State Space Models

Hans-Peter Höllwirth

Master Project Report
Barcelona Graduate School of Economics
Master Degree in Data Science
2017

Contents

1 Introduction

2 State Space Models

State space models consist of two set of data:

1. A series of **latent states** $\{x_t\}_{t=1}^T$ (with $x_t \in \mathcal{X}$) that forms a Markov chain. Thus, x_t is independent of all past states but x_{t-1}.
2. A set of **observations** $\{y_t\}_{t=1}^T$ (with $y_t \in \mathcal{Y}$) where any observation y_t only depends on its latent state x_t. In other words, an observation is a noisy representation of its underlying state.

Note that if the state space $\mathcal{X}$ and the observation state $\mathcal{Y}$ are both discrete sets, the state space model reduces to a Hidden Markov Model.

The relation between the latent states and observations can be summarized by two probability distributions:

1. The **transition density** from the current state to a new state $p(x_{t+1}|x_t, \boldsymbol{\theta})$.
2. The **measurement density** for an observation given the latent state $p(y_t|x_t, \boldsymbol{\theta})$.

Here, $\boldsymbol{\theta} \in \Theta$ denotes the parameter vector of a state space model.

2.1 Local Level Model

Arguably, the simplest state space model is the (univariate) local level model. It has the following form:

$$\begin{aligned} \text{observation:} \quad & y_t = x_t + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma_\epsilon^2) \\ \text{state:} \quad & x_{t+1} = x_t + \eta_t, \quad \eta_t \sim N(0, \sigma_\eta^2) \end{aligned}$$

with some initial state $x_1 \sim N(a_1, P_1)$. All noise elements, i.e. all ϵ_t's and η_t's, are both mutually independent and independent from the initial state x_1. Assuming that we know a_1 and P_1, the model is fully specified by the following vector of parameters:

$$\boldsymbol{\theta} = [\sigma_\eta^2, \sigma_\epsilon^2]^T$$

Note that in the case of noise-free observations (i.e. $\sigma_\eta^2 = 0$), the model reduces to a pure random-walk. Likewise, if $\sigma_\epsilon^2 = 0$, the observations $\{y_t\}_{t=1}^T$ are a white noise representation of a some value x_1.
The transition and measurement density of the local level model are simple to deduce:

$$\begin{aligned} p(x_{t+1}|x_t, \boldsymbol{\theta}) &\sim N(x_t, \sigma_\epsilon^2) \\ p(y_t|x_t, \boldsymbol{\theta}) &\sim N(x_t, \sigma_\eta^2) \end{aligned}$$

2.2 Latent State Inference

Often, the main objective in state space models is to infer the latent state from observations. Let $\mathcal{I}_t$ denote the set of observed values up to time t:

$$\mathcal{I}_t = \{y_1, y_2, \ldots, y_t\}$$

Then information about the latent state x_t can be summarized by the following two probability distributions:

1. The **prediction density**, $p(x_t|\mathcal{I}_{t-1}, \boldsymbol{\theta})$, gives the probability of x_t given past observations $\mathcal{I}_{t-1}$.

2. The **filtering density**, $p(x_t|\mathcal{I}_t, \boldsymbol{\theta})$, gives the probability of x_t given the current and past observations $\mathcal{I}_t$.

The prediction and filtering densities are recursively related. Given the filtering density for state x_{t-1}, the prediction density for state x_t is

$$p(x_t|\mathcal{I}_{t-1}, \boldsymbol{\theta}) = \int p(x_t|x_{t-1}, \boldsymbol{\theta}) p(x_{t-1}|\mathcal{I}_{t-1}, \boldsymbol{\theta}) dx_{t-1}$$

where the first term in the integral is the transition density from x_{t-1} to x_t, and the second term is the filtering density from before. Likewise, given the prediction density for state x_t, the filtering density for x_t is

$$p(x_t|\mathcal{I}_t, \boldsymbol{\theta}) = \int p(x_t|x_{t-1}, \boldsymbol{\theta}) p(x_{t-1}|\mathcal{I}_{t-1}, \boldsymbol{\theta}) dx_{t-1}$$

2.3 Parameter Inference

Assuming a particular state space model, another common objective is is to infer the model parameters from observations. This is usually achieved via **maximum likelihood estimation**. The log-likelihood of the observations for a given parameter vector $\boldsymbol{\theta}$ is the

product of the conditional densities of observations, given all previous observations:

$$\begin{aligned}\log \mathcal{L}(\boldsymbol{\theta}) &= \log \prod_{t=1}^{T} p(y_t|\mathcal{I}_{t-1}, \boldsymbol{\theta}) \\ &= \sum_{t=1}^{T} \log p(y_t|\mathcal{I}_{t-1}, \boldsymbol{\theta}) \\ &= \sum_{t=1}^{T} \log \int p(y_t|x_t, \boldsymbol{\theta}) p(x_t|\mathcal{I}_{t-1}, \boldsymbol{\theta}) dx_t\end{aligned} \tag{2.1}$$

The decomposition of the observation densities into measurement density and prediction density, however, makes the maximization problem analytically intractable for most state space models.

3 Filtering

The objective of filtering is to update our knowledge of the system each time a new observation y_t is brought in. That is, we want to find an estimate of the latent process $x_{0:t}$, given all observations $\mathcal{I}_t = y_{0:t}$:

$$x_{0:t}|\mathcal{I}_t$$

Note that the joint distribution of the latent process conditioned on the observations can be decomposed in a recursive form:

$$p(x_{0:t}|\mathcal{I}_t) = \Big[\frac{p(y_t|x_t)p(x_t|x_{t-1})}{p(y_t|\mathcal{I}_{t-1})}\Big]p(x_{0:(t-1)}|\mathcal{I}_{t-1})$$

This form allows for updating our knowledge of the system in an online fashion. This has significant computational advantages: We do not need to keep the whole time series in memory and we can simply update our knowledge once we observe a new y_t. Unfortunately, in many state space models the normalization term $p(y_t|\mathcal{I}_{t-1})$ is analytically intractable. One notable exception are linear Gaussian state space models such as the local level model.

3.1 Kalman Filter

State space models in which both the latent states $\{x_t\}_{t=1}^T$ and the observations $\{y_t\}_{t=1}^T$ have linear dependencies and are normally distributed, the joint distribution and of the latent process $p(x_{0:t}|y_{0:t})$ is also Gaussian (and so are the prediction and filtering density). The inference problem can be analytically solved by using standard results of multivariate Gaussian marginal and conditional distributions.

The Kalman filter, however, uses a more efficient way to infer the latent states. The filter recursively computes the Gaussian prediction density $p(x_t|\mathcal{I}_{t-1}, \boldsymbol{\theta}) = N(\mu_{t|t-1}, \Sigma_{t|t-1})$ and filtering density $p(x_t|\mathcal{I}_t, \boldsymbol{\theta}) = N(\mu_{t|t}, \Sigma_{t|t})$ by obtaining their respective mean and covariance. This method has the big advantage that it does not need all observations to be kept in memory and can easily update the system whenever a new observation is made. Consider the multivariate generalization of the univariate local level model from before:

$$\begin{aligned} \text{observation:} \quad & \boldsymbol{y}_t = \boldsymbol{x}_t + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim N(\mathbf{0}, \Sigma_\epsilon) \\ \text{state:} \quad & \boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim N(\mathbf{0}, \Sigma_\eta) \end{aligned}$$

For the case of this model, the mean and (co)variance of the prediction and filtering density are updated as follows:

1. The **prediction step** obtains the *a priori* estimate of $\boldsymbol{x}_t$, given $\mathcal{I}_{t-1}$ (using the *a posteriori* estimate with $t-1$).

$$\begin{aligned} \boldsymbol{\mu}_{t|t-1} &= \boldsymbol{\mu}_{t-1|t-1} \\ \Sigma_{t|t-1} &= \Sigma_{t-1|t-1} + \Sigma_\eta \end{aligned} \tag{3.1}$$

2. The **update step** combines a new observation $\boldsymbol{y}_t$ with the *a priori* estimate to obtain an improved *a posteriori* estimate.

$$\begin{aligned} \boldsymbol{\mu}_{t|t} &= \boldsymbol{\mu}_{t|t-1} + K_t \boldsymbol{v}_t \\ \Sigma_{t|t} &= \Sigma_{t|t-1}(1 - K_t) \end{aligned} \tag{3.2}$$

   where $\boldsymbol{v}_t = \boldsymbol{y}_t - \boldsymbol{\mu}_{t|t-1}$ denotes the difference between prediction and observation and $K_t = \Sigma_{t|t-1}(\Sigma_{t|t-1} + \Sigma_\epsilon)^{-1}$ denotes the *Kalman gain*. It determines how much the new observation affects the updated prediction.

Note that to start the recursion, we need an initial state density with $\boldsymbol{\mu}_1$ and Σ_1.

3.1.1 Algorithm

In the following version of the algorithm we assume $\boldsymbol{\mu}_1$ and Σ_1 to be known. There are various ways to initialize the algorithm when $\boldsymbol{\mu}_1$ and Σ_1 are unknown, however, these methods are beyond the scope of this project.

Algorithm 1 (Local Level) Kalman filter

procedure KALMANFILTER($\mathcal{I}_T, \boldsymbol{\theta}, \boldsymbol{\mu}_1, \Sigma_1$)
  $\boldsymbol{\mu}_{1|0} \leftarrow \boldsymbol{\mu}_1$ ▷ initialization
  $\Sigma_{1|0} \leftarrow \Sigma_1$
  for t `in` $1:T$ **do**
    $\boldsymbol{v}_t = \boldsymbol{y}_t - \boldsymbol{\mu}_{t|t-1}$
    $K_t = \Sigma_{t|t-1}(\Sigma_{t|t-1} + \Sigma_\epsilon)^{-1}$
    $\boldsymbol{\mu}_{t|t} = \boldsymbol{\mu}_{t|t-1} + K_t \boldsymbol{v}_t$ ▷ update step
    $\Sigma_{t|t} = \Sigma_{t|t-1}(1 - K_t)$
    $\boldsymbol{\mu}_{t+1|t} \leftarrow \boldsymbol{\mu}_{t|t}$ ▷ prediction step
    $\Sigma_{t+1|t} \leftarrow \Sigma_{t|t} + \Sigma_\eta$
  return $\{\boldsymbol{\mu}_{t|t}\}$, $\{\Sigma_{t|t}\}$

Algorithm 1 describes the recursive update of the prediction and filtering density. For our purpose, the algorithm returns the filtering densities. The sequence of filtering means

$\{\boldsymbol{\mu}_{t|t}\}$ is the best possible prediction for the latent states $\{\boldsymbol{x}_t\}$. The sequence of covariances $\{\Sigma_{t|t}\}$ can be used to obtain confidence intervals for these estimates.

Note that the algorithm needs to invert the matrix $F_t = \Sigma_{t|t-1} + \Sigma_\epsilon$ at each iteration. For large dimensions of the states/observations, this can slow down this version of the Kalman filter significantly.

3.1.2 Likelihood evaluation

The log-likelihood function for the linear Gaussian space model has the following (*prediction error decomposition*) form:

$$\log \mathcal{L}(\mathcal{I}_T, \boldsymbol{\theta}) = -\frac{Td}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{T}(\log|F_t| + v_t^T F_t^{-1} v_t)$$

Its elements F_t and v_t are routinely calculated by the Kalman filter and so the log-likelihood can be directly evaluated in the Kalman function. Note that F_t might not be singular for all $t = 1, \ldots, T$ for particular $\boldsymbol{\theta}$ values. Setting $\log \mathcal{L}(\mathcal{I}_T, \boldsymbol{\theta}) = -\infty$ in this case suffices for the purpose of maximum likelihood estimation.

3.1.3 Example

3.2 Particle Filter

If the state space model is not linear and Gaussian, both the joint distribution $p(x_{0:t}|\mathcal{I}_t)$ and the marginal distribution $p(x_t|\mathcal{I}_t)$ are usually not analytically solvable due to the intractability of the normalization constant $p(y_t|\mathcal{I}_{t-1})$. In this case we can only resort to sampling techniques to approximate these distribution densities.

Particle filtering methods (constituting a sub-class of *Sequential Monto Carlo* methods) approximate the prediction density $p(x_t|\mathcal{I}_{t-1}, \boldsymbol{\theta})$ and filtering density $p(x_t|\mathcal{I}_t, \boldsymbol{\theta})$ sequentially by using importance sampling techniques. There exist many different method variants, all of which involve two basic steps: simulating from the transition density $p(x_{t+1}|x_t, \boldsymbol{\theta})$ and evaluating the measurement density $p(y_t|x_t, \boldsymbol{\theta})$.

3.2.1 Sequential Importance Resampling (SIR)

One of the best-known particle filter methods is the *Sequential Importance Resampling (SIR)* algorithm by Gordon et al. (1993). Let P be the number of particles (= samples) per state. The algorithm recursively computes prediction and filtering particles:

1. The **prediction step** obtains a new prediction particle for each filtering particle by propagating the system, using the transition density:

$$x^i_{t|t-1} \sim p(x_t|x^i_{t-1|t-1}, \theta) \quad \text{for } i = 1, \ldots, P$$

2. The **filtering step** (or update step) computes the importance weight w^i_t of each prediction particle

$$w^i_t = \frac{p(y_t|x^i_{t|t-1}, \theta)}{\sum_{j=1}^{P} p(y_t|x^j_{t|t-1}, \theta)} \quad \text{for } i = 1, \ldots, P$$

and then picks the filtering particles via multinomial sampling, using the computed importance weights as respective probabilities:

$$x^j_{t|t} \sim MN(w^1_t, \ldots, w^P_t) \quad \text{for } j = 1, \ldots, P$$

The algorithm's main characteristic is the resampling within the filtering step which removes particles with small weights with high probability while likely copying particles with high weights multiple times. While this step increases the immediate Monte Carlo variance, it gives better stability for future steps by reducing the risk of weight degeneracy (Doucet and Johansen, 2008).

3.2.2 Algorithm

Algorithm 2 (Local Level) Particle filter

procedure ParticleFilter($\mathcal{I}_T, \boldsymbol{\theta}, \boldsymbol{\mu}_1, \Sigma_1$)
  $\boldsymbol{\mu}_{1|0} \leftarrow \boldsymbol{\mu}_1$ ▷ initialization
  $\Sigma_{1|0} \leftarrow \Sigma_1$
  for t `in` $1:T$ **do**
    $\boldsymbol{v}_t = \boldsymbol{y}_t - \boldsymbol{\mu}_{t|t-1}$
    $K_t = \Sigma_{t|t-1}(\Sigma_{t|t-1} + \Sigma_\epsilon)^{-1}$
    $\boldsymbol{\mu}_{t|t} = \boldsymbol{\mu}_{t|t-1} + K_t \boldsymbol{v}_t$ ▷ update step
    $\Sigma_{t|t} = \Sigma_{t|t-1}(1 - K_t)$
    $\boldsymbol{\mu}_{t+1|t} \leftarrow \boldsymbol{\mu}_{t|t}$ ▷ prediction step
    $\Sigma_{t+1|t} \leftarrow \Sigma_{t|t} + \Sigma_\eta$
  return $\{\boldsymbol{\mu}_{t|t}\}$, $\{\Sigma_{t|t}\}$

3.2.3 Likelihood evaluation

3.3 Importance Sampling Particle Filter

4 Illustration

4.1 Trivariate Local Level Model

4.1.1 The Model

Consider a time series of length T with each observation $\boldsymbol{y}_t = [y_{1t}, y_{2t}, y_{3t}]^T$ and each state $\boldsymbol{x}_t = [x_{1t}, x_{2t}, x_{3t}]^T$ being described by a 3-dimensional vector.

$$\begin{aligned} \text{observation:} \quad & \boldsymbol{y}_t = \boldsymbol{x}_t + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim N(\boldsymbol{0}, \sigma_\epsilon^2 I_3) \\ \text{state:} \quad & \boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim N(\boldsymbol{0}, \Sigma_\eta) \end{aligned}$$

with initial state $\boldsymbol{x}_1 \sim N(\boldsymbol{a}_1, P_1)$ and where we restrict the covariance matrix of the state disturbances, Σ_η, to the form

$$\Sigma_\eta = \begin{bmatrix} \sigma_{\eta 1}^2 & \rho\sigma_{\eta 1}\sigma_{\eta 2} & \rho\sigma_{\eta 1}\sigma_{\eta 3} \\ \rho\sigma_{\eta 1}\sigma_{\eta 2} & \sigma_{\eta 2}^2 & \rho\sigma_{\eta 2}\sigma_{\eta 3} \\ \rho\sigma_{\eta 1}\sigma_{\eta 3} & \rho\sigma_{\eta 2}\sigma_{\eta 3} & \sigma_{\eta 3}^2 \end{bmatrix}$$

Thus, Σ_η can be described by $\sigma_{\eta 1}^2$, $\sigma_{\eta 2}^2$, $\sigma_{\eta 3}^2 > 0$ and $\rho \in [0, 1]$. Furthermore, we assume for simplicity that the observation noise has the same variance in each dimension $\sigma_\epsilon^2 > 0$. Therefore, the model is fully specified by the following vector of parameters:

$$\boldsymbol{\theta} = [\rho, \sigma_{\eta 1}^2, \sigma_{\eta 2}^2, \sigma_{\eta 3}^2, \sigma_\epsilon^2]^T$$

The initial state parameters $\boldsymbol{a}_1$ and P_1 are assumed to be known.

4.1.2 Realization

Figure 4.1 plots the states and observations for a realization of the trivariate local level model with length $T = 100$. The model parameters are

$$\boldsymbol{\theta} = [\rho = 0.7, \sigma_{\eta 1}^2 = 4.2, \sigma_{\eta 2}^2 = 2.8, \sigma_{\eta 3}^2 = 0.9, \sigma_\epsilon^2 = 1.0]^T$$

The initial state x_1 is drawn from a standard normal.

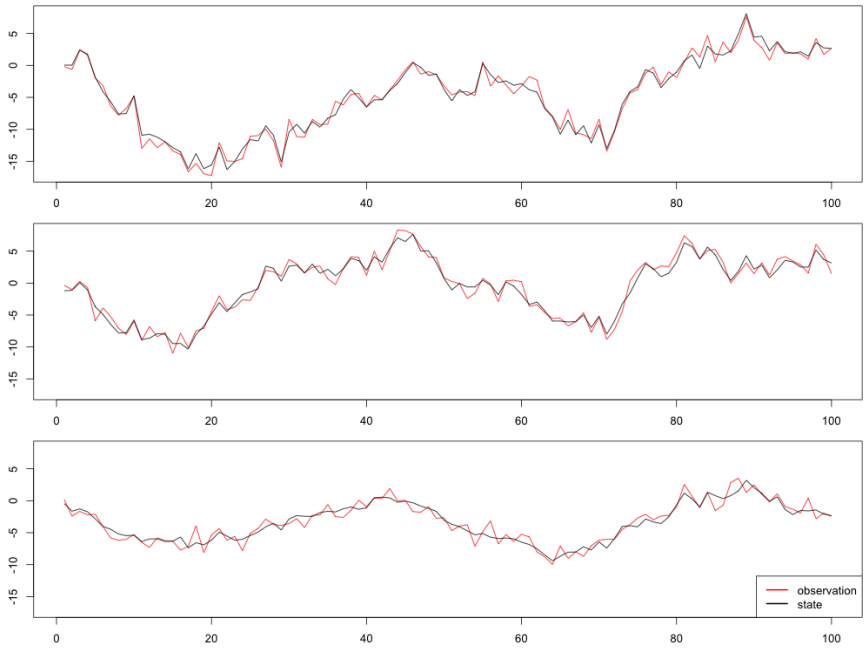

Figure 4.1: Realization of the model with $T = 100$

4.2 Hierarchical Dynamic Poisson Model

Explain the main idea and potential use cases.

4.2.1 The Model

Consider a time series over M days, each consisting of N intra-daily observations. Let m denote the day and n be the intraday index.

$$\begin{aligned} \text{observation:} \quad & y_{mn} && = \text{Poisson}(\lambda_{mn}) \\ \text{state:} \quad & \log \lambda_{mn} && = \log \lambda_m^{(D)} + \log \lambda_{mn}^{(I)} + \log \lambda_n^{(P)} \end{aligned}$$

where the state consists of a daily, an intra-daily, and a periodic component:

$$\begin{aligned} \text{daily component:} \quad & \log \lambda_{m+1}^{(D)} = \phi_0^{(D)} + \phi_1^{(D)} \log \lambda_m^{(D)} + \eta_m^{(D)} && \eta_t \sim N(0, \sigma_{(D)}^2) \\ \text{intra-daily component:} \quad & \log \lambda_{mn+1}^{(I)} = \phi_1^{(I)} \log \lambda_{mn}^{(I)} + \eta_{mn}^{(I)} && \eta_{mn} \sim N(0, \sigma_{(I)}^2) \\ \text{periodic component:} \quad & \log \lambda_n^{(P)} = \phi_1^{(P)} \sin(\pi(n-1)/M) \end{aligned}$$

The initial daily and intra-daily component is drawn from a normal with mean a_1 and covariance P_1:

$$\log \lambda_1^{(D)}, \log \lambda_1^{(I)} \sim N(a_1, P_1)$$

Note that both the daily and intra-daily component constitute an AR(1) model, with the mean of the intra-daily component $\phi_0^{(I)}$ set to 0. The model is fully specified by the following vector of parameters:

$$\boldsymbol{\theta} = [\phi_0^{(D)}, \phi_1^{(D)}, \sigma^2_{(D)}, \phi_1^{(I)}, \sigma^2_{(I)}, \phi_1^{(P)}]^T$$

Again, the initial state parameters a_1 and P_1 are assumed to be known.

4.2.2 Realization

Figure 4.2 plots the states and observations for a realization of the hierarchical dyanmic Poisson model over $N = 5$ days with $M = 20$ intra-daily observations. The model parameters are

$$\boldsymbol{\theta} = [\phi_0^{(D)} = 0.7, \phi_1^{(D)} = 0.6, \sigma^2_{(D)} = 0.6, \phi_1^{(I)} = 0.3, \sigma^2_{(I)} = 0.2, \phi_1^{(P)} = 0.8]^T$$

The initial daily and intra-daily state components were drawn from a standard normal.

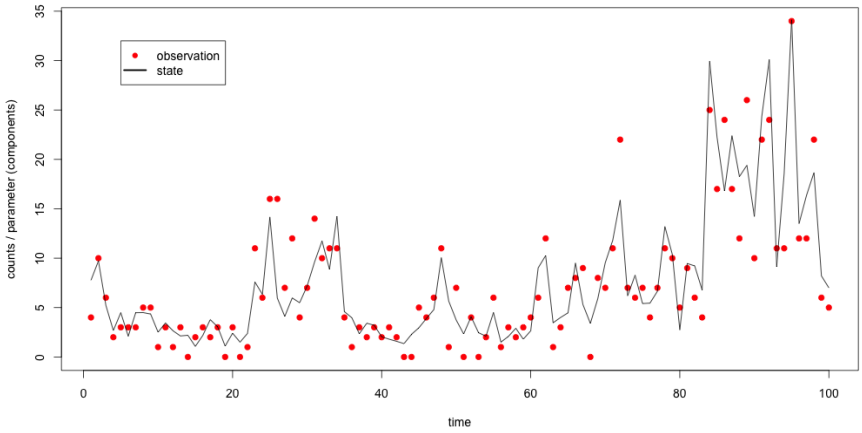

Figure 4.2: Realization of the model with $N = 5$ and $M = 20$

4.2.3 Densities

State transition and prediction density and how they are used in the particle filter

4.2.4 Maximum Likelihood Estimation

Show log-likelihood plots

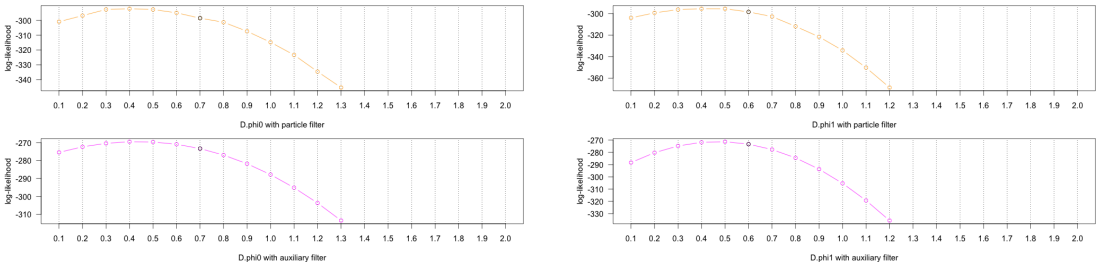

Figure 4.3: Belief convergence without misinformation after 300 and 2000 iterations

5 Evaluation

Particle filtering suffers from propagation error. Current approximations errors are related to past approximation errors. The number of particles P should increase with the sample size T to guarantee a good degree of approximation.

6 Conclusion

by Etessami et al.[**?**]