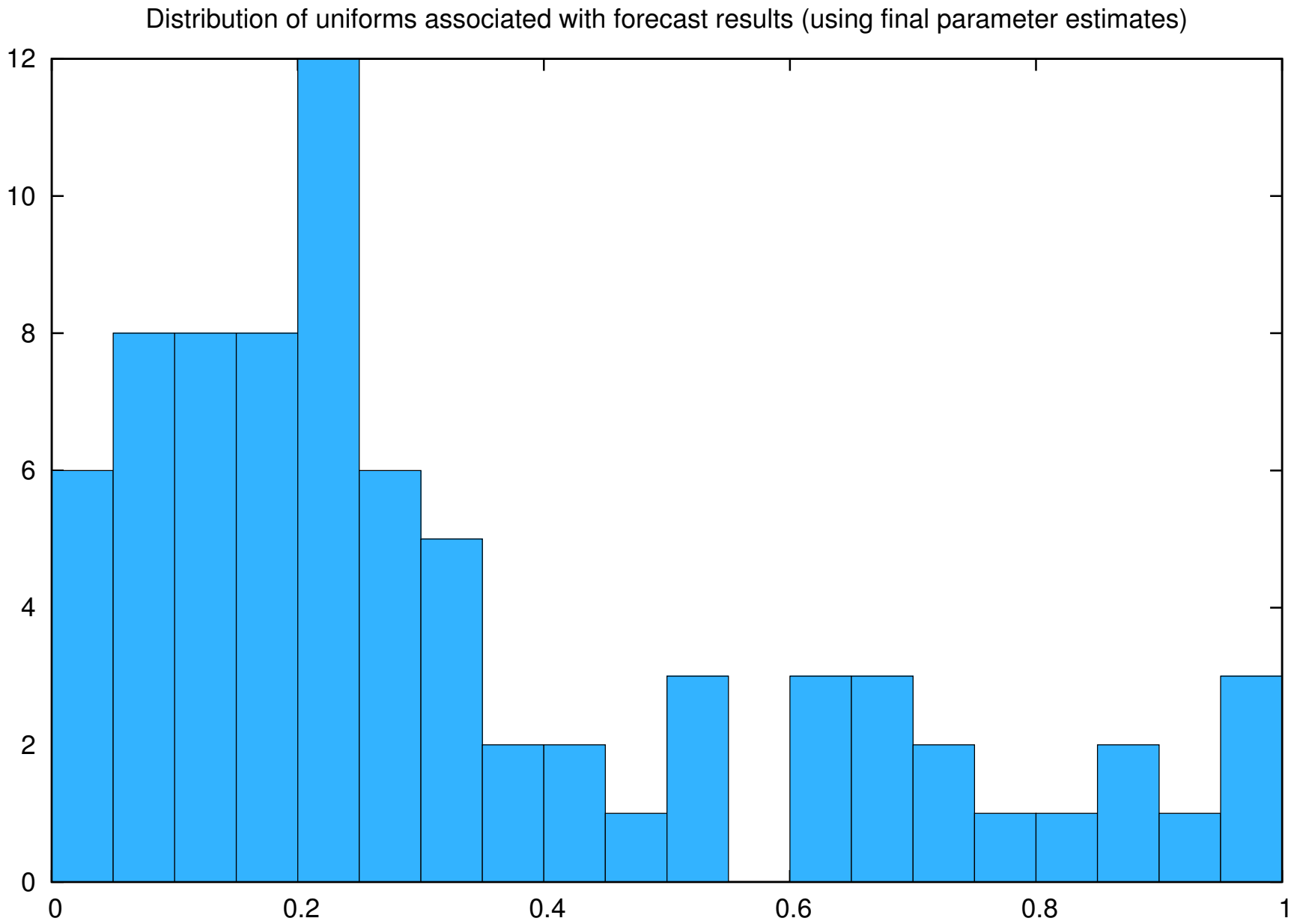
$$F_t(x) = \mathbb{P}\left[X_t < x \middle| \mathcal{G}_{t-1}\right], \tag{40}$$

where for simplicity we assume that F_t is a continuous and strictly increasing function (as it is the case for mixture Gaussian distribution) and $\mathcal{G}_{t-1}$ is the sigma algebra representing the information available at time $t-1$. Let $Y_t = F_t(X_t)$ for every t.

Then the random variables Y_t are uniformly distributed on $(0,1)$ and are mutually independent.

The proof is rather elementary, uses the existence and properties of the inverse of $F_t(x)$, as well as the tower property of conditional expectations.


Distribution of uniforms associated with forecast results
16
14
12
10
8
6
4
2
0
0
0.2
0.4
0.6
0.8
1


Distribution of uniforms associated with forecast results (using final parameter estimates)
12
10
8
6
4
2
0
0
0.2
0.4
0.6
0.8
1

Multivariate models with lags (inspired by VAR models)

It is often observed that the variable of interest is correlated with a lagged version of a different variable.

Assume that X_t is the variable of interest and Y_t is another observable variable, such that Y_{t-1} is correlated with X_t. The observable process at time t is composed of both X_t and Y_{t-1}:

$$O_t = [X_t Y_{t-1}]^{\mathrm{T}}. \tag{41}$$

To forecast X_{T+1} at time T we can use the available observation Y_T in a natural way:

$$\begin{aligned} \mathbb{P}\left[X_{T+1} = x \middle| X_1 \dots X_T, Y_0 \dots Y_T\right] &= \frac{\mathbb{P}\left[X_{T+1} = x, X_1 \dots X_T, Y_0 \dots Y_T\right]}{\mathbb{P}\left[X_1 \dots X_T, Y_0 \dots Y_T\right]} \\ &= \frac{\sum_{i=1}^{N} \alpha_T(i) \sum_{j=1}^{N} a_{ij} b_j(x, Y_T)}{\sum_{i=1}^{N} \alpha_T(i) \sum_{j=1}^{N} a_{ij} b_j^Y(Y_T)}, \end{aligned} \tag{42}$$

where $b_j^Y(Y_T)$ is the marginal distribution of Y_T. By applying the definition of conditional probability $b_l(x, Y_T) = b_l(x|Y_T) b_l^Y(Y_T)$ we can transform the

equation above to arrive at:

$$\mathbb{P}\left[X_{T+1} = x \middle| X_1 \dots X_T, Y_0 \dots Y_T\right] = \sum_{l=1}^{N} \psi_{T+1}^{l}(Y_T) b_l(x|Y_T), \tag{43}$$

where

$$\psi_{T+1}^{l}(Y_T) = \frac{\sum_{i=1}^{N} \alpha_T(i) a_{il} b_l^Y(Y_T)}{\sum_{i=1}^{N} \alpha_T(i) \sum_{j=1}^{N} a_{ij} b_j^Y(Y_T)} \tag{44}$$

The forecast is thus a mixture of distributions.

Moreover, if X and Y are jointly-normal distributed in every state i with parameters:

$$\begin{bmatrix} X \\ Y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_i^X \\ \mu_i^Y \end{bmatrix}, \begin{bmatrix} \Sigma_i^X & \Sigma_i^{XY} \\ \Sigma_i^{YX} & \Sigma_i^Y \end{bmatrix} \right), \tag{45}$$

then Y is normally distributed $Y \sim \mathcal{N}(\mu_i^Y, \Sigma_i^Y)$, and X, conditionally given Y, is also normally distributed with

$$X_{T+1}|Y_T = y \sim \mathcal{N}\left(\mu_i^X + \Sigma_i^{XY}(\Sigma_i^Y)^{-1}(y - \mu_i^Y), \Sigma_i^X - \Sigma_i^{XY}(\Sigma_i^Y)^{-1}\Sigma_i^{YX}\right). \tag{46}$$

As a result the forecast of X is given by a mixture of Gaussians.

It's not finished yet!

Further work:

1. Analyse different sets of explanatory data
2. Look at hand-crafted crash scenarios
3. Explore different distribution families