# Computer Intensive Methods in Mathematical Statistics

Johan Westerborn

Department of mathematics
KTH Royal Institute of Technology
johawes@kth.se

Lecture 16
Advanced topics in computational statistics
18 May 2017

# Plan of today's lecture

1. Last time: the EM algorithm

2. An Introduction to my Research
   - The Particle-Based Rapid Incremental Smoother

# Outline

# Last time: the EM algorithm

- The algorithm goes as follows.

  **Data:** Initial value $\theta_0$
  **Result:** $\{\theta_\ell; \ell \in \mathbb{N}\}$
  **for** $\ell \leftarrow 0, 1, 2, \ldots$ **do**
  set $\mathcal{Q}_{\theta_\ell}(\theta) \leftarrow \mathbb{E}_{\theta_\ell}(\log f_\theta(X, Y) \mid Y)$;
  set $\theta_{\ell+1} \leftarrow \arg\max_{\theta \in \Theta} \mathcal{Q}_{\theta_\ell}(\theta)$
  **end**

- The two steps within the main loop are referred to as expectation and maximization steps, respectively.

# Last time: the EM inequality

- By rearranging the terms we obtain the following.

**Theorem (the EM inequality)**

*For all* $(\theta, \theta') \in \Theta^2$ *it holds that*

$$\ell(\theta) - \ell(\theta') \geq \mathcal{Q}_{\theta'}(\theta) - \mathcal{Q}_{\theta'}(\theta'),$$

*where the equality is strict unless* $f_{\theta'}(x \mid Y) = f_{\theta}(x \mid Y)$.

- Thus, by the very construction of $\{\theta_\ell; \ell \in \mathbb{N}\}$ it is made sure that $\{\ell(\theta_\ell); \ell \in \mathbb{N}\}$ is non-decreasing. Hence, the EM algorithm is a monotone optimization algorithm.

# EM in exponential families

- In order to be practically useful, the E- and M-steps of EM have to be feasible. A rather general context in which this is the case is the following.

**Definition (exponential family)**

The family $\{f_\theta(x, y); \theta \in \Theta\}$ defines an exponential family if the complete data likelihood is of form

$$f_\theta(x, y) = \exp\left(\psi(\theta)^\intercal \phi(x) - c(\theta)\right) h(x),$$

where $\phi$ and $\psi$ are (possibly) vector-valued functions on $\mathbb{R}^d$ and $\Theta$, respectively, and $h$ is a non-negative real-valued function on $\mathbb{R}^d$. All these quantities may depend on $y$.

# EM in exponential families (cont'd)

- The intermediate quantity becomes

$$\mathcal{Q}_{\theta'}(\theta) = \psi(\theta)^\intercal \mathbb{E}_{\theta'}\left(\phi(x) \mid Y\right) - c(\theta) + \underbrace{\mathbb{E}_{\theta'}\left(\log h(x) \mid Y\right)}_{(*)},$$

  where $(*)$ does not depend on $\theta$ and may thus be ignored.
- Consequently, in order to be able to apply EM we need
  1. to be able to compute the smoothed sufficient statistics

$$\tau = \mathbb{E}_{\theta'}\left(\phi(X) \mid Y\right) = \int \phi(x) f_{\theta'}(x \mid Y),$$

  2. maximization of $\theta \mapsto \psi(\theta)^\intercal \tau - c(\theta)$ to be feasible for all $\tau$.

# Example: nonlinear Gaussian model

- Let $(y_i)_{i=1}^n$ be independent observations of a random variable

$$Y = h(X) + \sigma_y \varepsilon_y,$$

where $h$ is a possibly nonlinear function and

$$X = \mu + \sigma_x \varepsilon_x$$

is not observable. Here $\varepsilon_x$ and $\varepsilon_y$ are independent standard Gaussian noise variables.

- The parameters $\theta = (\mu, \sigma_x^2)$ governing the distribution of the unobservable variable $X$ are unknown while it is known that $\sigma_y = .4$.

# Example: nonlinear Gaussian model (cont.)

- In this case the complete data likelihood belongs to an exponential family with

$$\phi(x_{1:n}) = \left( \sum_{i=1}^{n} x_i^2, \sum_{i=1}^{n} x_i \right), \quad \psi(\theta) = \left( -\frac{1}{2\sigma_x^2}, \frac{\mu}{\sigma_x^2} \right),$$

  and

$$c(\theta) = \frac{n}{2} \log \sigma_x^2 + \frac{n\mu^2}{2\sigma_x^2}.$$

- Letting $\tau_i = \mathbb{E}_{\theta_\ell}(\phi_i(X_{1:n}) \mid Y_{1:n})$, $i \in \{1, 2\}$, leads to

$$\mu_{\ell+1} = \frac{\tau_2}{n},$$
$$(\sigma_x^2)_{\ell+1} = \frac{\tau_1}{n} - \mu_{\ell+1}^2.$$

# Example: nonlinear Gaussian model (cont.)

- Since the "smoothed" sufficient statistics are of additive form it holds that, e.g.,

$$\tau_1 = \mathbb{E}_{\theta_\ell}\left(\sum_{i=1}^{n} X_i^2 \mid Y_{1:n}\right) = \sum_{i=1}^{n} \mathbb{E}_{\theta_\ell}\left(X_i^2 \mid Y_i\right)$$

  (and similarly for $\tau_2$).
- However, computing expectations under

$$f_{\theta_\ell}(x_i \mid y_i) \propto \exp\left(-\frac{1}{2(\sigma_y^2)_\ell}\{y_i - h(x_i)\}^2 - \frac{1}{2\sigma_y^2}\{x_i - \mu_\ell\}\right)$$

  is in general infeasible (i.e. when the transformation $h$ is a general nonlinear function).

# Example: nonlinear Gaussian model (cont.)

- Thus, within each iteration of EM we sample each component $f_{\theta_\ell}(x_i \mid y_i)$ using MH.
- For simplicity we use the independent proposal $r(x_i) = f_{\theta_\ell}(x_i)$, yielding the MH acceptance probability

$$\alpha(X_i^{(k)}, X_i^*) = 1 \wedge \frac{f_{\theta_\ell}(Y_i \mid X_i^*)}{f_{\theta_\ell}(Y_i \mid X_i^{(k)})} =$$

$$1 \wedge \exp\left( -\frac{1}{2(\sigma_y^2)_\ell} \{h^2(X_i^*) - h^2(X_i^{(k)}) + 2Y_i(h(X_i^{(k)}) - h(X_i^*))\}^2 \right).$$

# Example: nonlinear Gaussian model (cont.)

**Data:** Initial value $\theta_0$; $Y_{1:n}$
**Result:** $\{\theta_\ell; \ell \in \mathbb{N}\}$
**for** $\ell \leftarrow 0, 1, 2, \ldots$ **do**
  set $\hat{\tau}_j \leftarrow 0$, $\forall j$;
  **for** $i \leftarrow 1, \ldots, n$ **do**
    run an MH sampler targeting $f_{\theta_\ell}(x_i \mid Y_i) \rightsquigarrow (X_i^{(k)})_{k=1}^{N_\ell}$;
    set $\hat{\tau}_1 \leftarrow \hat{\tau}_1 + \sum_{k=1}^{N_\ell} (X_i^{(k)})^2 / N_\ell$;
    set $\hat{\tau}_2 \leftarrow \hat{\tau}_2 + \sum_{k=1}^{N_\ell} X_i^{(k)} / N_\ell$;
  **end**
  set $\mu_{\ell+1} \leftarrow \hat{\tau}_2 / n$;
  set $(\sigma_x^2)_{\ell+1} \leftarrow \hat{\tau}_1 / n - \mu_{\ell+1}^2$;
**end**

# Example: nonlinear Gaussian model (cont.)

- In order to assess the performance of this Monte Carlo EM algorithm we focus on the linear case, $h(x) = x$.
- A data record comprising $n = 40$ values was produced by simulation under $(\mu^*, \sigma_x^*) = (1, .4)$ with $\sigma_y = .4$.
- In this case, the true MLE is known and given by (check this!)

$$\hat{\mu}(Y_{1:n}) = \frac{1}{n}\sum_{i=1}^{n} Y_i = 1.04,$$

$$\hat{\sigma}_x^2(Y_{1:n}) = \frac{1}{n}\sum_{i=1}^{n}\{Y_i - \hat{\mu}(Y_{1:n})\}^2 - .4^2 = .27.$$

# Example: nonlinear Gaussian model (cont.)

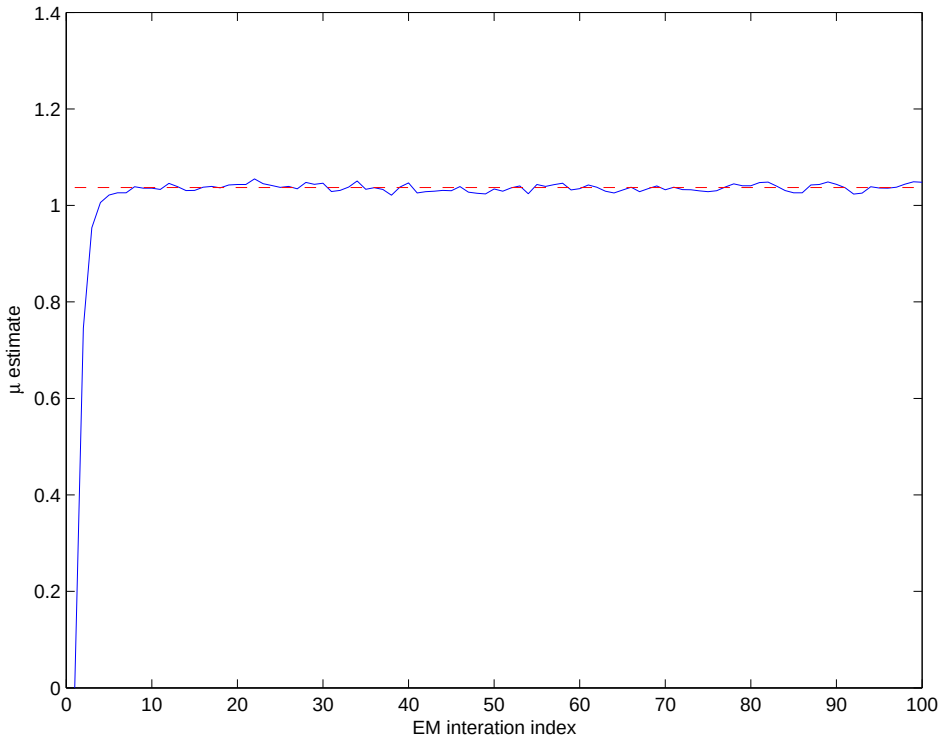



Figure: EM learning trajectory (blue curve) for $\mu$ in the case $h(x) = x$. Red-dashed line indicates true MLE. $N_\ell = 200$ for all iterations.

# Example: nonlinear Gaussian model (cont.)

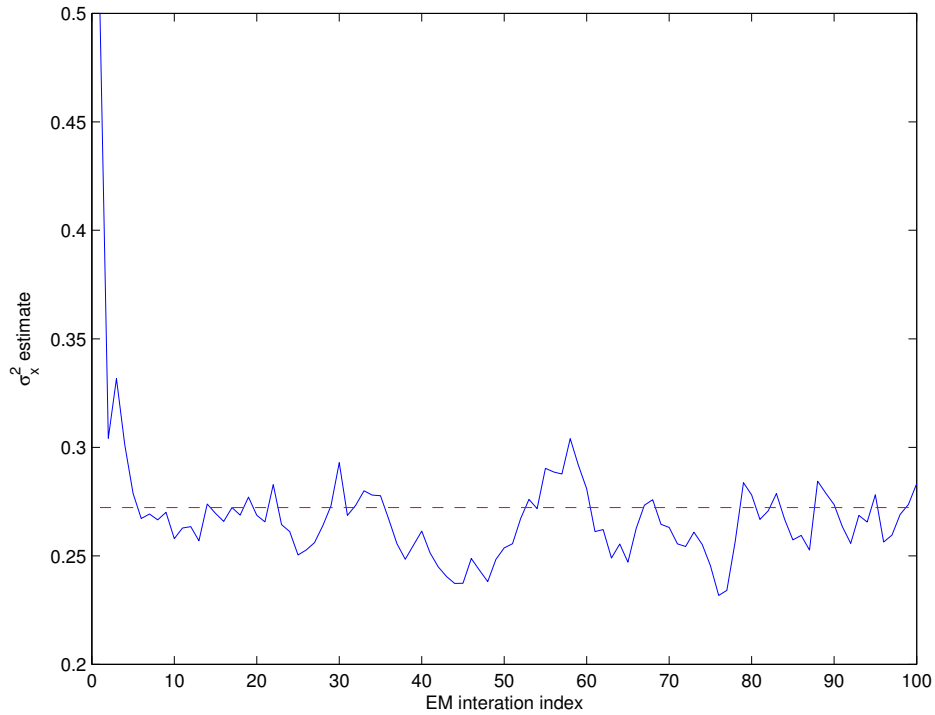



Figure: EM learning trajectory (blue curve) for $\sigma_x^2$ in the case $h(x) = x$. Red-dashed line indicates true MLE. $N_\ell = 200$ for all iterations.

# Averaging

- Roughly, for the MCEM algorithm one may prove that the variance of $\theta_\ell - \hat{\theta}$ (where $\hat{\theta}$ is the MLE) is of order $1/N_\ell$.
- In the idealised situation where the estimates $(\theta_\ell)_\ell$ are uncorrelated (which is not the case here) one may obtain an improved estimator of $\hat{\theta}$ by combining the individual estimates $\theta_\ell$ in proportion of the inverse of their variance. Starting the averaging at iteration $\ell_0$ leads to

$$\tilde{\theta}_\ell = \sum_{\ell^*=\ell_0}^{\ell} \frac{N_{\ell^*}}{\sum_{\ell'=\ell_0}^{\ell} N_{\ell'}} \theta_{\ell^*}, \quad \ell \geq \ell_0.$$

- In the idealised situation the variance of $\tilde{\theta}_\ell$ decreases as $1/\sum_{\ell'=\ell_0}^{\ell} N_{\ell'}$ (= total number of simulations).

# Example: nonlinear Gaussian model (cont.)

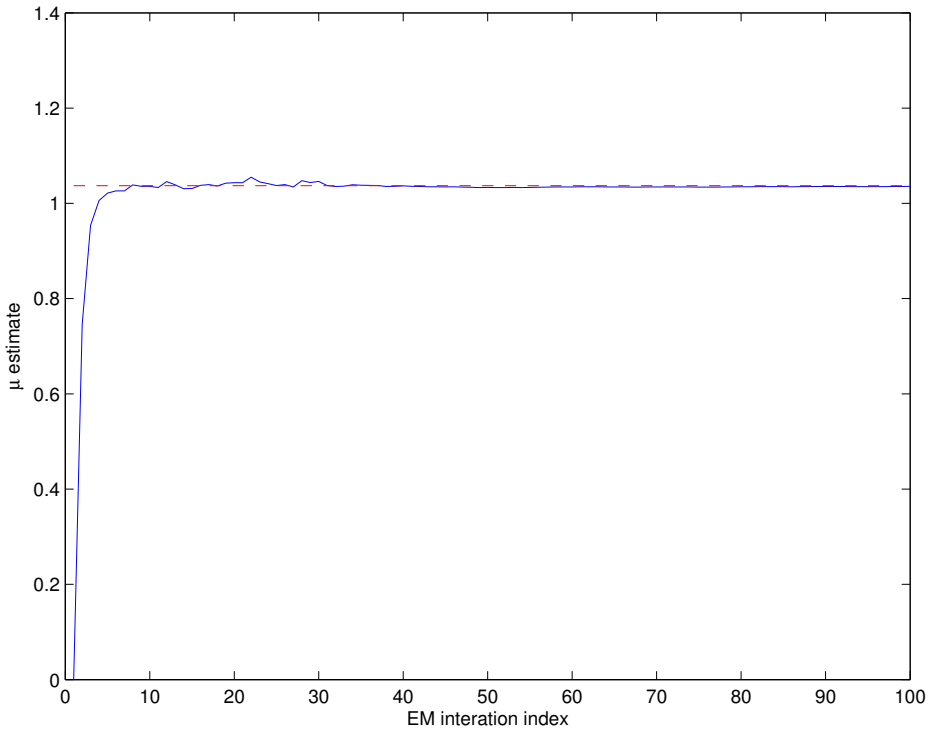



Figure: EM learning trajectory (blue curve) for $\mu$ in the case $h(x) = x$. Red-dashed line indicates true MLE. Averaging after $\ell_0 = 30$ iterations.

# Example: nonlinear Gaussian model (cont.)

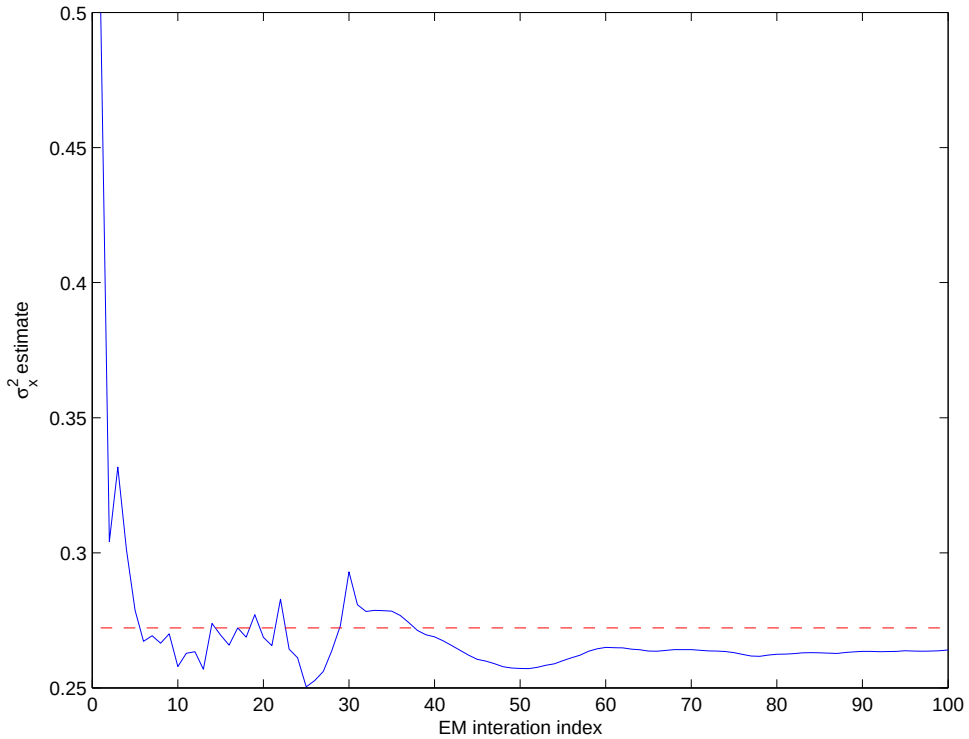



Figure: EM learning trajectory (blue curve) for $\sigma_x^2$ in the case $h(x) = x$. Red-dashed line indicates true MLE. Averaging after $\ell_0 = 30$ iterations.

# Outline

# Outline

# General hidden Markov models (HMMs)

- A hidden Markov model (HMM) comprises
  1. a Markov chain $(X_k)_{k\geq 0}$ with transition density $q$, i.e.

$$X_{k+1} \mid X_k = x_k \sim q(x_{k+1} \mid x_k),$$

  2. which is hidden away from us but partially observed through an observation process $(Y_k)_{k\geq 0}$ such that conditionally on the chain $(X_k)_{k\geq 0}$,
     (i) the $Y_k$'s are independent with
     (ii) conditional distribution of each $Y_k$ depending on the corresponding $X_k$ only.
- The density of the conditional distribution $Y_k \mid (X_k)_{k\geq 0} \overset{\text{d.}}{=} Y_k \mid X_k$ will be denoted by $p(y_k \mid x_k)$.

# General hidden Markov models (HMMs)

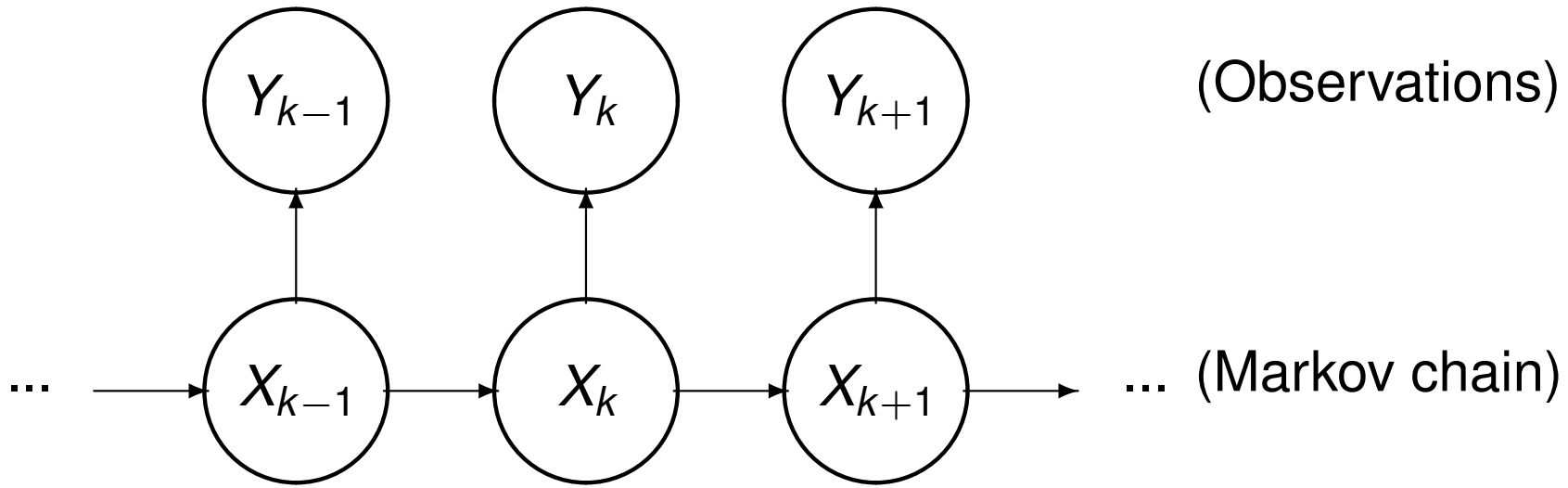
- A hidden Markov model (HMM) comprises
  1. a Markov chain $(X_k)_{k\geq 0}$ with transition density $q$, i.e.

$$X_{k+1} \mid X_k = x_k \sim q(x_{k+1} \mid x_k),$$

  2. which is hidden away from us but partially observed through an observation process $(Y_k)_{k\geq 0}$ such that conditionally on the chain $(X_k)_{k\geq 0}$,
     - (i) the $Y_k$'s are independent with
     - (ii) conditional distribution of each $Y_k$ depending on the corresponding $X_k$ only.
- The density of the conditional distribution $Y_k \mid (X_k)_{k\geq 0} \overset{\text{d.}}{=} Y_k \mid X_k$ will be denoted by $p(y_k \mid x_k)$.

# General HMMs (cont.)

- Graphically:




$$Y_k \mid X_k = x_k \sim p(y_k \mid x_k) \qquad \text{(Observation density)}$$
$$X_{k+1} \mid X_k = x_k \sim q(x_{k+1} \mid x_k) \qquad \text{(Transition density)}$$
$$X_0 \sim \chi(x_0) \qquad \text{(Initial distribution)}$$

# The smoothing distribution

- In an HMM, the smoothing distribution $f_n(x_{0:n} \mid y_{0:n})$ is the conditional distribution of $X_{0:n}$ given $Y_{0:n} = y_{0:n}$.

## Theorem (Smoothing distribution)

$$f_n(x_{0:n} \mid y_{0:n}) = \frac{\chi(x_0)p(y_0 \mid x_0)\prod_{k=1}^{n} p(y_k \mid x_k)q(x_k \mid x_{k-1})}{L_n(y_{0:n})},$$

*where*

$$L_n(y_{0:n}) = \textit{density of the observations } y_{0:n}$$
$$= \int \chi(x_0)p(y_0 \mid x_0)\prod_{k=1}^{n} p(y_k \mid x_k)q(x_k \mid x_{k-1})\, dx_{0:n}.$$

# The goal

- We wish, for $n \in \mathbb{N}$, to estimate the expected value

$$\tau_n = \mathbb{E}[h_n(X_{0:n}) \mid Y_{0:n}],$$

  under the smoothing distribution.
- Here $h_n(x_{0:n})$ is of additive form, that is

$$h_n(x_{0:n}) = \sum_{i=0}^{n-1} \tilde{h}_i(x_{i:i+1}).$$

- For instance sufficient statistics which are needed for the EM algorithm.

# The goal

- We wish, for $n \in \mathbb{N}$, to estimate the expected value

$$\tau_n = \mathbb{E}[h_n(X_{0:n}) \mid Y_{0:n}],$$

  under the smoothing distribution.
- Here $h_n(x_{0:n})$ is of additive form, that is

$$h_n(x_{0:n}) = \sum_{i=0}^{n-1} \tilde{h}_i(x_{i:i+1}).$$

- For instance sufficient statistics which are needed for the EM algorithm.

# Using basic SISR

- Given particles and weights $\{(X_{0:n}^i, \omega_n^i)\}_{i=1}^N$ targeting $f(x_{0:n} \mid y_{0:n})$ we update this to estimate at time $n+1$ using the following steps:

**Selection:** Draw indices $\{I_n^i\}_{i=1}^N \sim \text{Mult}(\{\omega_n^i\}_{i=1}^N)$

**Mutation:** For $i = 1, \ldots, N$ draw $X_{n+1}^i \sim q(\cdot \mid X_n^{I_n^i})$ and set $X_{0:n+1}^i = (X_{0:n}^{I_n^i}, X_{n+1}^i)$

**Weighting:** For $i = 1, \ldots, N$ set $\omega_{n+1}^i = p(y_{n+1} \mid X_{n+1}^i)$.

**Estimation:** We estimate $\tau_{n+1}$ using

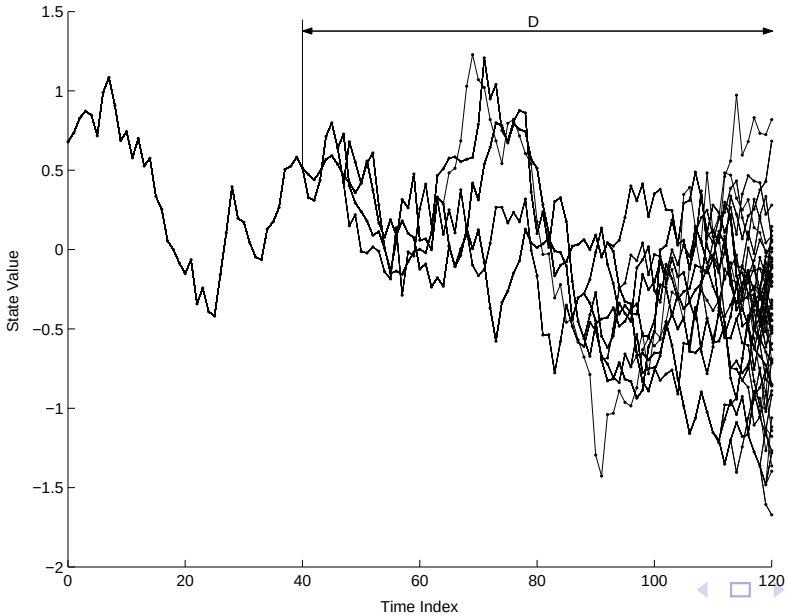
$$\hat{\tau}_{n+1} = \sum_{i=1}^N \frac{\omega_{n+1}^i}{\sum_{\ell=1}^N \omega_{n+1}^\ell} h_{n+1}(X_{0:n+1}^i).$$

# The problem of resampling

- We saw when comparing with the SIS algorithm that the resampling was necessary to achieve a stable estimate.
- But the resampling also depletes the historical trajectories.
- There will be an integer $k$ such that at time $n$ in the algorithm $X_{0:k}^i = X_{0:k}^j$ for all $i$ and $j$.

# The problem of resampling

- We saw when comparing with the SIS algorithm that the resampling was necessary to achieve a stable estimate.
- But the resampling also depletes the historical trajectories.
- There will be an integer $k$ such that at time $n$ in the algorithm $X_{0:k}^i = X_{0:k}^j$ for all $i$ and $j$.



# Fixing the degeneracy

- The goal to fixing this problem is the following observation:
  - Conditioned on the observations the hidden Markov chain is also a Markov chain in the reverse direction.
  - We denote the transition density of this Markov chain by $\overleftarrow{q}(x_k \mid x_{k+1}, y_{0:k})$, this density can be written as

$$\overleftarrow{q}(x_k \mid x_{k+1}, y_{0:k}) = \frac{f(x_k \mid y_{0:k}) q(x_{k+1} \mid x_k)}{\int f(x' \mid y_{0:k}) q(x_{k+1} \mid x') \mathrm{d}x'}.$$

  - Since we can estimate $f(x_k \mid y_{0:k})$ well using SISR we can use the following particle approximation of the backward distribution

$$\overleftarrow{q}^N(x_k^i \mid x_{k+1}^j, y_{0:k}) = \frac{\omega_k^i q(x_{k+1}^j \mid x_k^i)}{\sum_{\ell=1}^N \omega_k^\ell q(x_{k+1}^j \mid x_k^\ell)}.$$

# Fixing the degeneracy

- The goal to fixing this problem is the following observation:
  - Conditioned on the observations the hidden Markov chain is also a Markov chain in the reverse direction.
  - We denote the transition density of this Markov chain by $\overleftarrow{q}(x_k \mid x_{k+1}, y_{0:k})$, this density can be written as

$$\overleftarrow{q}(x_k \mid x_{k+1}, y_{0:k}) = \frac{f(x_k \mid y_{0:k}) q(x_{k+1} \mid x_k)}{\int f(x' \mid y_{0:k}) q(x_{k+1} \mid x') \mathrm{d}x'}.$$

  - Since we can estimate $f(x_k \mid y_{0:k})$ well using SISR we can use the following particle approximation of the backward distribution

$$\overleftarrow{q}^N(x_k^i \mid x_{k+1}^j, y_{0:k}) = \frac{\omega_k^i q(x_{k+1}^j \mid x_k^i)}{\sum_{\ell=1}^N \omega_k^\ell q(x_{k+1}^j \mid x_k^\ell)}.$$

# Forward Filtering Backward Smoothing (FFBSm)

- Using this approximation of the transition density we get the Forward Filtering Backward Smoothing algorithm.
- First we need to run the SISR algorithm and save all the filter estimates $\{(x_k^i, \omega_k^i)\}_{i=1}^N$ for $k = 0, \ldots, N$.
- We then estimate $\tau_n$ using

$$\hat{\tau}_n = \sum_{i_0=1}^{N} \cdots \sum_{i_n=0}^{N} \prod_{s=0}^{n-1} \frac{\omega_s^{i_s} q(x_{s+1}^{i_{s+1}} \mid x_s^{i_s})}{\sum_{\ell=1}^{N} \omega_s^{\ell} q(x_{s+1}^{i_{s+1}} \mid x_s^{\ell})} \times \frac{\omega_n^{i_n}}{\sum_{\ell=1}^{N} \omega_n^{\ell}} h_n(x_0^{i_0}, x_1^{i_1}, \ldots, x_n^{i_n})$$

# A Forward only version

- The previous algorithm requires two passes of the data, first the forward filtering, then the backward smoothing.
- When working with functions of additive form it is possible to perform smoothing in the following way.
  - Introduce the auxiliary function $T_k(x)$, which we define as

$$T_k(x) = \mathbb{E}[h_k(X_{0:k}) \mid Y_{0:k}, X_k = x]$$

  - We can update this function recursively by noting that

$$T_{k+1}(x) = \mathbb{E}[T_k(X_k) + \tilde{h}_k(X_k, X_{k+1}) \mid Y_{0:k}, X_{k+1} = x],$$

    that is the expected value of the previous function with the next additive part under the backward distribution!

# Using this decomposition

- Given that we have estimates for the filter at time $k$, $\{(x_k^i, \omega_k^i)\}_{i=1}^N$, and estimates of $\{T_k(x_k^i)\}_{i=1}^N$.
- Propagate the filter estimates using one step of SISR algorithm to get $\{(x_{k+1}^i, \omega_{k+1}^i)\}_{i=1}^N$, now for $i = 1, \ldots, N$ estimate the function $T_{k+1}(x_{k+1}^i)$ using

$$T_{k+1}(x_{k+1}^i) = \sum_{j=1}^N \frac{\omega_k^j q(x_{k+1}^i \mid x_k^j)}{\sum_{\ell=1}^N \omega_k^\ell q(x_{k+1}^i \mid x_k^\ell)} (T_k(x_k^j) + \tilde{h}(x_k^j, x_{k+1}^i))$$

- $\tau_{k+1}$ is estimated using

$$\hat{\tau}_{k+1} = \sum_{i=1}^N \frac{\omega_{k+1}^i}{\sum_{\ell=1}^N \omega_{k+1}^\ell} T_{k+1}(x_{k+1}^i)$$

# Speeding up the Algorithm

- This algorithm works, but it is quite slow.
- Idea! Can we replace calculating the expected value with an MC estimator? Can we sample sufficiently fast from the backward distribution?
- Formally we wish to sample $J$ such that given all particles and weights at time $k$ and $k+1$ and an index $i$

$$\mathbb{P}(J = j) = \frac{\omega_k^j q(x_{k+1}^i \mid x_k^j)}{\sum_{\ell=1}^N \omega_k^\ell q(x_{k+1}^i \mid x_k^\ell)}$$

- We can do this efficiently using accept-reject sampling!

# Speeding up the Algorithm

- This algorithm works, but it is quite slow.
- Idea! Can we replace calculating the expected value with an MC estimator? Can we sample sufficiently fast from the backward distribution?
- Formally we wish to sample $J$ such that given all particles and weights at time $k$ and $k+1$ and an index $i$

$$\mathbb{P}(J = j) = \frac{\omega_k^j q(x_{k+1}^i \mid x_k^j)}{\sum_{\ell=1}^N \omega_k^\ell q(x_{k+1}^i \mid x_k^\ell)}$$

- We can do this efficiently using accept-reject sampling!

# Speeding up the Algorithm

- This algorithm works, but it is quite slow.
- Idea! Can we replace calculating the expected value with an MC estimator? Can we sample sufficiently fast from the backward distribution?
- Formally we wish to sample $J$ such that given all particles and weights at time $k$ and $k+1$ and an index $i$

$$\mathbb{P}(J = j) = \frac{\omega_k^j q(x_{k+1}^i \mid x_k^j)}{\sum_{\ell=1}^N \omega_k^\ell q(x_{k+1}^i \mid x_k^\ell)}$$

- We can do this efficiently using accept-reject sampling!

# The PaRIS algorithm

- Now we can describe the PaRIS algorithm.
  - Given particles and weights $\{(x_k^i, \omega_k^i)\}_{i=1}^N$ targeting the filter distribution at time $k$. Together with values $\{T_k(x_k^i)\}_{i=1}^N$ estimating $T_k(x)$.
  - We proceed by first taking one step using the SISR algorithm to get particles and weights $\{(x_{k+1}^i, \omega_{k+1}^i)\}_{i=1}^n$ targeting the filter at time $k+1$.
  - For $i = 1, \ldots, N$ we draw $\tilde{N}$ indices $\{J^{i'}\}_{i'=1}^{\tilde{N}}$ from the previous slide and calculate

$$T_{k+1}(x_{k+1}^i) = \frac{1}{\tilde{N}} \sum_{i'=1}^{\tilde{N}} \left( T_k(x_k^{J^{i'}}) + \tilde{h}_k(x_k^{J^{i'}}, x_{k+1}^i) \right)$$

  - The estimate of $\tau_{k+1}$ is then given by

$$\hat{\tau}_{k+1} = \sum_{i=1}^N \frac{\omega_{k+1}^i}{\sum_{\ell=1}^N \sum_{k+1}^{\ell}} T_{k+1}(x_{k+1}^i)$$

# The PaRIS algorithm (cont.)

- Using this algorithm we have an efficient algorithm for online estimation of the expected value under the joint smoothing distribution.
- We require the target function to be of additive form.
- The number of backward samples ($\tilde{N}$) needed in the algorithm turns out to be 2.
- We can (under some additional assumptions) prove the following:
  - The computational complexity grows linearly with $N$.
  - The estimate is asymptotically consistent.
  - We can find a CLT for the error which behaves like $1/\sqrt{N}$.

# An Example, Paramter Estimation.

It turns out that the PaRIS algorithm can be used efficently when performing online parameter estimation.

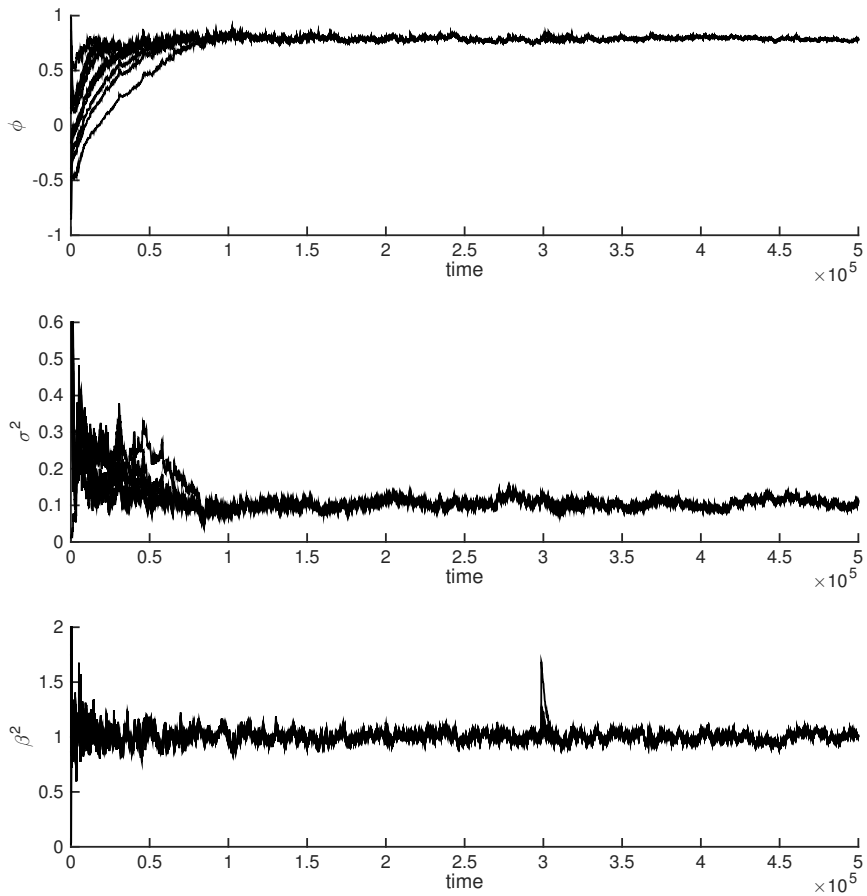
We tested our method on the stochastic volatility model

$$\begin{aligned} X_{t+1} &= \phi X_t + \sigma V_{t+1}, \\ Y_t &= \beta \exp(X_t/2) U_t, \end{aligned} \qquad t \in \mathbb{N},$$

where $\{V_t\}_{t\in\mathbb{N}}$ and $\{U_t\}_{t\in\mathbb{N}}$ are independent sequences of mutually independent standard Gaussian noise variables.

Parameters to be estimated are $\theta = (\phi, \sigma^2, \beta^2)$.

# Parameter Estimation (cont.)



# End of the course

- That is it for the lectures in this course. Hope you have enjoyed it.
- Review of HA2 sent by mail to `johawes@kth.se` by 24 May 12:00:00
- Exam 30 May, 14:00:00 – 19:00:00
- Re-Exam 14 August, 08:00:00 – 13:00:00