CPSC 535
Sequential Importance Sampling & Resampling

AD

13th February 2007

- Sequential Importance Sampling

- Sequential Importance Sampling
- Resampling

Importance Sampling Review

- Let $\pi(x) = \frac{\gamma(x)}{Z}$ and $q(x)$ be pdf on $\mathcal{X}$ such that $\pi(x) > 0 \Rightarrow q(x) > 0$.

Importance Sampling Review

- Let $\pi(x) = \frac{\gamma(x)}{Z}$ and $q(x)$ be pdf on $\mathcal{X}$ such that $\pi(x) > 0 \Rightarrow q(x) > 0$.
- IS is based on the identities

$$\begin{aligned} \pi(x) &= \frac{w(x)\, q(x)}{Z},\ Z = \int w(x)\, q(x)\, dx, \\ \text{where } w(x) &= \frac{\gamma(x)}{q(x)} \propto \frac{\pi(x)}{q(x)}. \end{aligned}$$

Importance Sampling Review

- Let $\pi(x) = \frac{\gamma(x)}{Z}$ and $q(x)$ be pdf on $\mathcal{X}$ such that $\pi(x) > 0 \Rightarrow q(x) > 0$.
- IS is based on the identities

$$\begin{aligned} \pi(x) &= \frac{w(x)\, q(x)}{Z}, \; Z = \int w(x)\, q(x)\, dx, \\ \text{where } w(x) &= \frac{\gamma(x)}{q(x)} \propto \frac{\pi(x)}{q(x)}. \end{aligned}$$

- Given

$$\widehat{q}_N(x) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X^{(i)}}(x) \text{ where } X^{(i)} \overset{\text{i.i.d.}}{\sim} q$$

then

$$\begin{aligned} \widehat{Z} &= \frac{1}{N} \sum_{i=1}^{N} w\left(X^{(i)}\right), \\ \widehat{\pi}_N(x) &= \sum_{i=1}^{N} W_i \delta_{X^{(i)}}(x) \text{ where } W_i \propto w\left(X^{(i)}\right), \; \sum_{i=1}^{N} W_i = 1 \end{aligned}$$

Sequential Importance Sampling

- In practice, IS will "work" well if q is close to π; it is difficult to design such a q if $\mathcal{X}$ is an high-dimensional space.

Sequential Importance Sampling

- In practice, IS will "work" well if q is close to π; it is difficult to design such a q if $\mathcal{X}$ is an high-dimensional space.
- A simple way to come up with reasonably good proposal distributions consists of building the proposal sequentially; i.e. if $x = (x_1, ..., x_n)$ then we propose to build an importance distribution of the form

$$q_n(x_{1:n}) = q_1(x_1)\, q_2(x_2 | x_1) \cdots q_n(x_n | x_{1:n-1})$$

Sequential Importance Sampling

- In practice, IS will "work" well if q is close to π; it is difficult to design such a q if $\mathcal{X}$ is an high-dimensional space.
- A simple way to come up with reasonably good proposal distributions consists of building the proposal sequentially; i.e. if $x = (x_1, ..., x_n)$ then we propose to build an importance distribution of the form

$$q_n(x_{1:n}) = q_1(x_1)\, q_2(x_2 | x_1) \cdots q_n(x_n | x_{1:n-1})$$

- The advantage of this approach is that we've broken up the original design problem in n "simpler" models.

Sequential Importance Sampling

- In practice, IS will "work" well if q is close to π; it is difficult to design such a q if $\mathcal{X}$ is an high-dimensional space.
- A simple way to come up with reasonably good proposal distributions consists of building the proposal sequentially; i.e. if $x = (x_1, ..., x_n)$ then we propose to build an importance distribution of the form

$$q_n(x_{1:n}) = q_1(x_1)\, q_2(x_2|x_1) \cdots q_n(x_n|x_{1:n-1})$$

- The advantage of this approach is that we've broken up the original design problem in n "simpler" models.
- Given the fact that

$$\pi(x_{1:n}) = \pi_n(x_{1:n}) = \pi_n(x_1)\, \pi_n(x_2|x_1) \cdots \pi_n(x_n|x_{1:n-1}),$$

where $\pi_n(x_k|x_{1:k-1}) \propto \gamma_n(x_k|x_{1:k-1})$ it seems sensible to take

$$q_k(x_k|x_{1:k-1}) \approx \pi_n(x_k|x_{1:k-1}).$$

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.
- At time $k \geq 2$

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.
- At time $k \geq 2$
  - sample $X_k^{(i)} \sim q_k\left(\cdot \middle| X_{1:k-1}^{(i)}\right)$

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.
- At time $k \geq 2$
  - sample $X_k^{(i)} \sim q_k\left(\cdot \middle| X_{1:k-1}^{(i)}\right)$
  - compute $w_k\left(X_{1:k}^{(i)}\right) = w_{k-1}\left(X_{1:k-1}^{(i)}\right) \dfrac{\gamma_n\left(X_k^{(i)} \middle| X_{1:k-1}^{(i)}\right)}{q_k\left(X_k^{(i)} \middle| X_{1:k-1}^{(i)}\right)}$.

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.
- At time $k \geq 2$
  - sample $X_k^{(i)} \sim q_k\left(\cdot \middle| X_{1:k-1}^{(i)}\right)$
  - compute $w_k\left(X_{1:k}^{(i)}\right) = w_{k-1}\left(X_{1:k-1}^{(i)}\right) \dfrac{\gamma_n\left(X_k^{(i)} \middle| X_{1:k-1}^{(i)}\right)}{q_k\left(X_k^{(i)} \middle| X_{1:k-1}^{(i)}\right)}$.
- Clearly at time n we have obtained $X_{1:n}^{(i)} \sim q_n$ and indeed $w_n\left(X_{1:n}^{(i)}\right) = \dfrac{\gamma_n\left(X_{1:n}^{(i)}\right)}{q_n\left(X_{1:n}^{(i)}\right)}$.

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.
- At time $k \geq 2$
  - sample $X_k^{(i)} \sim q_k\left(\cdot \middle| X_{1:k-1}^{(i)}\right)$
  - compute $w_k\left(X_{1:k}^{(i)}\right) = w_{k-1}\left(X_{1:k-1}^{(i)}\right) \dfrac{\gamma_n\left(X_k^{(i)} \middle| X_{1:k-1}^{(i)}\right)}{q_k\left(X_k^{(i)} \middle| X_{1:k-1}^{(i)}\right)}$.
- Clearly at time n we have obtained $X_{1:n}^{(i)} \sim q_n$ and indeed $w_n\left(X_{1:n}^{(i)}\right) = \dfrac{\gamma_n\left(X_{1:n}^{(i)}\right)}{q_n\left(X_{1:n}^{(i)}\right)}$.
- Although this algorithm is simple, **it typically cannot be implemented** as $\pi_n\left(x_k \middle| x_{1:k-1}\right)$ is unknown even up to a normalizing constant.

- Now consider the following modification where we define a **sequence of intermediate target distributions** $\pi_1(x_1)$, $\pi_2(x_{1:2}), ..., \pi_{n-1}(x_{1:n-1})$ **to move smoothly towards** $\pi_n(x_{1:n})$; that is at each time k we provide an IS approximation of $\pi_k(x_{1:k})$.

- Now consider the following modification where we define a **sequence of intermediate target distributions** $\pi_1(x_1)$, $\pi_2(x_{1:2}), ..., \pi_{n-1}(x_{1:n-1})$ **to move smoothly towards** $\pi_n(x_{1:n})$; that is at each time k we provide an IS approximation of $\pi_k(x_{1:k})$.
- By construction, we know $\pi_k(x_{1:k})$ up to a normalizing constant

$$\pi_k(x_{1:k}) = \frac{\gamma_k(x_{1:k})}{Z_k}.$$

- Now consider the following modification where we define a **sequence of intermediate target distributions** $\pi_1(x_1)$, $\pi_2(x_{1:2}), ..., \pi_{n-1}(x_{1:n-1})$ **to move smoothly towards** $\pi_n(x_{1:n})$; that is at each time k we provide an IS approximation of $\pi_k(x_{1:k})$.
- By construction, we know $\pi_k(x_{1:k})$ up to a normalizing constant

$$\pi_k(x_{1:k}) = \frac{\gamma_k(x_{1:k})}{Z_k}.$$

- We also use an importance distribution

$$\begin{aligned} q_n(x_{1:n}) &= q_1(x_1)\, q_2(x_2|x_1) \cdots q_n(x_n|x_{1:n-1}) \\ &= q_{n-1}(x_{1:n-1})\, q_n(x_n|x_{1:n-1}) \\ &= q_k(x_{1:k}) \prod_{j=k+1}^{n} q_j(x_j|x_{1:j-1}) \end{aligned}$$

but it is now such that

$$q_k(x_k|x_{1:k-1}) \approx \pi_k(x_k|x_{1:k-1}).$$

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.
- At time $k \geq 2$

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.
- At time $k \geq 2$
  - sample $X_k^{(i)} \sim q_k\left(\cdot \mid X_{1:k-1}^{(i)}\right)$

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.
- At time $k \geq 2$
  - sample $X_k^{(i)} \sim q_k\left(\cdot \middle| X_{1:k-1}^{(i)}\right)$
  - compute $w_k\left(X_{1:k}^{(i)}\right) = w_{k-1}\left(X_{1:k-1}^{(i)}\right) \dfrac{\gamma_k\left(X_{1:k}^{(i)}\right)}{\gamma_{k-1}\left(X_{1:k-1}^{(i)}\right) q_k\left(X_k^{(i)} \middle| X_{1:k-1}^{(i)}\right)}$.

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.
- At time $k \geq 2$
  - sample $X_k^{(i)} \sim q_k\left(\cdot \middle| X_{1:k-1}^{(i)}\right)$
  - compute $w_k\left(X_{1:k}^{(i)}\right) = w_{k-1}\left(X_{1:k-1}^{(i)}\right) \dfrac{\gamma_k\left(X_{1:k}^{(i)}\right)}{\gamma_{k-1}\left(X_{1:k-1}^{(i)}\right) q_k\left(X_k^{(i)} \middle| X_{1:k-1}^{(i)}\right)}$.
- At any time k, we have

$$X_{1:k}^{(i)} \sim q_k(x_{1:k}), \quad w_k\left(X_{1:k}^{(i)}\right) = \frac{\gamma_k\left(X_{1:k}^{(i)}\right)}{q_k\left(X_{1:k}^{(i)}\right)}$$

that is an IS approximation of $\pi_k(x_{1:k})$ and of Z_k.

- To check that is indeed true, note that

$$w_1(x_1) = \frac{\gamma_1(x_1)}{q_1(x_1)}$$

and

$$\begin{aligned} w_k(x_{1:k}) &= \frac{\gamma_1(x_1)}{q_1(x_1)} \prod_{j=1}^{k} \frac{\gamma_j(x_{1:j})}{\gamma_{j-1}(x_{1:j-1})\, q_j(x_j | x_{1:j-1})} \\ &= \frac{\gamma_k(x_{1:k})}{q_1(x_1)\, q_2(x_2|x_1) \cdots q_k(x_k|x_{1:k-1})} \\ &= \frac{\gamma_k(x_{1:k})}{q_k(x_{1:k})} \end{aligned}$$

- To check that is indeed true, note that

$$w_1(x_1) = \frac{\gamma_1(x_1)}{q_1(x_1)}$$

and

$$\begin{aligned} w_k(x_{1:k}) &= \frac{\gamma_1(x_1)}{q_1(x_1)} \prod_{j=1}^{k} \frac{\gamma_j(x_{1:j})}{\gamma_{j-1}(x_{1:j-1})\, q_j(x_j | x_{1:j-1})} \\ &= \frac{\gamma_k(x_{1:k})}{q_1(x_1)\, q_2(x_2 | x_1) \cdots q_k(x_k | x_{1:k-1})} \\ &= \frac{\gamma_k(x_{1:k})}{q_k(x_{1:k})} \end{aligned}$$

- A key problem remains to be solved, how to select $\pi_k(x_{1:k})$?

- **Example**: Bayesian inference for hidden Markov models

$$\text{Hidden Markov process: } X_1 \sim \mu, \;\; X_k \left| \left(X_{k-1} = x_{k-1} \right) \right. \sim f\left(\cdot \left| \, x_{k-1} \right. \right)$$

$$\text{Observation process: } \;\; Y_k \left| \left(X_k = x_k \right) \right. \sim g\left(\cdot \left| \, x_k \right. \right)$$

- **Example**: Bayesian inference for hidden Markov models

$$\text{Hidden Markov process: } X_1 \sim \mu, \;\; X_k \left| \left(X_{k-1} = x_{k-1} \right) \right. \sim f\left(\cdot \left| \, x_{k-1} \right. \right)$$

$$\text{Observation process: } \;\; Y_k \left| \left(X_k = x_k \right) \right. \sim g\left(\cdot \left| \, x_k \right. \right)$$

- This class of models appears in numerous areas: statistics, vision, robotics, econometrics, tracking etc.

- **Example**: Bayesian inference for hidden Markov models

$$\text{Hidden Markov process: } X_1 \sim \mu, \;\; X_k \left| \left(X_{k-1} = x_{k-1} \right) \right. \sim f\left(\cdot \left| x_{k-1} \right. \right)$$

$$\text{Observation process: } \;\; Y_k \left| \left(X_k = x_k \right) \right. \sim g\left(\cdot \left| x_k \right. \right)$$

- This class of models appears in numerous areas: statistics, vision, robotics, econometrics, tracking etc.
- Assume we receive $y_{1:n}$, we are interested in sampling from

$$\pi_n\left(x_{1:n}\right) = p\left(x_{1:n} \left| y_{1:n}\right.\right) = \frac{p\left(x_{1:n}, y_{1:n}\right)}{p\left(y_{1:n}\right)}$$

and estimating $p\left(y_{1:n}\right)$ where

$$\gamma_n\left(x_{1:n}\right) = p\left(x_{1:n}, y_{1:n}\right) = \mu\left(x_1\right) \prod_{k=2}^{n} f\left(x_k \left| x_{k-1}\right.\right) \prod_{k=1}^{n} g\left(y_k \left| x_k\right.\right),$$

$$Z_n = p\left(y_{1:n}\right) = \int \cdots \int \mu\left(x_1\right) \prod_{k=2}^{n} f\left(x_k \left| x_{k-1}\right.\right) \prod_{k=1}^{n} g\left(y_k \left| x_k\right.\right) dx_{1:n}.$$

Sequential Importance Sampling for Hidden Markov Models

- If we are interested only in $p\left(x_{1:n} \middle| y_{1:n}\right)$ for a **fixed** n, then the "best" sequential strategy would be to construct importance distributions

$$q_k\left(x_k \middle| x_{1:k-1}\right) \approx \pi_n\left(x_k \middle| x_{1:k-1}\right) = p\left(x_k \middle| y_{k:n}, x_{k-1}\right).$$

This is typically impossible because $p\left(x_k \middle| y_{k:n}, x_{k-1}\right)$ is unknown even up to a normalizing constant

$$p\left(x_k \middle| y_{k:n}, x_{k-1}\right) \propto f\left(x_k \middle| x_{k-1}\right) p\left(y_{k:n} \middle| x_k\right)$$

where

$$p\left(y_{k:n} \middle| x_k\right) = \int \cdots \int \prod_{j=k+1}^{n} f\left(x_j \middle| x_{j-1}\right) \prod_{j=k}^{n} g\left(y_j \middle| x_j\right) dx_{k+1:n}$$

Sequential Importance Sampling for Hidden Markov Models

- If we are interested only in $p\left(x_{1:n}\middle|y_{1:n}\right)$ for a **fixed** n, then the "best" sequential strategy would be to construct importance distributions

$$q_k\left(x_k\middle|x_{1:k-1}\right) \approx \pi_n\left(x_k\middle|x_{1:k-1}\right) = p\left(x_k\middle|y_{k:n}, x_{k-1}\right).$$

  This is typically impossible because $p\left(x_k\middle|y_{k:n}, x_{k-1}\right)$ is unknown even up to a normalizing constant

$$p\left(x_k\middle|y_{k:n}, x_{k-1}\right) \propto f\left(x_k\middle|x_{k-1}\right) p\left(y_{k:n}\middle|x_k\right)$$

  where

$$p\left(y_{k:n}\middle|x_k\right) = \int \cdots \int \prod_{j=k+1}^{n} f\left(x_j\middle|x_{j-1}\right) \prod_{j=k}^{n} g\left(y_j\middle|x_j\right) dx_{k+1:n}$$

- Alternatively, we can simply propose to sample from an intermediate sequence of distributions $\pi_k\left(x_{1:k}\right)$. In this context, a natural choice consists of using

$$\pi_k\left(x_{1:k}\right) = p\left(x_{1:k}\middle|y_{1:k}\right).$$

  This is only one possibility but very important in practice.

- We pick the proposal distributions such that

$$\begin{aligned} q_k\left(x_k \middle| x_{1:k-1}\right) &\approx \pi_k\left(x_k \middle| x_{1:k-1}\right) = p\left(x_k \middle| y_k, x_{k-1}\right) \\ &\propto f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right). \end{aligned}$$

- We pick the proposal distributions such that

$$\begin{aligned} q_k\left(x_k \middle| x_{1:k-1}\right) &\approx \pi_k\left(x_k \middle| x_{1:k-1}\right) = p\left(x_k \middle| y_k, x_{k-1}\right) \\ &\propto f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right). \end{aligned}$$

- We will use the notation $q\left(x_k \middle| y_k, x_{k-1}\right)$ in this context and

$$q\left(x_{1:k} \middle| y_{1:k}\right) = q\left(x_1 \middle| y_1\right) q\left(x_2 \middle| y_2, x_1\right) \cdots q\left(x_k \middle| y_k, x_{k-1}\right)$$

- We pick the proposal distributions such that

$$\begin{aligned} q_k\left(x_k \middle| x_{1:k-1}\right) &\approx \pi_k\left(x_k \middle| x_{1:k-1}\right) = p\left(x_k \middle| y_k, x_{k-1}\right) \\ &\propto f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right). \end{aligned}$$

- We will use the notation $q\left(x_k \middle| y_k, x_{k-1}\right)$ in this context and

$$q\left(x_{1:k} \middle| y_{1:k}\right) = q\left(x_1 \middle| y_1\right) q\left(x_2 \middle| y_2, x_1\right) \cdots q\left(x_k \middle| y_k, x_{k-1}\right)$$

- Note that we will sample $X_k^{(i)}$ using only the observation y_k available at time k.

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set
$$w_1\left(X_1^{(i)}\right) = \frac{\mu\left(X_1^{(i)}\right) g\left(y_1 | X_1^{(i)}\right)}{q\left(X_1^{(i)} \middle| y_1\right)}.$$

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set
$$w_1\left(X_1^{(i)}\right) = \frac{\mu\left(X_1^{(i)}\right) g\left(y_1 | X_1^{(i)}\right)}{q\left(X_1^{(i)} \middle| y_1\right)}.$$
- At time $k \geq 2$

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set
  $w_1\left(X_1^{(i)}\right) = \frac{\mu\left(X_1^{(i)}\right) g\left(y_1 | X_1^{(i)}\right)}{q\left(X_1^{(i)} \middle| y_1\right)}$.
- At time $k \geq 2$
  - sample $X_k^{(i)} \sim q\left(\cdot \,|\, y_k, X_{k-1}^{(i)}\right)$

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set
$$w_1\left(X_1^{(i)}\right) = \frac{\mu\left(X_1^{(i)}\right) g\left(y_1 | X_1^{(i)}\right)}{q\left(X_1^{(i)} \middle| y_1\right)}.$$
- At time $k \geq 2$
  - sample $X_k^{(i)} \sim q\left(\cdot \middle| y_k, X_{k-1}^{(i)}\right)$
  - compute
$$w_k\left(X_{1:k}^{(i)}\right) = w_{k-1}\left(X_{1:k-1}^{(i)}\right) \frac{p\left(X_{1:k}^{(i)}, y_{1:k}\right)}{p\left(X_{1:k-1}^{(i)}, y_{1:k-1}\right) q\left(X_k^{(i)} \middle| y_k, X_{k-1}^{(i)}\right)} =$$
$$w_{k-1}\left(X_{1:k-1}^{(i)}\right) \frac{f\left(X_k^{(i)} \middle| X_{k-1}^{(i)}\right) g\left(y_k | X_k^{(i)}\right)}{q\left(X_k^{(i)} \middle| y_k, X_{k-1}^{(i)}\right)}.$$

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set
$$w_1\left(X_1^{(i)}\right) = \frac{\mu\left(X_1^{(i)}\right) g\left(y_1 | X_1^{(i)}\right)}{q\left(X_1^{(i)} \middle| y_1\right)}.$$
- At time $k \geq 2$
  - sample $X_k^{(i)} \sim q\left(\cdot \middle| y_k, X_{k-1}^{(i)}\right)$
  - compute
$$w_k\left(X_{1:k}^{(i)}\right) = w_{k-1}\left(X_{1:k-1}^{(i)}\right) \frac{p\left(X_{1:k}^{(i)}, y_{1:k}\right)}{p\left(X_{1:k-1}^{(i)}, y_{1:k-1}\right) q\left(X_k^{(i)} \middle| y_k, X_{k-1}^{(i)}\right)} =$$
$$w_{k-1}\left(X_{1:k-1}^{(i)}\right) \frac{f\left(X_k^{(i)} \middle| X_{k-1}^{(i)}\right) g\left(y_k | X_k^{(i)}\right)}{q\left(X_k^{(i)} \middle| y_k, X_{k-1}^{(i)}\right)}.$$
- At any time k, we have
$$X_{1:k}^{(i)} \sim q\left(x_{1:k} \middle| y_{1:k}\right), \quad w_k\left(X_{1:k}^{(i)}\right) = \frac{p\left(X_{1:k}^{(i)}, y_{1:k}\right)}{q\left(X_{1:k}^{(i)} \middle| y_{1:k}\right)}$$
that is an IS approximation of $\pi_k\left(x_{1:k}\right) = p\left(x_{1:k} \middle| y_{1:k}\right)$ and of $Z_k = p\left(y_{1:k}\right)$.

- This algorithm provides an approximation of ALL the distributions $p\left(x_{1:k} \middle| y_{1:k}\right)$ for any $k \geq 1$.

- This algorithm provides an approximation of ALL the distributions $p\left(x_{1:k} \middle| y_{1:k}\right)$ for any $k \geq 1$.
- It can be implemented for real time applications.

- This algorithm provides an approximation of ALL the distributions $p\left(x_{1:k} \middle| y_{1:k}\right)$ for any $k \geq 1$.
- It can be implemented for real time applications.
- The computational complexity at each time step is fixed.

- A "locally" optimal choice consists of selecting

$$q\left(x_k \middle| y_{1:k}, x_{k-1}\right) = p\left(x_k \middle| y_k, x_{k-1}\right) = \frac{f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right)}{\int f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right) dx_k}$$

which yields

$$\frac{f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right)}{q\left(x_k \middle| y_{1:k}, x_{k-1}\right)} = \int f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right) dx_k .$$

- A "locally" optimal choice consists of selecting

$$q\left(x_k \middle| y_{1:k}, x_{k-1}\right) = p\left(x_k \middle| y_k, x_{k-1}\right) = \frac{f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right)}{\int f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right) dx_k}$$

  which yields

$$\frac{f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right)}{q\left(x_k \middle| y_{1:k}, x_{k-1}\right)} = \int f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right) dx_k.$$

- We might not be able to compute $\int f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right) dx_k$ so we can either get an unbiased estimate of it or approximate $p\left(x_k \middle| y_k, x_{k-1}\right)$ using standard techniques (EKF, Unscented...).

- A "locally" optimal choice consists of selecting

$$q\left(x_k \middle| y_{1:k}, x_{k-1}\right) = p\left(x_k \middle| y_k, x_{k-1}\right) = \frac{f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right)}{\int f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right) dx_k}$$

which yields

$$\frac{f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right)}{q\left(x_k \middle| y_{1:k}, x_{k-1}\right)} = \int f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right) dx_k.$$

- We might not be able to compute $\int f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right) dx_k$ so we can either get an unbiased estimate of it or approximate $p\left(x_k \middle| y_k, x_{k-1}\right)$ using standard techniques (EKF, Unscented...).
- The lazy user can simply select

$$q\left(x_k \middle| y_{1:k}, x_{k-1}\right) = p\left(x_k \middle| x_{k-1}\right)$$

which yields

$$\frac{f\left(x_k \middle| x_{k-1}\right) g\left(y_k \middle| x_k\right)}{q\left(x_k \middle| y_{1:k}, x_{k-1}\right)} = g\left(y_k \middle| x_k\right).$$

- We present a simple application to stochastic volatility model where

$$\begin{aligned} f\left(x_k \middle| x_{k-1}\right) &= \mathcal{N}\left(x_k; \phi x_{k-1}, \sigma^2\right), \\ g\left(y_k \middle| x_k\right) &= \mathcal{N}\left(y_k; 0, \beta^2 \exp\left(x_k\right)\right). \end{aligned}$$

- We present a simple application to stochastic volatility model where

$$\begin{aligned} f\left(x_k \middle| x_{k-1}\right) &= \mathcal{N}\left(x_k; \phi x_{k-1}, \sigma^2\right), \\ g\left(y_k \middle| x_k\right) &= \mathcal{N}\left(y_k; 0, \beta^2 \exp\left(x_k\right)\right). \end{aligned}$$

- We cannot sample from $p\left(x_k \middle| y_k, x_{k-1}\right)$ but it is unimodal and we can compute numerically its mode $m_k\left(x_{k-1}\right)$ and use a $t-$distribution with 5 degrees of freedom and scale set as the inverse of the negated second-order of $\log p\left(x_k \middle| y_k, x_{k-1}\right)$ evaluated at $m_k\left(x_{k-1}\right)$ and given by

$$\sigma_k^2\left(x_{k-1}\right) = \left(\frac{1}{\sigma^2} + \frac{y_k^2}{2\beta^2} \exp\left(-m_k\left(x_{k-1}\right)\right)\right)^{-1}.$$

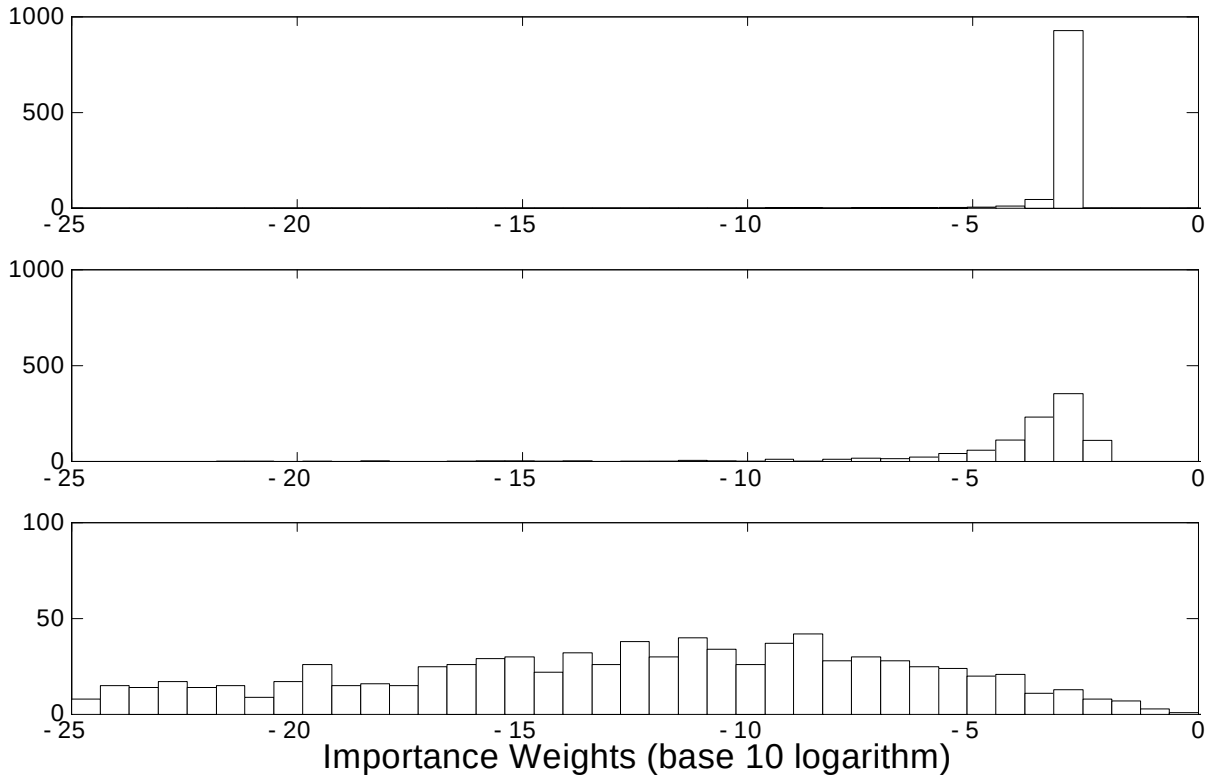

Figure: Histograms of the base 10 logarithm of $W_n^{(i)}$ for $n = 1$ (top), $n = 50$ (middle) and $n = 100$ (bottom).

- The algorithm performance collapse as n increases... After a few time steps, only a very small number of particles have non negligible weights.

- You should not be surprised: This algorithm is nothing but an implementation of IS where we have restricted the structure of the importance distribution.

- You should not be surprised: This algorithm is nothing but an implementation of IS where we have restricted the structure of the importance distribution.
- As the dimension of the target $p\left(x_{1:n} \middle| y_{1:n}\right)$ increases over time, the problem is becoming increasingly difficult. In practice, the discrepancy between the target and the IS distribution $q\left(x_{1:n} \middle| y_{1:n}\right)$ can only also increase (on average).

- You should not be surprised: This algorithm is nothing but an implementation of IS where we have restricted the structure of the importance distribution.
- As the dimension of the target $p\left(x_{1:n} \middle| y_{1:n}\right)$ increases over time, the problem is becoming increasingly difficult. In practice, the discrepancy between the target and the IS distribution $q\left(x_{1:n} \middle| y_{1:n}\right)$ can only also increase (on average).
- As n increases the variance of the weights increases (typically geometrically) and the IS approximation collapses.

- You should not be surprised: This algorithm is nothing but an implementation of IS where we have restricted the structure of the importance distribution.
- As the dimension of the target $p\left(x_{1:n} \middle| y_{1:n}\right)$ increases over time, the problem is becoming increasingly difficult. In practice, the discrepancy between the target and the IS distribution $q\left(x_{1:n} \middle| y_{1:n}\right)$ can only also increase (on average).
- As n increases the variance of the weights increases (typically geometrically) and the IS approximation collapses.
- You can use any IS distribution you want (even the locally optimal one), the algorithm will collapse.

- You should not be surprised: This algorithm is nothing but an implementation of IS where we have restricted the structure of the importance distribution.
- As the dimension of the target $p\left(x_{1:n} \middle| y_{1:n}\right)$ increases over time, the problem is becoming increasingly difficult. In practice, the discrepancy between the target and the IS distribution $q\left(x_{1:n} \middle| y_{1:n}\right)$ can only also increase (on average).
- As n increases the variance of the weights increases (typically geometrically) and the IS approximation collapses.
- You can use any IS distribution you want (even the locally optimal one), the algorithm will collapse.
- These negative remarks also hold for the general case and not only for hidden Markov models.

Summary

- Sequential Importance Sampling is an attractive idea: sequential and parallelizable, only requires designing low-dimensional proposal distributions.

Summary

- Sequential Importance Sampling is an attractive idea: sequential and parallelizable, only requires designing low-dimensional proposal distributions.
- Sequential Importance Sampling can only work for moderate size problems.

Summary

- Sequential Importance Sampling is an attractive idea: sequential and parallelizable, only requires designing low-dimensional proposal distributions.
- Sequential Importance Sampling can only work for moderate size problems.
- Is there a way to **partially** fix this problem?

Resampling

- *Intuitive KEY idea*: As the time index k increases, the variance of the unnormalized weights $\left\{ w_k \left(X_{1:k}^{(i)} \right) \right\}$ increases and all the mass is concentrated on a few random samples/particles. We propose to reset the approximation by getting rid in a principled way of the particles with low weights $W_k^{(i)}$ (relative to $1/N$) and multiply the particles with high weights $W_k^{(i)}$ (relative to $1/N$).

Resampling

- *Intuitive KEY idea*: As the time index k increases, the variance of the unnormalized weights $\left\{ w_k \left(X_{1:k}^{(i)} \right) \right\}$ increases and all the mass is concentrated on a few random samples/particles. We propose to reset the approximation by getting rid in a principled way of the particles with low weights $W_k^{(i)}$ (relative to $1/N$) and multiply the particles with high weights $W_k^{(i)}$ (relative to $1/N$).
- The main reason is that if a particle at time n has a low weight then typically it will still have a low weight at time $n+1$ (though I can easily give you a counterexample).

Resampling

- *Intuitive KEY idea*: As the time index k increases, the variance of the unnormalized weights $\left\{ w_k \left(X_{1:k}^{(i)} \right) \right\}$ increases and all the mass is concentrated on a few random samples/particles. We propose to reset the approximation by getting rid in a principled way of the particles with low weights $W_k^{(i)}$ (relative to $1/N$) and multiply the particles with high weights $W_k^{(i)}$ (relative to $1/N$).
- The main reason is that if a particle at time n has a low weight then typically it will still have a low weight at time $n+1$ (though I can easily give you a counterexample).
- You want to focus your computational efforts on the "promising" parts of the space.

- At time k, IS provides the following approximation of $\pi_k\left(x_{1:k}\right)$

$$\widehat{\pi}_k\left(x_{1:k}\right) = \sum_{i=1}^{N} W_k^{(i)} \delta_{X_{1:k}^{(i)}}\left(x_{1:k}\right).$$

- At time k, IS provides the following approximation of $\pi_k\left(x_{1:k}\right)$

$$\widehat{\pi}_k\left(x_{1:k}\right) = \sum_{i=1}^{N} W_k^{(i)} \delta_{X_{1:k}^{(i)}}\left(x_{1:k}\right).$$

- The simplest resampling schemes consists of sampling N times $\widetilde{X}_{1:k}^{(i)} \sim \widehat{\pi}_k\left(x_{1:k}\right)$ to build the new approximation

$$\widetilde{\pi}_k\left(x_{1:k}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\widetilde{X}_{1:k}^{(i)}}\left(x_{1:k}\right).$$

- At time k, IS provides the following approximation of $\pi_k\left(x_{1:k}\right)$

$$\widehat{\pi}_k\left(x_{1:k}\right) = \sum_{i=1}^{N} W_k^{(i)} \delta_{X_{1:k}^{(i)}}\left(x_{1:k}\right).$$

- The simplest resampling schemes consists of sampling N times $\widetilde{X}_{1:k}^{(i)} \sim \widehat{\pi}_k\left(x_{1:k}\right)$ to build the new approximation

$$\widetilde{\pi}_k\left(x_{1:k}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\widetilde{X}_{1:k}^{(i)}}\left(x_{1:k}\right).$$

- The new resampled particles $\left\{\widetilde{X}_{1:k}^{(i)}\right\}$ are approximately distributed according to $\pi_k\left(x_{1:k}\right)$ but statistically dependent. This is theoretically much more difficult to study.

- Note that we can rewrite

$$\widetilde{\pi}_k\left(x_{1:k}\right) = \sum_{i=1}^{N} \frac{N_k^{(i)}}{N} \delta_{X_{1:k}^{(i)}}\left(x_{1:k}\right)$$

where $\left(N_k^{(1)}, ..., N_k^{(N)}\right) \sim \mathcal{M}\left(N; W_k^{(1)}, ..., W_k^{(N)}\right)$ thus $\mathbb{E}\left[N_k^{(i)}\right] = NW_k^{(i)}$, $var\left[N_k^{(1)}\right] = NW_k^{(i)}\left(1 - W_k^{(i)}\right)$.

- Note that we can rewrite

$$\widetilde{\pi}_k\left(x_{1:k}\right) = \sum_{i=1}^{N} \frac{N_k^{(i)}}{N} \delta_{X_{1:k}^{(i)}}\left(x_{1:k}\right)$$

where $\left(N_k^{(1)}, ..., N_k^{(N)}\right) \sim \mathcal{M}\left(N; W_k^{(1)}, ..., W_k^{(N)}\right)$ thus $\mathbb{E}\left[N_k^{(i)}\right] = NW_k^{(i)}$, $var\left[N_k^{(1)}\right] = NW_k^{(i)}\left(1 - W_k^{(i)}\right)$.

- It follows that the resampling step is an unbiased operation

$$\mathbb{E}\left[\widetilde{\pi}_k\left(x_{1:k}\right) \middle| \widehat{\pi}_k\left(x_{1:k}\right)\right] = \widehat{\pi}_k\left(x_{1:k}\right)$$

but clearly it introduces some errors "locally" in time. That is for any test function, we have

$$var_{\widetilde{\pi}_k}\left[\varphi\left(X_{1:k}\right)\right] \geq var_{\widehat{\pi}_k}\left[\varphi\left(X_{1:k}\right)\right]$$

- Note that we can rewrite

$$\widetilde{\pi}_k\left(x_{1:k}\right) = \sum_{i=1}^{N} \frac{N_k^{(i)}}{N} \delta_{X_{1:k}^{(i)}}\left(x_{1:k}\right)$$

where $\left(N_k^{(1)}, ..., N_k^{(N)}\right) \sim \mathcal{M}\left(N; W_k^{(1)}, ..., W_k^{(N)}\right)$ thus $\mathbb{E}\left[N_k^{(i)}\right] = NW_k^{(i)}$, $var\left[N_k^{(1)}\right] = NW_k^{(i)}\left(1 - W_k^{(i)}\right)$.

- It follows that the resampling step is an unbiased operation

$$\mathbb{E}\left[\widetilde{\pi}_k\left(x_{1:k}\right) \middle| \widehat{\pi}_k\left(x_{1:k}\right)\right] = \widehat{\pi}_k\left(x_{1:k}\right)$$

but clearly it introduces some errors "locally" in time. That is for any test function, we have

$$var_{\widetilde{\pi}_k}\left[\varphi\left(X_{1:k}\right)\right] \geq var_{\widehat{\pi}_k}\left[\varphi\left(X_{1:k}\right)\right]$$

- Better resampling steps can be designed such that $\mathbb{E}\left[N_k^{(i)}\right] = NW_k^{(i)}$ but $var\left[N_k^{(i)}\right] < NW_k^{(i)}\left(1 - W_k^{(i)}\right)$.

- Note that we can rewrite

$$\widetilde{\pi}_k\left(x_{1:k}\right) = \sum_{i=1}^{N} \frac{N_k^{(i)}}{N} \delta_{X_{1:k}^{(i)}}\left(x_{1:k}\right)$$

where $\left(N_k^{(1)}, ..., N_k^{(N)}\right) \sim \mathcal{M}\left(N; W_k^{(1)}, ..., W_k^{(N)}\right)$ thus $\mathbb{E}\left[N_k^{(i)}\right] = NW_k^{(i)}$, $var\left[N_k^{(1)}\right] = NW_k^{(i)}\left(1 - W_k^{(i)}\right)$.

- It follows that the resampling step is an unbiased operation

$$\mathbb{E}\left[\widetilde{\pi}_k\left(x_{1:k}\right) \middle| \widehat{\pi}_k\left(x_{1:k}\right)\right] = \widehat{\pi}_k\left(x_{1:k}\right)$$

but clearly it introduces some errors "locally" in time. That is for any test function, we have

$$var_{\widetilde{\pi}_k}\left[\varphi\left(X_{1:k}\right)\right] \geq var_{\widehat{\pi}_k}\left[\varphi\left(X_{1:k}\right)\right]$$

- Better resampling steps can be designed such that $\mathbb{E}\left[N_k^{(i)}\right] = NW_k^{(i)}$ but $var\left[N_k^{(i)}\right] < NW_k^{(i)}\left(1 - W_k^{(i)}\right)$.
- Resampling is beneficial for future time steps.

- A popular alternative to multinomial resampling consists of selecting

$$U_1 \sim \mathcal{U}\left[0, \frac{1}{N}\right]$$

and for $i = 2, ..., N$

$$U_i = U_1 + \frac{i-1}{N} = U_{i-1} + \frac{1}{N}.$$

- A popular alternative to multinomial resampling consists of selecting

$$U_1 \sim \mathcal{U}\left[0, \frac{1}{N}\right]$$

and for $i = 2, ..., N$

$$U_i = U_1 + \frac{i-1}{N} = U_{i-1} + \frac{1}{N}.$$

- Then we set

$$N_k^{(i)} = \#\left\{ U_j : \sum_{m=1}^{i-1} W_k^{(m)} \leq U_j < \sum_{m=1}^{i} W_k^{(m)} \right\}$$

where $\sum_{m=1}^{0} = 0$.

- A popular alternative to multinomial resampling consists of selecting

$$U_1 \sim \mathcal{U}\left[0, \frac{1}{N}\right]$$

and for $i = 2, ..., N$

$$U_i = U_1 + \frac{i-1}{N} = U_{i-1} + \frac{1}{N}.$$

- Then we set

$$N_k^{(i)} = \#\left\{ U_j : \sum_{m=1}^{i-1} W_k^{(m)} \leq U_j < \sum_{m=1}^{i} W_k^{(m)} \right\}$$

where $\sum_{m=1}^{0} = 0$.

- It is trivial to check that $\mathbb{E}\left[N_k^{(i)}\right] = NW_k^{(i)}$.

Degeneracy Measures

- Resampling at each time step is harmful. We should resample only when necessary.

Degeneracy Measures

- Resampling at each time step is harmful. We should resample only when necessary.
- To measure the variation of the weights, we can use the Effective Sample Size (ESS) or the coefficient of variation CV

$$ESS = \left(\sum_{i=1}^{N} \left(W_n^{(i)} \right)^2 \right)^{-1}, \quad CV = \left(\frac{1}{N} \sum_{i=1}^{N} \left(N W_n^{(i)} - 1 \right)^2 \right)^{1/2}$$

Degeneracy Measures

- Resampling at each time step is harmful. We should resample only when necessary.
- To measure the variation of the weights, we can use the Effective Sample Size (ESS) or the coefficient of variation CV

$$ESS = \left(\sum_{i=1}^{N} \left(W_n^{(i)} \right)^2 \right)^{-1}, \quad CV = \left(\frac{1}{N} \sum_{i=1}^{N} \left(N W_n^{(i)} - 1 \right)^2 \right)^{1/2}$$

- We have $ESS = N$ and $CV = 0$ if $W_n^{(i)} = 1/N$ for any i.

Degeneracy Measures

- Resampling at each time step is harmful. We should resample only when necessary.
- To measure the variation of the weights, we can use the Effective Sample Size (ESS) or the coefficient of variation CV

$$ESS = \left(\sum_{i=1}^{N} \left(W_n^{(i)} \right)^2 \right)^{-1}, \; CV = \left(\frac{1}{N} \sum_{i=1}^{N} \left(N W_n^{(i)} - 1 \right)^2 \right)^{1/2}$$

- We have $ESS = N$ and $CV = 0$ if $W_n^{(i)} = 1/N$ for any i.
- We have $ESS = 1$ and $CV = \sqrt{N-1}$ if $W_n^{(i)} = 1$ and $W_n^{(j)} = 1$ for $j \neq i$.

- We can also use the entropy

$$Ent = -\sum_{i=1}^{N} W_n^{(i)} \log_2 \left(W_n^{(i)} \right)$$

- We can also use the entropy

$$Ent = -\sum_{i=1}^{N} W_n^{(i)} \log_2 \left(W_n^{(i)} \right)$$

- We have $Ent = \log_2 (N)$ if $W_n^{(i)} = 1/N$ for any i. We have $Ent = 0$ if $W_n^{(i)} = 1$ and $W_n^{(j)} = 1$ for $j \neq i$.

- We can also use the entropy

$$Ent = -\sum_{i=1}^{N} W_n^{(i)} \log_2 \left(W_n^{(i)} \right)$$

- We have $Ent = \log_2 (N)$ if $W_n^{(i)} = 1/N$ for any i. We have $Ent = 0$ if $W_n^{(i)} = 1$ and $W_n^{(j)} = 1$ for $j \neq i$.
- **Dynamic Resampling**: If the variation of the weights as measured by ESS, CV or Ent is too high, then resample the particles.

Generic Sequential Monte Carlo Scheme

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.

Generic Sequential Monte Carlo Scheme

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.
- Resample $\left\{X_1^{(i)}, W_1^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_1^{(i)}\right\}$

Generic Sequential Monte Carlo Scheme

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.
- Resample $\left\{X_1^{(i)}, W_1^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_1^{(i)}\right\}$
- At time $k \geq 2$

Generic Sequential Monte Carlo Scheme

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.
- Resample $\left\{X_1^{(i)}, W_1^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_1^{(i)}\right\}$
- At time $k \geq 2$
  - sample $X_k^{(i)} \sim q_k\left(\cdot | X_{1:k-1}^{(i)}\right)$

Generic Sequential Monte Carlo Scheme

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.
- Resample $\left\{X_1^{(i)}, W_1^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_1^{(i)}\right\}$
- At time $k \geq 2$
  - sample $X_k^{(i)} \sim q_k\left(\cdot \middle| X_{1:k-1}^{(i)}\right)$
  - compute $w_k\left(X_{1:k}^{(i)}\right) = \dfrac{\gamma_k\left(X_{1:k}^{(i)}\right)}{\gamma_{k-1}\left(X_{1:k-1}^{(i)}\right) q_k\left(X_k^{(i)} \middle| X_{1:k-1}^{(i)}\right)}$.

Generic Sequential Monte Carlo Scheme

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.
- Resample $\left\{X_1^{(i)}, W_1^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_1^{(i)}\right\}$
- At time $k \geq 2$
  - sample $X_k^{(i)} \sim q_k\left(\cdot \middle| X_{1:k-1}^{(i)}\right)$
  - compute $w_k\left(X_{1:k}^{(i)}\right) = \dfrac{\gamma_k\left(X_{1:k}^{(i)}\right)}{\gamma_{k-1}\left(X_{1:k-1}^{(i)}\right) q_k\left(X_k^{(i)} \middle| X_{1:k-1}^{(i)}\right)}$.
- Resample $\left\{X_{1:k}^{(i)}, W_k^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_{1:k}^{(i)}\right\}$

- At any time k, we have two approximation of $\pi_k\left(x_{1:k}\right)$

$$\begin{aligned}
\widehat{\pi}_k\left(x_{1:k}\right) &= \sum_{i=1}^{N} W_k^{(i)} \delta_{X_{1:k}^{(i)}}\left(x_{1:k}\right) \text{ (before resampling)} \\
\widetilde{\pi}_k\left(x_{1:k}\right) &= \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:k}^{(i)}}\left(x_{1:k}\right) \text{ (after resampling).}
\end{aligned}$$

- At any time k, we have two approximation of $\pi_k\left(x_{1:k}\right)$

$$\begin{aligned}\widehat{\pi}_k\left(x_{1:k}\right) &= \sum_{i=1}^{N} W_k^{(i)} \delta_{X_{1:k}^{(i)}}\left(x_{1:k}\right) \text{ (before resampling)} \\ \widetilde{\pi}_k\left(x_{1:k}\right) &= \frac{1}{N}\sum_{i=1}^{N} \delta_{X_{1:k}^{(i)}}\left(x_{1:k}\right) \text{ (after resampling).}\end{aligned}$$

- We also have

$$\frac{\widehat{Z_k}}{Z_{k-1}} = \frac{1}{N}\sum_{i=1}^{N} w_k\left(X_{1:k}^{(i)}\right).$$

Sequential Monte Carlo for Hidden Markov Models

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set
$$w_1\left(X_1^{(i)}\right) = \frac{\mu\left(X_1^{(i)}\right) g\left(y_1 | X_1^{(i)}\right)}{q\left(X_1^{(i)} \middle| y_1\right)}.$$

Sequential Monte Carlo for Hidden Markov Models

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set
  $w_1\left(X_1^{(i)}\right) = \frac{\mu\left(X_1^{(i)}\right) g\left(y_1 | X_1^{(i)}\right)}{q\left(X_1^{(i)} \middle| y_1\right)}$.
- Resample $\left\{X_1^{(i)}, W_1^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_1^{(i)}\right\}$

Sequential Monte Carlo for Hidden Markov Models

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set
  $w_1\left(X_1^{(i)}\right) = \frac{\mu\left(X_1^{(i)}\right) g\left(y_1 | X_1^{(i)}\right)}{q\left(X_1^{(i)} \middle| y_1\right)}$.
- Resample $\left\{X_1^{(i)}, W_1^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_1^{(i)}\right\}$
- At time $k \geq 2$

Sequential Monte Carlo for Hidden Markov Models

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set
  $$w_1\left(X_1^{(i)}\right) = \frac{\mu\left(X_1^{(i)}\right) g\left(y_1 | X_1^{(i)}\right)}{q\left(X_1^{(i)} \middle| y_1\right)}.$$
- Resample $\left\{X_1^{(i)}, W_1^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_1^{(i)}\right\}$
- At time $k \geq 2$
  - sample $X_k^{(i)} \sim q\left(\cdot | y_k, X_{k-1}^{(i)}\right)$

Sequential Monte Carlo for Hidden Markov Models

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set
  $$w_1\left(X_1^{(i)}\right) = \frac{\mu\left(X_1^{(i)}\right) g\left(y_1 | X_1^{(i)}\right)}{q\left(X_1^{(i)} \middle| y_1\right)}.$$
- Resample $\left\{X_1^{(i)}, W_1^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_1^{(i)}\right\}$
- At time $k \geq 2$
  - sample $X_k^{(i)} \sim q\left(\cdot | y_k, X_{k-1}^{(i)}\right)$
  - compute $w_k\left(X_{1:k}^{(i)}\right) = \frac{f\left(X_k^{(i)} \middle| X_{k-1}^{(i)}\right) g\left(y_k | X_k^{(i)}\right)}{q\left(X_k^{(i)} \middle| y_k, X_{k-1}^{(i)}\right)}.$

Sequential Monte Carlo for Hidden Markov Models

- At time $k = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set
  $w_1\left(X_1^{(i)}\right) = \frac{\mu\left(X_1^{(i)}\right) g\left(y_1 | X_1^{(i)}\right)}{q\left(X_1^{(i)} \middle| y_1\right)}$.
- Resample $\left\{X_1^{(i)}, W_1^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_1^{(i)}\right\}$
- At time $k \geq 2$
  - sample $X_k^{(i)} \sim q\left(\cdot | y_k, X_{k-1}^{(i)}\right)$
  - compute $w_k\left(X_{1:k}^{(i)}\right) = \frac{f\left(X_k^{(i)} \middle| X_{k-1}^{(i)}\right) g\left(y_k | X_k^{(i)}\right)}{q\left(X_k^{(i)} \middle| y_k, X_{k-1}^{(i)}\right)}$.
- Resample $\left\{X_{1:k}^{(i)}, W_k^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_{1:k}^{(i)}\right\}$

- **Example**: Linear Gaussian model

$$\begin{aligned} X_1 &\sim \mathcal{N}(0,1), \; X_n = \alpha X_{n-1} + \sigma_v V_n, \\ Y_n &= X_n + \sigma_w W_n \end{aligned}$$

where $V_n \sim \mathcal{N}(0,1)$ and $W_n \sim \mathcal{N}(0,1)$.

- **Example**: Linear Gaussian model

$$\begin{aligned} X_1 &\sim \mathcal{N}(0,1), \; X_n = \alpha X_{n-1} + \sigma_v V_n, \\ Y_n &= X_n + \sigma_w W_n \end{aligned}$$

  where $V_n \sim \mathcal{N}(0,1)$ and $W_n \sim \mathcal{N}(0,1)$.
- We know that $p(x_{1:n} | y_{1:n})$ is Gaussian and its parameters can be computed using Kalman techniques. In particular $p(x_n | y_{1:n})$ is also a Gaussian which can be computed using the Kalman filter.

- **Example**: Linear Gaussian model

$$\begin{aligned} X_1 &\sim \mathcal{N}(0,1), \; X_n = \alpha X_{n-1} + \sigma_v V_n, \\ Y_n &= X_n + \sigma_w W_n \end{aligned}$$

  where $V_n \sim \mathcal{N}(0,1)$ and $W_n \sim \mathcal{N}(0,1)$.
- We know that $p(x_{1:n} | y_{1:n})$ is Gaussian and its parameters can be computed using Kalman techniques. In particular $p(x_n | y_{1:n})$ is also a Gaussian which can be computed using the Kalman filter.
- We apply the SMC method with $q(x_k | y_k, x_{k-1}) = f(x_k | x_{k-1}) = \mathcal{N}(x_k; \alpha x_{n-1}, \sigma_v^2)$.

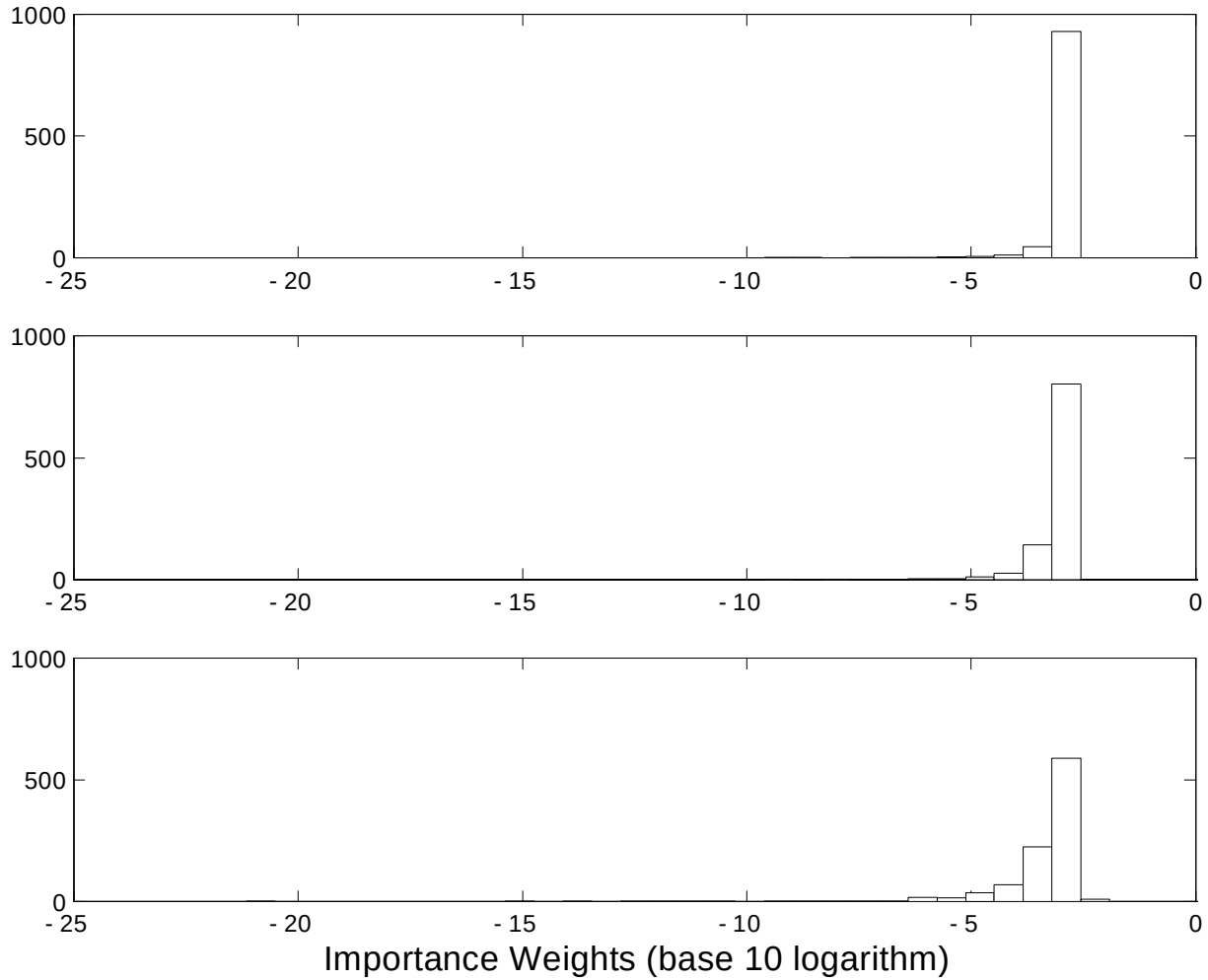

Figure: Histograms of the base 10 logarithm of $W_n^{(i)}$ for $n = 1$ (top), $n = 50$ (middle) and $n = 100$ (bottom).

- By itself this graph does not mean that the procedure is efficient!

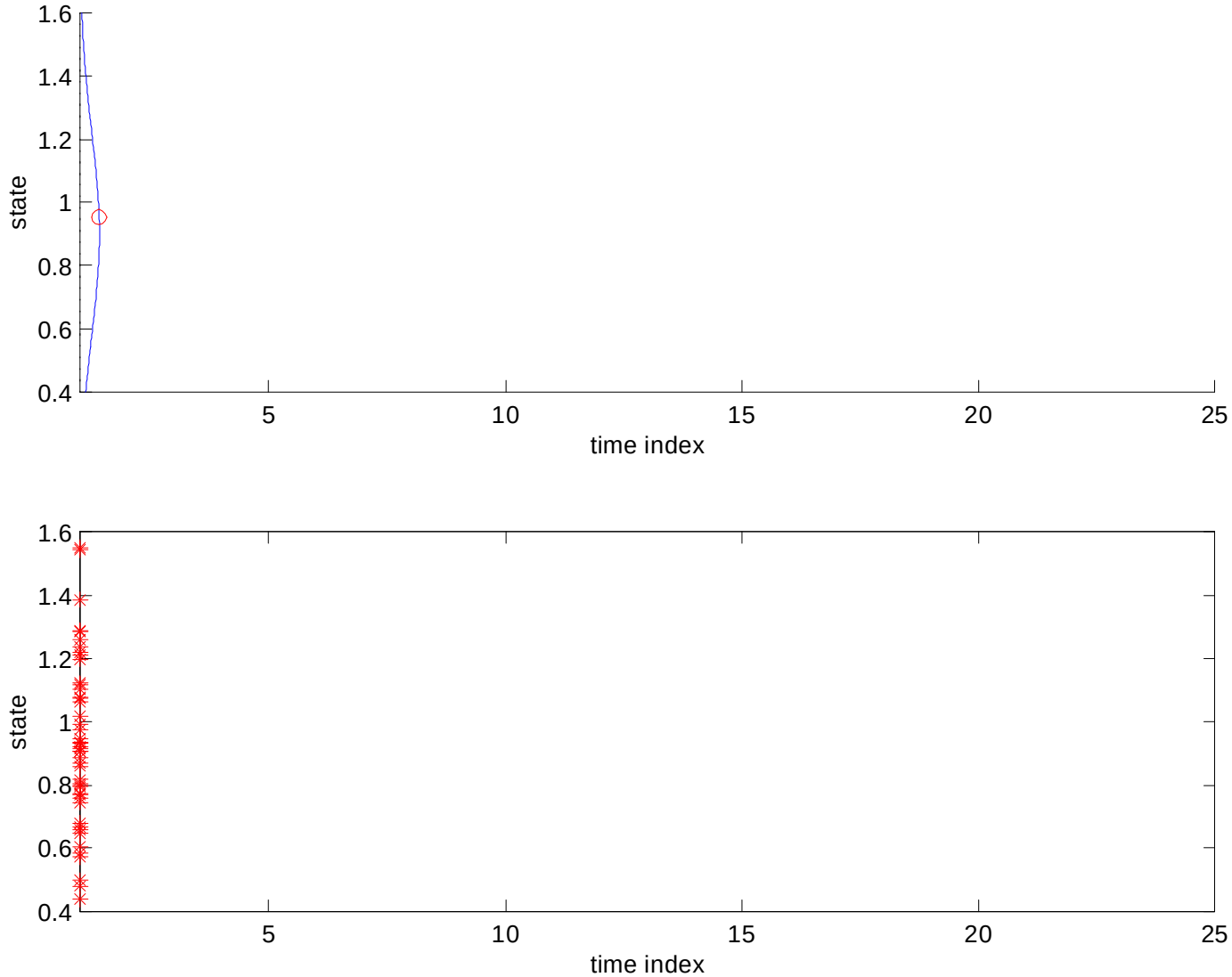

Figure: $p(x_1 | y_1)$ and $\widehat{\mathbb{E}}[X_1 | y_1]$ (top) and its particle approximation (bottom)

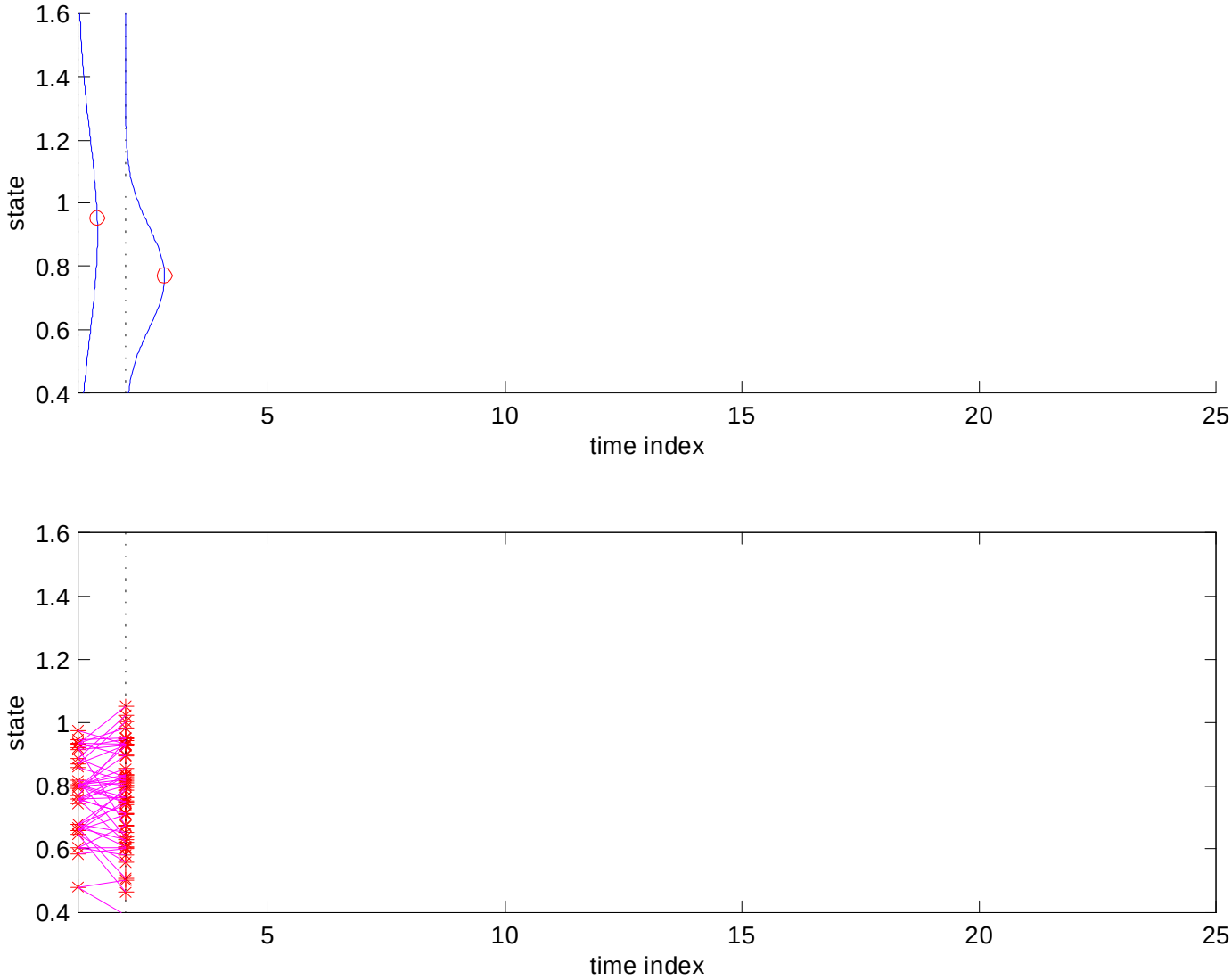

Figure: $p\left(x_1 \middle| y_1\right)$, $p\left(x_2 \middle| y_{1:2}\right)$and $\widehat{\mathbb{E}}\left[X_1 \middle| y_1\right]$, $\widehat{\mathbb{E}}\left[X_2 \middle| y_{1:2}\right]$ (top) and particle approximation of $p\left(x_{1:2} \middle| y_{1:2}\right)$ (bottom)

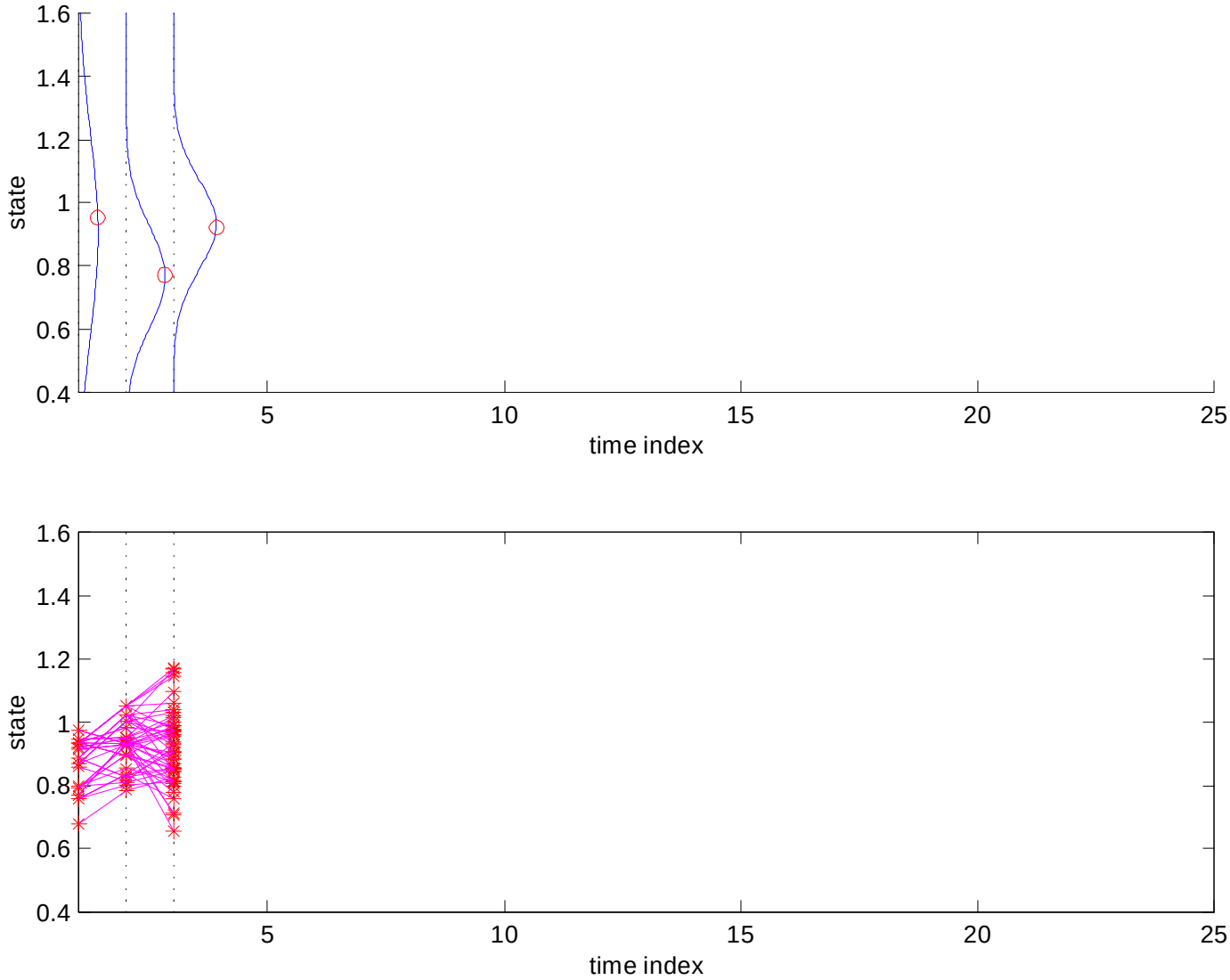

Figure: $p\left(x_k \middle| y_{1:k}\right)$ and $\widehat{\mathbb{E}}\left[X_k \middle| y_{1:k}\right]$ for $k = 1, 2, 3$ (top) and particle approximation of $p\left(x_{1:3} \middle| y_{1:3}\right)$ (bottom)

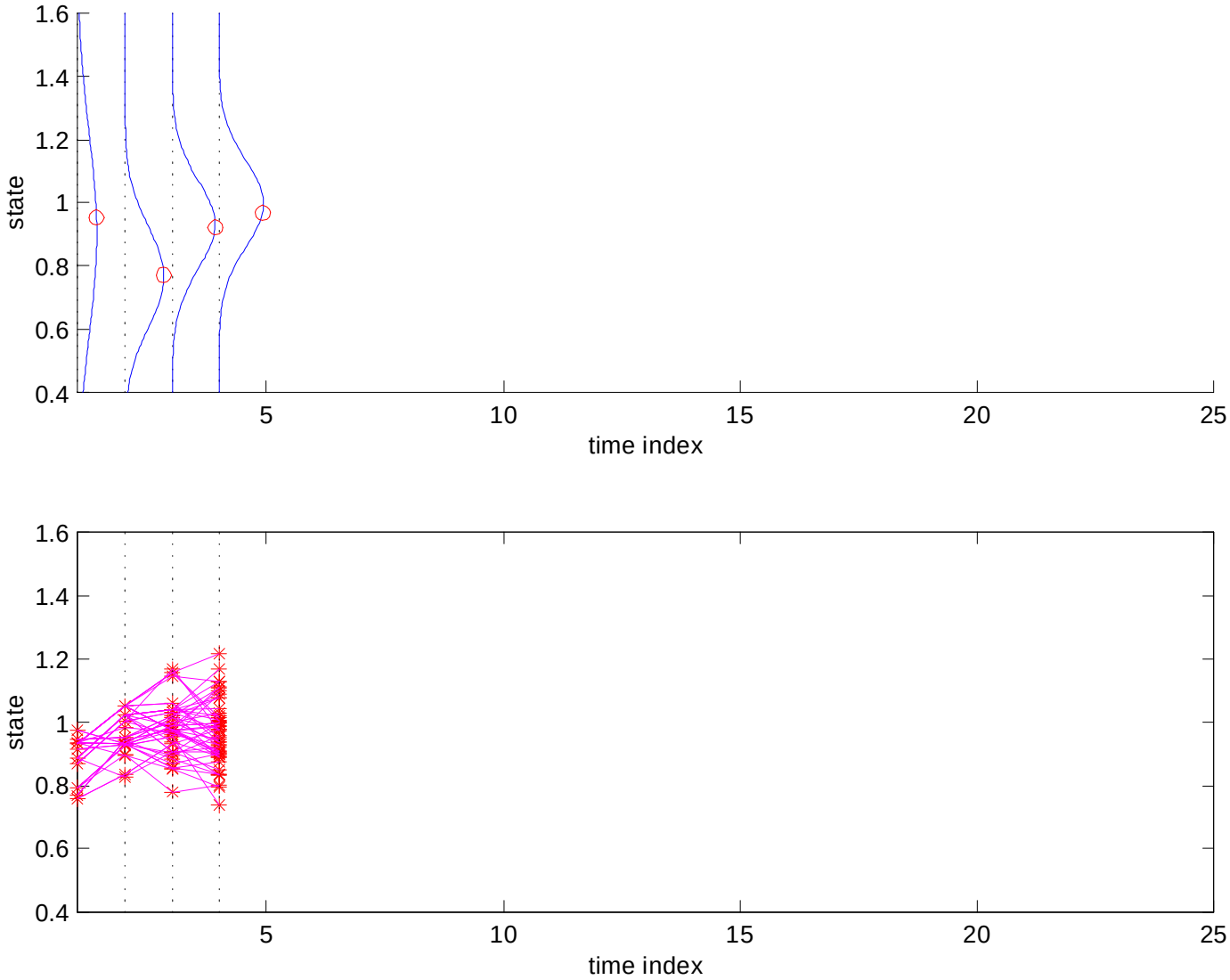

Figure: $p\left(x_k \middle| y_{1:k}\right)$ and $\widehat{\mathbb{E}}\left[X_k \middle| y_{1:k}\right]$ for $k = 1, .., 4$ (top) and particle approximation of $p\left(x_{1:4} \middle| y_{1:4}\right)$ (bottom)

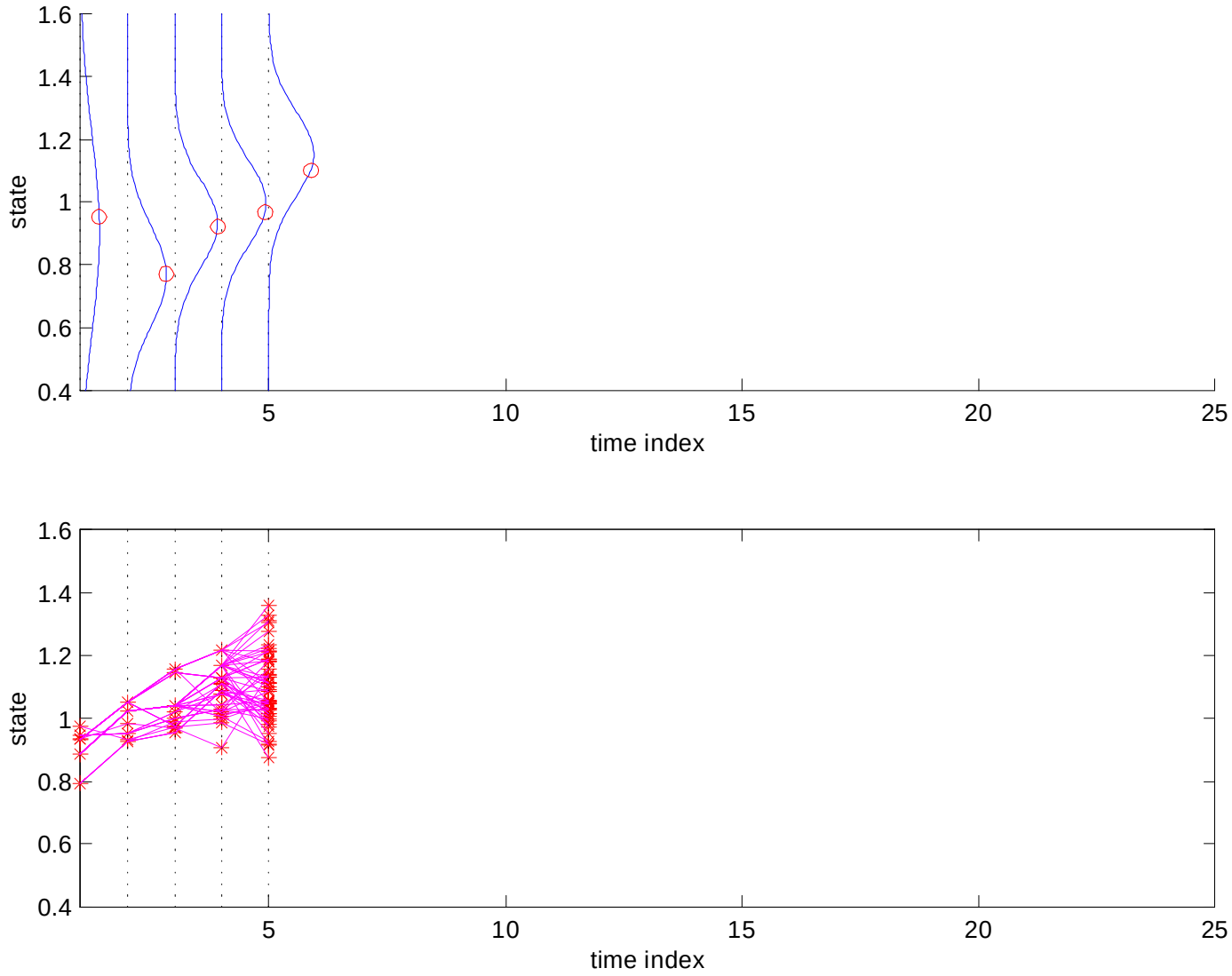

Figure: $p\left(x_k \middle| y_{1:k}\right)$ and $\widehat{\mathbb{E}}\left[X_k \middle| y_{1:k}\right]$ for $k = 1, \ldots, 5$ (top) and particle approximation of $p\left(x_{1:5} \middle| y_{1:5}\right)$ (bottom)

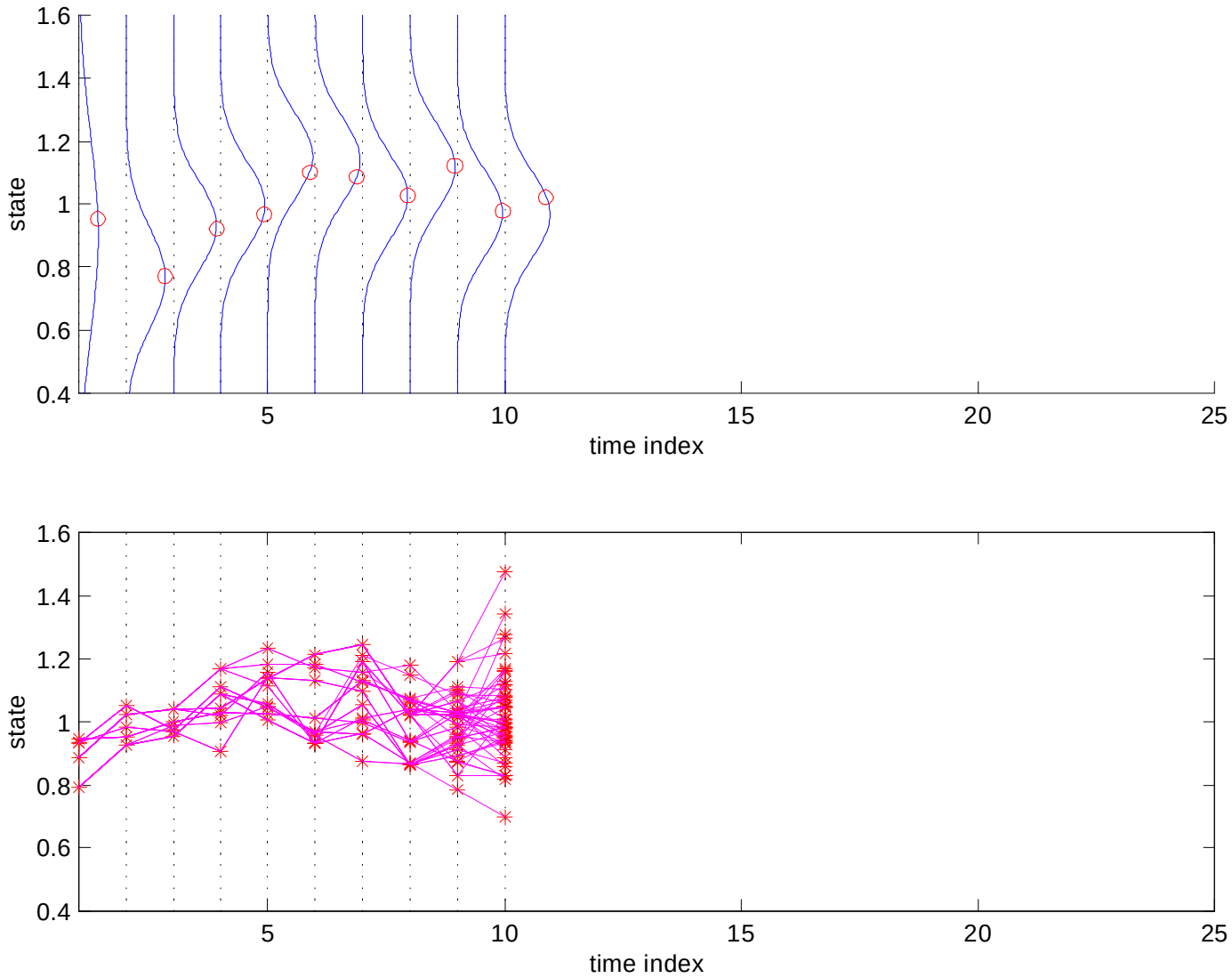

Figure: $p\left(x_k \middle| y_{1:k}\right)$ and $\widehat{\mathbb{E}}\left[X_k \middle| y_{1:k}\right]$ for $k = 1, ..., 10$ (top) and particle approximation of $p\left(x_{1:10} \middle| y_{1:10}\right)$ (bottom)

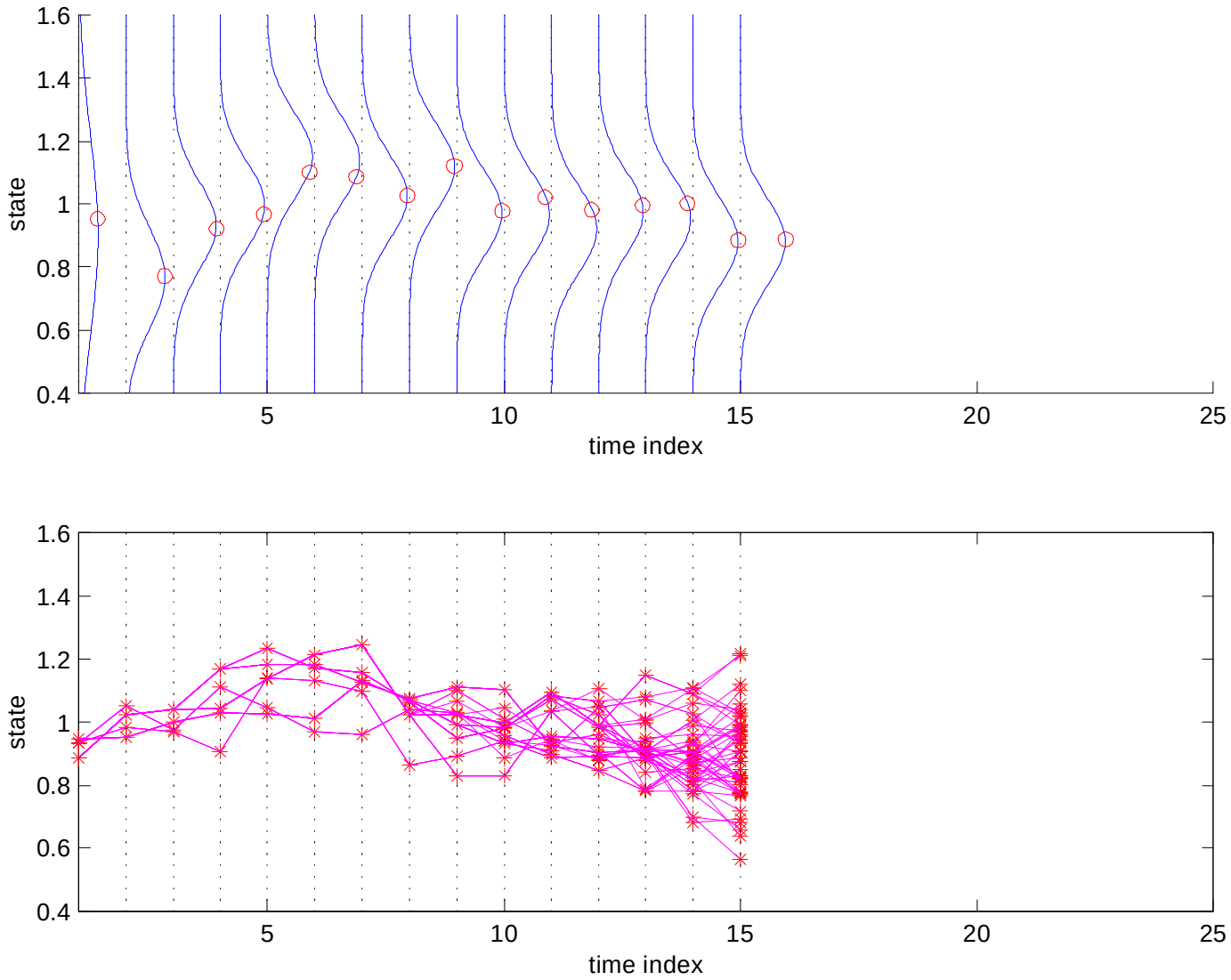

Figure: $p\left(x_k \middle| y_{1:k}\right)$ and $\widehat{\mathbb{E}}\left[X_k \middle| y_{1:k}\right]$ for $k = 1, ..., 15$ (top) and particle approximation of $p\left(x_{1:15} \middle| y_{1:15}\right)$ (bottom)

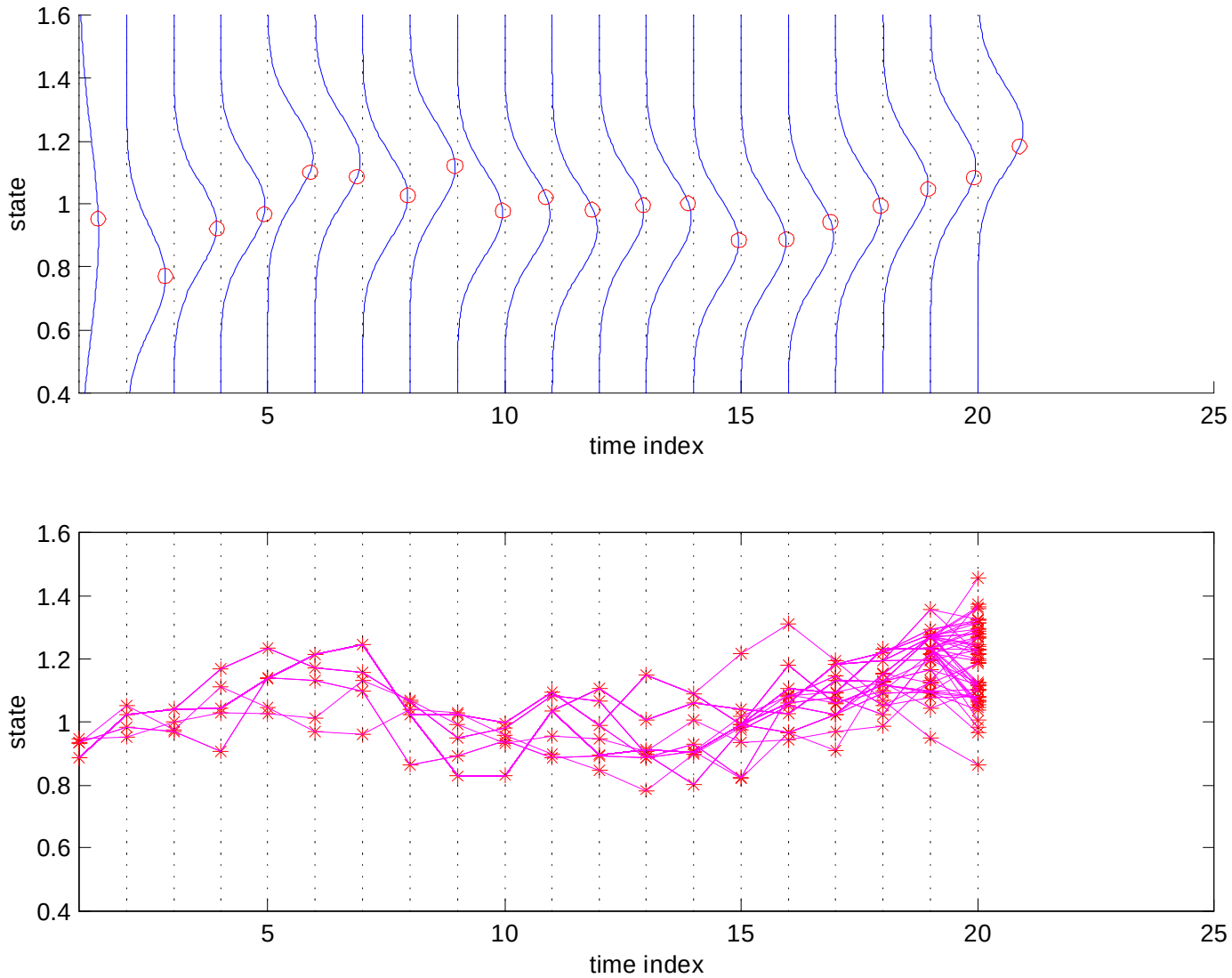

Figure: $p(x_k | y_{1:k})$ and $\widehat{\mathbb{E}}[X_k | y_{1:k}]$ for $k = 1, ..., 20$ (top) and particle approximation of $p(x_{1:20} | y_{1:20})$ (bottom)

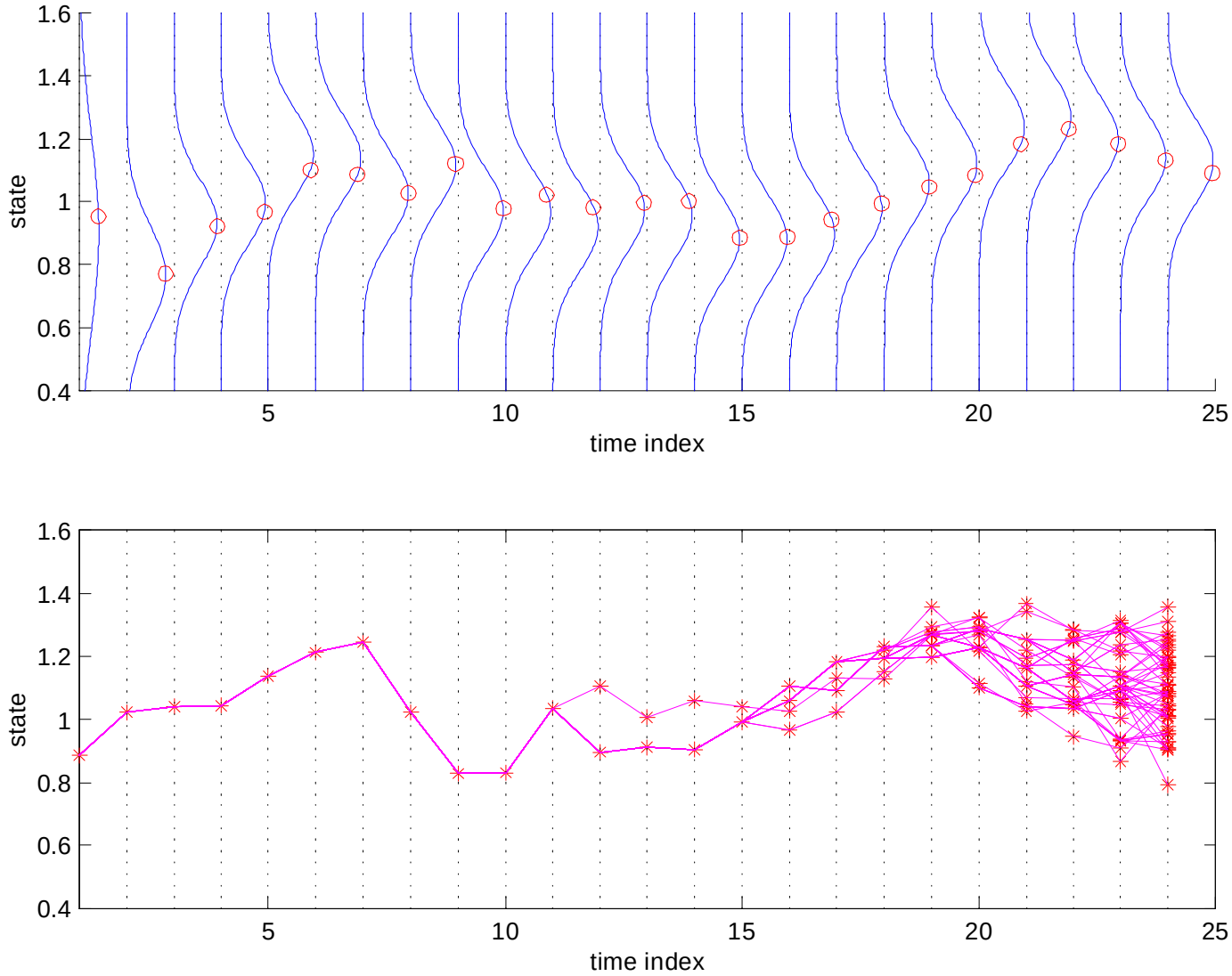

Figure: $p(x_k | y_{1:k})$ and $\widehat{\mathbb{E}}[X_k | y_{1:k}]$ for $k = 1, ..., 24$ (top) and particle approximation of $p(x_{1:24} | y_{1:24})$ (bottom)

- This SMC strategy performs remarkably well in terms of estimation of the marginals $p\left(x_k \middle| y_{1:k}\right)$. This is what is only necessary in many applications thankfully.

- This SMC strategy performs remarkably well in terms of estimation of the marginals $p\left(x_k \middle| y_{1:k}\right)$. This is what is only necessary in many applications thankfully.
- However, the joint distribution $p\left(x_{1:k} \middle| y_{1:k}\right)$ is poorly estimated when k is large; i.e. we have in the previous example

$$\widehat{p}\left(x_{1:11} \middle| y_{1:24}\right) = \delta_{X_{1:11}}\left(x_{1:11}\right).$$

- This SMC strategy performs remarkably well in terms of estimation of the marginals $p\left(x_k \middle| y_{1:k}\right)$. This is what is only necessary in many applications thankfully.
- However, the joint distribution $p\left(x_{1:k} \middle| y_{1:k}\right)$ is poorly estimated when k is large; i.e. we have in the previous example

$$\widehat{p}\left(x_{1:11} \middle| y_{1:24}\right) = \delta_{X_{1:11}}\left(x_{1:11}\right).$$

- The same conclusion holds for most sequences of distributions $\pi_k\left(x_{1:k}\right)$.

- This SMC strategy performs remarkably well in terms of estimation of the marginals $p\left(x_k \middle| y_{1:k}\right)$. This is what is only necessary in many applications thankfully.
- However, the joint distribution $p\left(x_{1:k} \middle| y_{1:k}\right)$ is poorly estimated when k is large; i.e. we have in the previous example

$$\widehat{p}\left(x_{1:11} \middle| y_{1:24}\right) = \delta_{X_{1:11}}\left(x_{1:11}\right).$$

- The same conclusion holds for most sequences of distributions $\pi_k\left(x_{1:k}\right)$.
- **Resampling only solves partially our problems**.

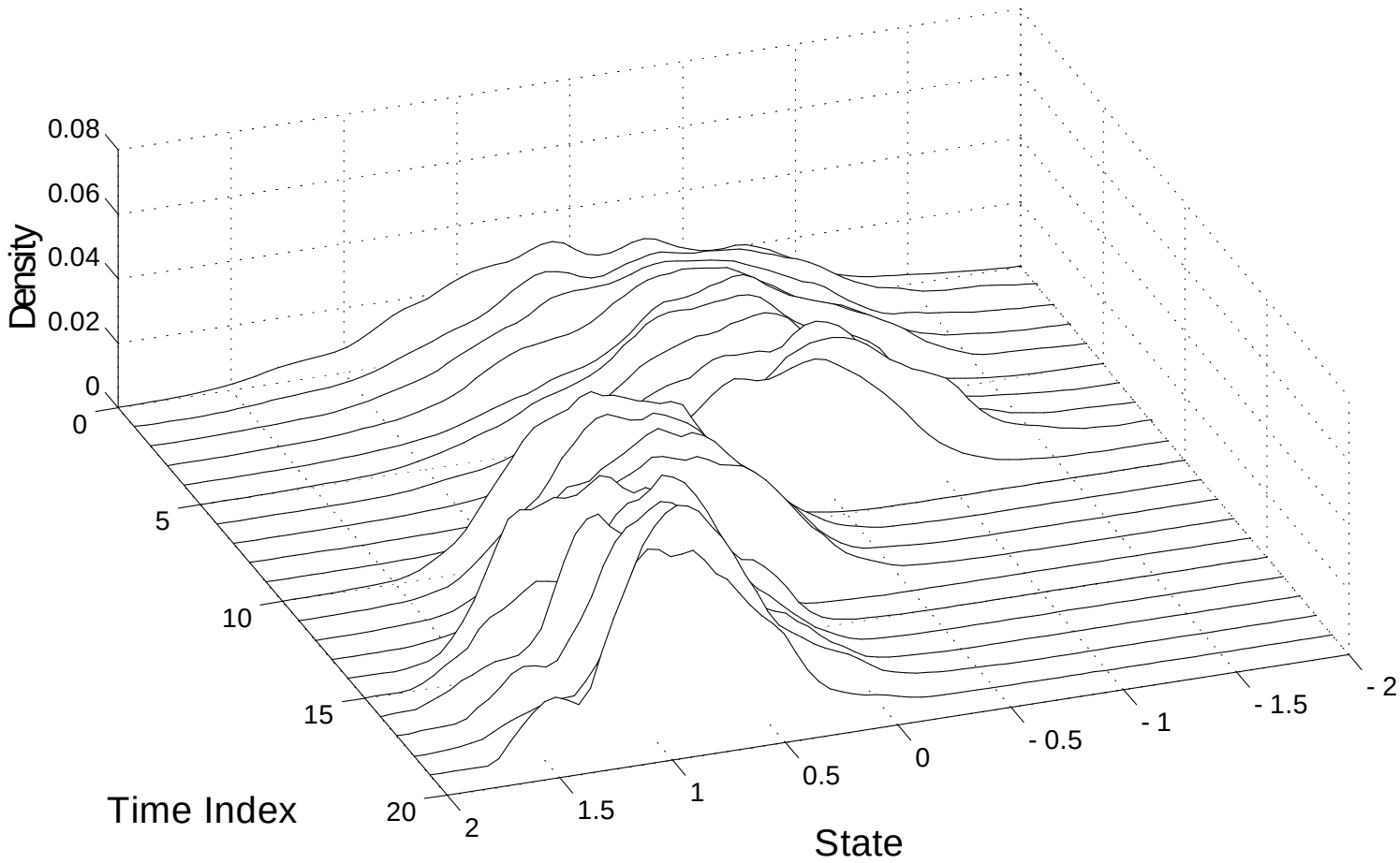

Figure: SMC estimates of the marginal distributions $p(x_n | y_{1:n})$.

Stochastic volatility model revisited using SMC.

Convergence of Sequential Monte Carlo

- Establishing convergence results for SMC is beyond the scope of this course but many results are available (e.g. Del Moral, 2004).

Convergence of Sequential Monte Carlo

- Establishing convergence results for SMC is beyond the scope of this course but many results are available (e.g. Del Moral, 2004).
- In particular we have for any bounded function φ and any $p > 1$

$$\mathbb{E}\left[\left|\int \varphi\left(x_{1:n}\right)\left(\widehat{\pi}_n\left(x_{1:n}\right) - \pi_n\left(x_{1:n}\right)\right) dx_{1:n}\right|^p\right]^{1/p} \leq \frac{C_n \left\|\varphi\right\|_\infty}{N}.$$

Convergence of Sequential Monte Carlo

- Establishing convergence results for SMC is beyond the scope of this course but many results are available (e.g. Del Moral, 2004).
- In particular we have for any bounded function φ and any $p > 1$

$$\mathbb{E}\left[\left|\int \varphi\left(x_{1:n}\right)\left(\widehat{\pi}_n\left(x_{1:n}\right) - \pi_n\left(x_{1:n}\right)\right) dx_{1:n}\right|^p\right]^{1/p} \leq \frac{C_n \left\|\varphi\right\|_\infty}{N}.$$

- It looks like a nice result... but it is rather useless as C_n increases polynomially/exponentially with time.

Convergence of Sequential Monte Carlo

- Establishing convergence results for SMC is beyond the scope of this course but many results are available (e.g. Del Moral, 2004).
- In particular we have for any bounded function φ and any $p > 1$

$$\mathbb{E}\left[\left|\int \varphi\left(x_{1:n}\right)\left(\widehat{\pi}_n\left(x_{1:n}\right) - \pi_n\left(x_{1:n}\right)\right) dx_{1:n}\right|^p\right]^{1/p} \leq \frac{C_n \left\|\varphi\right\|_\infty}{N}.$$

- It looks like a nice result... but it is rather useless as C_n increases polynomially/exponentially with time.
- To achieve a fixed precision, this would require to use a time-increasing number of particles N.

- One cannot hope to estimate with a fixed precision a target distribution of increasing dimension.

- One cannot hope to estimate with a fixed precision a target distribution of increasing dimension.
- So at best, we can expect results of the following form

$$\left[\left|\int \varphi\left(x_{n-L+1:n}\right)\left(\widehat{\pi}_n\left(x_{n-L+1:n}\right)-\pi_n\left(x_{n-L+1:n}\right)\right)dx_{n-L+1:n}\right|^p\right]^{1/p} \leq \frac{M_L\left\|\varphi\right\|_\infty}{N}$$

if the model has nice forgetting/mixing properties, i.e.

$$\int \left|\pi_n\left(x_n\middle|x_1\right)-\pi_n\left(x_n\middle|x_1'\right)\right|dx_n \leq \lambda^{n-1}$$

with $0 \leq \lambda < 1$.

- One cannot hope to estimate with a fixed precision a target distribution of increasing dimension.
- So at best, we can expect results of the following form

$$\left[\left|\int \varphi\left(x_{n-L+1:n}\right)\left(\widehat{\pi}_n\left(x_{n-L+1:n}\right)-\pi_n\left(x_{n-L+1:n}\right)\right)dx_{n-L+1:n}\right|^p\right]^{1/p} \leq \frac{M_L\left\|\varphi\right\|_\infty}{N}$$

  if the model has nice forgetting/mixing properties, i.e.

$$\int \left|\pi_n\left(x_n\middle|\, x_1\right)-\pi_n\left(x_n\middle|\, x_1'\right)\right| dx_n \leq \lambda^{n-1}$$

  with $0 \leq \lambda < 1$.
- In the HMM case, it means that

$$\int \left|p\left(x_n\middle|\, y_{1:n}, x_1\right)-p\left(x_n\middle|\, y_{1:n}, x_1'\right)\right| dx_n \leq \lambda^{n-1}$$

- We can have also a CLT. For the standard IS,

$$\sqrt{N}\left(\mathbb{E}_{\widehat{\pi}_n}\left(\varphi\left(X_n\right)\right)-\mathbb{E}_{\pi_n}\left(\varphi\left(X_n\right)\right)\right) \Rightarrow \mathcal{N}\left(0, \sigma^2_{IS,n}\left(\varphi\right)\right)$$
$$\text{where } \sigma^2_{IS,n}\left(\varphi\right) = \int \frac{\pi_n^2(x_{1:n})}{q_n(x_{1:n})}\left(\varphi\left(x_n\right)-\mathbb{E}_{\pi_n}\left(\varphi\left(X_n\right)\right)\right)^2 dx_{1:n}.$$

- We can have also a CLT. For the standard IS,

$$\sqrt{N}\left(\mathbb{E}_{\widehat{\pi}_n}\left(\varphi\left(X_n\right)\right)-\mathbb{E}_{\pi_n}\left(\varphi\left(X_n\right)\right)\right)\Rightarrow\mathcal{N}\left(0,\sigma_{IS,n}^2\left(\varphi\right)\right)$$
$$\text{where } \sigma_{IS,n}^2\left(\varphi\right)=\int\frac{\pi_n^2\left(x_{1:n}\right)}{q_n\left(x_{1:n}\right)}\left(\varphi\left(x_n\right)-\mathbb{E}_{\pi_n}\left(\varphi\left(X_n\right)\right)\right)^2dx_{1:n}.$$

- For SMC, we have

$$\sqrt{N}\left(\int\varphi\left(x_n\right)\left(\widehat{\pi}_n\left(x_n\right)-\pi_n\left(x_n\right)\right)dx_n\right)\Rightarrow\mathcal{N}\left(0,\sigma_{SMC,n}^2\left(\varphi\right)\right)$$
$$\text{where } \sigma_{SMC,n}^2\left(\varphi\right)=\int\frac{\pi_n^2\left(x_1\right)}{q_1\left(x_1\right)}\left(\int\varphi\left(x_n\right)\pi_n\left(x_n\middle|x_1\right)dx_n-\mathbb{E}_{\pi_n}\left(\varphi\left(X_n\right)\right)\right)^2dx_1$$
$$+\sum_{k=2}^{n-1}\int\frac{\pi_n\left(x_{k-1},x_k\right)^2}{\pi_{k-1}\left(x_{k-1}\right)q_k\left(x_k\middle|x_{k-1}\right)}\left(\int\varphi\left(x_n\right)\pi_n\left(x_n\middle|x_k\right)dx_n-\mathbb{E}_{\pi_n}\left(\varphi\left(X_n\right)\right)\right)^2dx_k$$
$$+\int\frac{\pi_n\left(x_{n-1},x_n\right)^2}{\pi_{n-1}\left(x_{n-1}\right)q_n\left(x_n\middle|x_{n-1}\right)}\left(\varphi\left(x_n\right)-\mathbb{E}_{\pi_n}\left(\varphi\left(X_n\right)\right)\right)^2dx_{n-1:n}.$$

- These results also demonstrate that one cannot expect to obtain good performance if the model has static parameters, i.e. if we have

$$X_1 \sim \mu, \;\; X_k \left| \left(X_{k-1} = x_{k-1} \right) \right. \sim f_\theta \left(\cdot \left| \, x_{k-1} \right. \right),$$

$$Y_k \left| \left(X_k = x_k \right) \right. \sim g_\theta \left(\cdot \left| \, x_k \right. \right).$$

where $\theta \sim \pi(\theta)$ and we want to estimate $p\left(x_{1:n}, \theta \left| \, y_{1:n} \right. \right)$.

- These results also demonstrate that one cannot expect to obtain good performance if the model has static parameters, i.e. if we have

$$X_1 \sim \mu, \;\; X_k \left| \left(X_{k-1} = x_{k-1} \right) \right. \sim f_\theta \left(\cdot \left| \, x_{k-1} \right. \right),$$

$$Y_k \left| \left(X_k = x_k \right) \right. \sim g_\theta \left(\cdot \left| \, x_k \right. \right).$$

  where $\theta \sim \pi(\theta)$ and we want to estimate $p\left(x_{1:n}, \theta \left| \, y_{1:n} \right. \right)$.
- Indeed the dynamic model $Z_n = (X_n, \theta)$ is not ergodic as

$$f\left(x', \theta' \left| \, x, \theta \right. \right) = \delta_\theta\left(\theta' \right) f_\theta\left(x' \left| \, x \right. \right).$$

- These results also demonstrate that one cannot expect to obtain good performance if the model has static parameters, i.e. if we have

$$X_1 \sim \mu, \;\; X_k \big| \left(X_{k-1} = x_{k-1}\right) \sim f_\theta\left(\cdot \big| x_{k-1}\right),$$

$$Y_k \big| \left(X_k = x_k\right) \sim g_\theta\left(\cdot \big| x_k\right).$$

  where $\theta \sim \pi(\theta)$ and we want to estimate $p\left(x_{1:n}, \theta \big| y_{1:n}\right)$.
- Indeed the dynamic model $Z_n = (X_n, \theta)$ is not ergodic as

$$f\left(x', \theta' \big| x, \theta\right) = \delta_\theta\left(\theta'\right) f_\theta\left(x' \big| x\right).$$

- This is intuitive! At time 1, we sample N particles $\theta^{(i)}$ and these values are never ever modified later on.

- At first glance, this is really bad news. SMC appears unable to deal with static parameters

- At first glance, this is really bad news. SMC appears unable to deal with static parameters
- A dirty solution consists of adding noise to a fixed parameter to transform it as a time-varying parameter

$$\theta_n = \theta_{n-1} + \varepsilon_n.$$

- At first glance, this is really bad news. SMC appears unable to deal with static parameters
- A dirty solution consists of adding noise to a fixed parameter to transform it as a time-varying parameter

$$\theta_n = \theta_{n-1} + \varepsilon_n.$$

- This is not clean and we will discuss later on a rigorous approach which requires a "deeper" understanding of SMC.