Advanced Computational Methods in Statistics
Lecture 4

O. Deniz Akyildiz

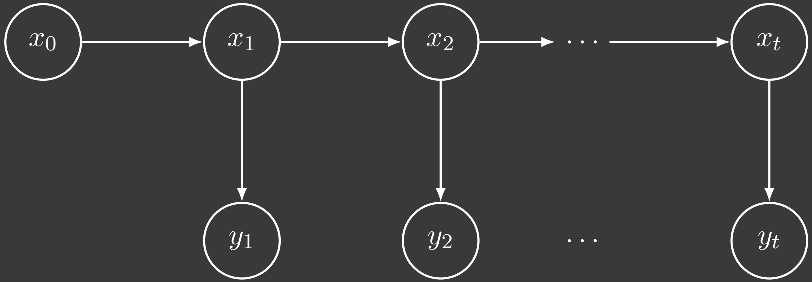
LTCC Advanced Course

December 4, 2023

Imperial College London

State-space models

problem definition



The conditional independence structure of a state-space model.

$(x_t)_{t\in\mathbb{N}_+}$: *hidden* signal process, $(y_t)_{t\in\mathbb{N}_+}$ the observation process.

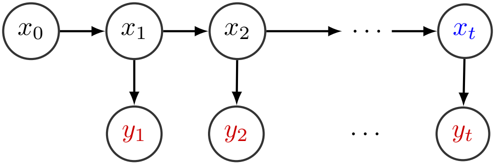
$$x_0 \sim \pi_0(\mathsf{d}x_0), \quad \text{(prior distribution)}$$
$$x_t|x_{t-1} \sim \tau_t(\mathrm{d}x_t|x_{t-1}), \quad \text{(transition model)}$$
$$y_t|x_t \sim g_t(y_t|x_t), \quad \text{(likelihood)}$$

$x_t \in \mathsf{X}$ where X is the state-space. We use: $g_t(x_t) = g_t(y_t|x_t)$.

We are interested in estimating expectations,

$$(\varphi, \pi_t) = \int \varphi(x_t)\pi_t(x_t|y_{1:t})\mathrm{d}x_t = \int \varphi(x_t)\pi_t(\mathrm{d}x_t),$$

sequentially as new data arrives.



Algorithm:

Predict

$$\xi_t(\mathrm{d}x_t) = \int \pi_{t-1}(\mathrm{d}x_{t-1})\tau_t(\mathrm{d}x_t|x_{t-1})$$

Update

$$\pi_t(\mathrm{d}x_t) = \xi_t(\mathrm{d}x_t)\frac{g_t(y_t|x_t)}{p(y_t|y_{1:t-1})}.$$

Before we go into the details of the derivation, let us directly look at the algorithm.

Before we go into the details of the derivation, let us directly look at the algorithm. A general algorithm to estimate expectations of any test function $\varphi(x_t)$ given $y_{1:t}$.

- Sampling: draw
$$\bar{x}_t^{(i)} \sim \tau_t(\mathrm{d}x_t | x_{t-1}^{(i)})$$
independently for every $i = 1, \ldots, N$.
- Weighting: compute
$$w_t^{(i)} = g_t(\bar{x}_t^{(i)}) / \bar{Z}_t^N$$
for every $i = 1, \ldots, N$, where $\bar{Z}_t^N = \sum_{i=1}^N g_t(\bar{x}_t^{(i)})$.
- Resampling: draw independently,
$$x_t^{(i)} \sim \tilde{\pi}_t(\mathrm{d}x) := \sum_i w_t^{(i)} \delta_{\bar{x}_t^{(i)}}(\mathrm{d}x) \quad \text{for } i = 1, ..., N.$$

$$\pi_{t-1}^N \underbrace{\rightarrow}_{\text{sampling}} \xi_t^N \underbrace{\rightarrow}_{\text{weighting}} \tilde{\pi}_t^N \underbrace{\rightarrow}_{\text{resampling}} \pi_t^N.$$

The key recursion on the path distributions.

$$\begin{aligned}
\pi_t(x_{0:t}|y_{1:t}) &= \frac{\gamma(x_{0:t}, y_{1:t})}{p(y_{1:t})} \\
&= \frac{\gamma(x_{0:t-1}, y_{1:t-1})}{p(y_{1:t-1})} \frac{\tau(x_t|x_{t-1})g(y_t|x_t)}{p(y_t|y_{1:t-1})} \\
&= \pi_t(x_{0:t-1}|y_{1:t-1}) \frac{\tau(x_t|x_{t-1})g(y_t|x_t)}{p(y_t|y_{1:t-1})}.
\end{aligned}$$

Recall importance sampling: Assume that we aim at estimating expectations of a given density π, i.e., we would like to compute

$$(\varphi, \pi) = \int \varphi(x)\pi(x)\mathrm{d}x.$$

We also assume that sampling from this density is not possible and we can only evaluate the *unnormalised* density $\gamma(x)$.

One way to estimate this expectation is to sample from a proposal measure q and rewrite the integral as

$$\begin{aligned}(\varphi, \pi) &= \int \varphi(x)\pi(x)\mathrm{d}x, \\ &= \frac{\int \varphi(x)\frac{\gamma(x)}{q(x)}q(x)\mathrm{d}x}{\int \frac{\gamma(x)}{q(x)}q(x)\mathrm{d}x}, \\ &\approx \frac{\frac{1}{N}\sum_{i=1}^{N} \varphi(x^{(i)})\frac{\gamma(x^{(i)})}{q(x^{(i)})}}{\frac{1}{N}\sum_{i=1}^{N} \frac{\gamma(x^{(i)})}{q(x^{(i)})}}, \qquad x^{(i)} \sim q, \quad i = 1, \dots, N.\end{aligned} \tag{1}$$

Let us now introduce the unnormalised weight function

$$W(x) = \frac{\gamma(x)}{q(x)}. \tag{2}$$

With this, the Eq. (1) becomes

$$\begin{aligned}(\varphi, \pi^N) &= \frac{\frac{1}{N}\sum_{i=1}^N \varphi(x^{(i)}) W(x^{(i)})}{\frac{1}{N}\sum_{i=1}^N W(x^{(i)})}, \qquad x^{(i)} \sim q, \quad i = 1, \dots, N, \\ &= \frac{\sum_{i=1}^N \varphi(x^{(i)}) \mathsf{W}^{(i)}}{\sum_{i=1}^N \mathsf{W}^{(i)}}, \qquad x^{(i)} \sim q, \quad i = 1, \dots, N,\end{aligned}$$

where $\mathsf{W}^{(i)} = W(x^{(i)})$ are called *the unnormalised weights*.

Finally, we can obtain the estimator in a more convenient form,

$$(\varphi, \pi^N) = \sum_{i=1}^{N} \mathsf{w}^{(i)} \varphi(x^{(i)}).$$

by introducing the *normalised importance weights*

$$\mathsf{w}^{(i)} = \frac{\mathsf{W}^{(i)}}{\sum_{i=1}^{N} \mathsf{W}^{(i)}}, \tag{3}$$

for $i = 1, \ldots, N$. We note that the particle approximation of π in this case is given as

$$\pi^N(\mathrm{d}x) = \sum_{i=1}^{N} \mathsf{w}^{(i)} \delta_{x^{(i)}}(\mathrm{d}x). \tag{4}$$

In the following, we will derive the importance sampler aiming at building particle approximations of $\pi_t(x_{0:t}|y_{1:t})$ for a state-space model.

The proposal over the entire path space $x_{0:t}$ denoted $q(x_{0:t})$. Note

$$\gamma(x_{0:t}, y_{1:t}) = \mu(x_0) \prod_{k=1}^{t} \tau(x_k|x_{k-1}) g(y_k|x_k). \tag{5}$$

The proposal over the entire path space $x_{0:t}$ denoted $q(x_{0:t})$. Note

$$\gamma(x_{0:t}, y_{1:t}) = \mu(x_0) \prod_{k=1}^{t} \tau(x_k|x_{k-1}) g(y_k|x_k). \tag{5}$$

This simply the joint distribution of all variables $(x_{0:t}, y_{1:t})$. Just as in the regular importance sampling

$$W_{0:t}(x_{0:t}) = \frac{\gamma(x_{0:t}, y_{1:t})}{q(x_{0:t})}.$$

Obviously, given samples from the proposal $x_{0:t}^{(i)} \sim q(x_{0:t})$, by evaluating the weight $\mathsf{W}_{0:t}^{(i)} = W_{0:t}(x_{0:t}^{(i)})$ for $i = 1, \ldots, N$ and building a particle approximation

$$\pi^N(\mathrm{d}x_{0:t}) = \sum_{i=1}^{N} \mathsf{w}_{0:t}^{(i)} \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t}).$$

where

$$\mathsf{w}_{0:t}^{(i)} = \frac{\mathsf{W}_{0:t}^{(i)}}{\sum_{i=1}^{N} \mathsf{W}_{0:t}^{(i)}}.$$

The path space importance sampler

- Sample $x_{0:T}^{(i)} \sim q(x_{0:T})$ for $i = 1, \ldots, N$.

The path space importance sampler

- Sample $x_{0:T}^{(i)} \sim q(x_{0:T})$ for $i = 1, \ldots, N$.
- Compute weights:

$$\mathsf{W}_{0:t}^{(i)} = \frac{\gamma(x_{0:t}, y_{1:t})}{q(x_{0:t})}.$$

  and normalise

$$\mathsf{w}_{0:t}^{(i)} = \frac{\mathsf{W}_{0:t}^{(i)}}{\sum_{i=1}^N \mathsf{W}_{0:t}^{(i)}}.$$

- Report

$$\pi_t^N(\mathrm{d}x_{0:t}) = \sum_{i=1}^N \mathsf{w}_{0:t}^{(i)} \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t}).$$

Let us consider a decomposition of the proposal

$$q(x_{0:t}) = q(x_0) \prod_{k=1}^{t} q(x_k | x_{1:k-1}).$$

Note that, based on this, we can build a recursion for the function $W(x_{0:t})$ by writing

$$\begin{aligned} W_{0:t}(x_{0:t}) &= \frac{\gamma(x_{0:t}, y_{1:t})}{q(x_{0:t})}, \\ &= \frac{\gamma(x_{0:t-1}, y_{1:t-1})}{q(x_{0:t-1})} \frac{\tau(x_t | x_{t-1}) g(y_t | x_t)}{q(x_t | x_{0:t-1})}, \\ &= W_{0:t-1}(x_{0:t-1}) \frac{\tau(x_t | x_{t-1}) g(y_t | x_t)}{q(x_t | x_{0:t-1})}, \\ &= W_{0:t-1}(x_{0:t-1}) W_t(x_{0:t}). \qquad (6) \end{aligned}$$

This is still not optimal, as we still need to store the whole path.

This is still not optimal, as we still need to store the whole path.

We can further simplify our proposal by assuming a Markov structure.

$$q(x_{0:t}) = q(x_0) \prod_{k=1}^{t} q(x_k|x_{k-1}).$$

This allows us to obtain purely recursive weight computation

$$\begin{aligned} W_{0:t}(x_{0:t}) &= \frac{\gamma(x_{0:t}, y_{1:t})}{q(x_{0:t})}, & (7) \\ &= \frac{\gamma(x_{0:t-1}, y_{1:t-1})}{q(x_{0:t-1})} \frac{\tau(x_t|x_{t-1})g(y_t|x_t)}{q(x_t|x_{t-1})}, & (8) \\ &= W_{0:t-1}(x_{0:t-1}) \frac{\tau(x_t|x_{t-1})g(y_t|x_t)}{q(x_t|x_{t-1})}, & (9) \\ &= W_{0:t-1}(x_{0:t-1}) W_t(x_t, x_{t-1}), & (10) \end{aligned}$$

- Assume that we have computed the unnormalised weights $\mathsf{W}_{0:t-1}^{(i)} = W(x_{0:t-1}^{(i)})$ recursively and obtained samples $x_{0:t-1}^{(i)}$.

- Assume that we have computed the unnormalised weights $\mathsf{W}_{0:t-1}^{(i)} = W(x_{0:t-1}^{(i)})$ recursively and obtained samples $x_{0:t-1}^{(i)}$.
- We only need the last sample $x_{t-1}^{(i)}$ to obtain the weight update given in (10).

- Assume that we have computed the unnormalised weights $\mathsf{W}_{0:t-1}^{(i)} = W(x_{0:t-1}^{(i)})$ recursively and obtained samples $x_{0:t-1}^{(i)}$.
- We only need the last sample $x_{t-1}^{(i)}$ to obtain the weight update given in (10).
- And also note that $\mathsf{W}_{0:t-1}^{(i)}$ for $i = 1, \ldots, N$ are just numbers, they do not require the storage of previous samples.

We can now sample from the Markov proposal $x_t^{(i)} \sim q(x_t|x_{t-1}^{(i)})$ and compute the weights of the path sampler at time t as

$$\mathsf{W}_{1:t}^{(i)} = \mathsf{W}_{1:t-1}^{(i)} \times \mathsf{W}_t^{(i)},$$

where

$$\mathsf{W}_t^{(i)} = \frac{\tau(x_t^{(i)}|x_{t-1}^{(i)})g(y_t|x_t^{(i)})}{q(x_t^{(i)}|x_{t-1}^{(i)})}.$$

Given the samples $x_{t-1}^{(i)}$, we first perform sampling step

$$x_t^{(i)} \sim q(x_t|x_{t-1})$$

Given the samples $x_{t-1}^{(i)}$, we first perform sampling step

$$x_t^{(i)} \sim q(x_t|x_{t-1})$$

and then compute

$$\mathsf{W}_t^{(i)} = \frac{\tau(x_t^{(i)}|x_{t-1}^{(i)})g(y_t|x_t^{(i)})}{q(x_t^{(i)}|x_{t-1}^{(i)})}.$$

and update

$$\mathsf{W}_{0:t}^{(i)} = \mathsf{W}_{0:t-1}^{(i)} \times \mathsf{W}_t^{(i)}.$$

Given the samples $x_{t-1}^{(i)}$, we first perform sampling step

$$x_t^{(i)} \sim q(x_t | x_{t-1})$$

and then compute

$$\mathsf{W}_t^{(i)} = \frac{\tau(x_t^{(i)} | x_{t-1}^{(i)}) g(y_t | x_t^{(i)})}{q(x_t^{(i)} | x_{t-1}^{(i)})}.$$

and update

$$\mathsf{W}_{0:t}^{(i)} = \mathsf{W}_{0:t-1}^{(i)} \times \mathsf{W}_t^{(i)}.$$

These are unnormalised weights and we normalise them to obtain,

$$\mathsf{w}_{0:t}^{(i)} = \frac{\mathsf{W}_{0:t}^{(i)}}{\sum_{i=1}^N \mathsf{W}_{0:t}^{(i)}},$$

which finally leads to the empirical measure,

$$\pi^N(\mathrm{d}x_{0:t}) = \sum_{i=1}^{N} \mathsf{w}_{0:t}^{(i)} \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t}).$$

- Sample $x_0^{(i)} \sim q(x_0)$ for $i = 1, \ldots, N$.
- For $t \geq 1$
  - Sample: $x_t^{(i)} \sim q(x_t | x_{t-1}^{(i)})$,
  - Compute weights:

$$\mathsf{W}_t^{(i)} = \frac{\tau(x_t^{(i)} | x_{t-1}^{(i)}) g(y_t | x_t^{(i)})}{q(x_t^{(i)} | x_{t-1}^{(i)})}.$$

    and update

$$\mathsf{W}_{0:t}^{(i)} = \mathsf{W}_{0:t-1}^{(i)} \times \mathsf{W}_t^{(i)}.$$

    Normalise weights,

$$\mathsf{w}_{0:t}^{(i)} = \frac{\mathsf{W}_{0:t}^{(i)}}{\sum_{i=1}^N \mathsf{W}_{0:t}^{(i)}}.$$

  - Report

$$\pi_t^N(\mathrm{d}x_{0:t}) = \sum_{i=1}^N \mathsf{w}_{0:t}^{(i)} \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t}).$$

There is a well-known problem with this scheme: *Weight degeneracy*.

There is a well-known problem with this scheme: *Weight degeneracy*.

Quiz: What is the problem with the weights?

There is a well-known problem with this scheme: *Weight degeneracy*.

Quiz: What is the problem with the weights?

Addition in the log-domain will cause big discrepancies between log-weights, which will result in degeneracy after normalisation.

There is a well-known problem with this scheme: *Weight degeneracy*.

Quiz: What is the problem with the weights?

Addition in the log-domain will cause big discrepancies between log-weights, which will result in degeneracy after normalisation.

To resolve this, the approach is to introduce resampling steps.

- Sample $x_0^{(i)} \sim q(x_0)$ for $i = 1, \ldots, N$.
- For $t \geq 1$
  - Sample: $\bar{x}_t^{(i)} \sim q(x_t | x_{t-1}^{(i)})$,
  - Compute weights:

$$\mathsf{W}_t^{(i)} = \frac{\tau(\bar{x}_t^{(i)} | x_{t-1}^{(i)}) g(y_t | \bar{x}_t^{(i)})}{q(\bar{x}_t^{(i)} | x_{t-1}^{(i)})}.$$

  Normalise: $\mathsf{w}_t^{(i)} = \mathsf{W}_t^{(i)} / \sum_{i=1}^N \mathsf{W}_t^{(i)}$
  - Report

$$\pi_t^N(\mathrm{d}x_t) = \sum_{i=1}^N \mathsf{w}_t^{(i)} \delta_{\bar{x}_t^{(i)}}(\mathrm{d}x_t).$$

  - Resample:

$$x_t^{(i)} \sim \sum_{i=1}^N \mathsf{w}_t^{(i)} \delta_{\bar{x}_t^{(i)}}(\mathrm{d}x_t).$$

SISR (and variants) also approximates the path distributions.

SISR (and variants) also approximates the path distributions.

But... When we resample the last particle, we essentially perform resampling on the path.

SISR (and variants) also approximates the path distributions.

But... When we resample the last particle, we essentially perform resampling on the path.

Say at time t, we resample

$$x_t^{(i)} = \bar{x}_t^{a_t^{(i)}},$$

in essence, we are also resampling past paths, so on the path space

$$x_{0:t}^{(i)} = (\bar{x_t}^{a_t^{(i)}}, x_{0:t-1}^{a_t^{(i)}}).$$

where $a_t^{(i)}$ are sampled from a discrete distribution with probabilities $\mathsf{w}_t^{(i)}$.

The bootstrap particle filter (BPF) is the SISR algorithm with the following choices:

$$q(x_t|x_{t-1}) = \tau(x_t|x_{t-1}),$$

- Sample $x_0^{(i)} \sim q(x_0)$ for $i = 1, \ldots, N$.
- For $t \geq 1$
  - Sample: $\bar{x}_t^{(i)} \sim \tau(x_t | x_{t-1}^{(i)})$,
  - Compute weights:

$$\mathsf{W}_t^{(i)} = g(y_t | \bar{x}_t^{(i)}),$$

    Normalise: $\mathsf{w}_t^{(i)} = \mathsf{W}_t^{(i)} / \sum_{i=1}^N \mathsf{W}_t^{(i)}$
  - Report

$$\pi_t^N(\mathrm{d}x_t) = \sum_{i=1}^N \mathsf{w}_t^{(i)} \delta_{\bar{x}_t^{(i)}}(\mathrm{d}x_t).$$

  - Resample:

$$x_t^{(i)} \sim \sum_{i=1}^N \mathsf{w}_t^{(i)} \delta_{\bar{x}_t^{(i)}}(\mathrm{d}x_t).$$

Quiz: How to estimate expectations of a given function $\varphi(x_t)$?

Consider the following state-space model

$$
\begin{aligned}
x_0 &\sim \mathcal{N}(x_0; 0, I), \\
x_t | x_{t-1} &\sim \mathcal{N}(x_t; A x_{t-1}, Q), \\
y_t | x_t &\sim \mathcal{N}(y_t; H x_t, R).
\end{aligned}
$$

where

$$
A = \begin{pmatrix} 1 & 0 & \kappa & 0 \\ 0 & 1 & 0 & \kappa \\ 0 & 0 & 0.99 & 0 \\ 0 & 0 & 0 & 0.99 \end{pmatrix} \quad \text{and} \quad Q = \begin{pmatrix} \frac{\kappa^3}{3} & 0 & \frac{\kappa^2}{2} & 0 \\ 0 & \frac{\kappa^3}{3} & 0 & \frac{\kappa^2}{2} \\ \frac{\kappa^2}{2} & 0 & \kappa & 0 \\ 0 & \frac{\kappa^2}{2} & 0 & \kappa \end{pmatrix}
$$

and

$$
H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad \text{and} \quad R = r \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},
$$

where $r = 5$.

Particle filter for this model: Given $x_{1:t-1}^{(i)}$ for $i = 1, \ldots, N$,

- Sample: $x_t^{(i)} \sim \mathcal{N}(x_t; A x_{t-1}^{(i)}, Q)$,
- Compute weights:

$$\mathsf{W}_t^{(i)} = \mathcal{N}(y_t; H x_t^{(i)}, R),$$

  Normalise: $\mathsf{w}_t^{(i)} = \mathsf{W}_t^{(i)} / \sum_{i=1}^N \mathsf{W}_t^{(i)}$

- Report

$$\pi_t^N(\mathrm{d}x_t) = \sum_{i=1}^N \mathsf{w}_t^{(i)} \delta_{x_t^{(i)}}(\mathrm{d}x_t).$$

- Resample:

$$x_t^{(i)} \sim \sum_{i=1}^N \mathsf{w}_t^{(i)} \delta_{\tilde{x}_t^{(i)}}(\mathrm{d}x_t).$$

Let us look the following Lorenz 63 model

$$\begin{aligned} x_{1,t} &= x_{1,t-1} - \gamma \mathsf{s}(x_{1,t} - x_{2,t}) + \sqrt{\gamma}\xi_{1,t}, \\ x_{2,t} &= x_{2,t-1} + \gamma(\mathsf{r} x_{1,t} - x_{2,t} - x_{1,t}x_{3,t}) + \sqrt{\gamma}\xi_{2,t}, \\ x_{3,t} &= x_{3,t-1} + \gamma(x_{1,t}x_{2,t} - \mathsf{b} x_{3,t}) + \sqrt{\gamma}\xi_{3,t}, \end{aligned}$$

where $\gamma = 0.01$, $\mathsf{r} = 28$, $\mathsf{b} = 8/3$, $\mathsf{s} = 10$, and $\xi_{1,t}, \xi_{2,t}, \xi_{3,t} \sim \mathcal{N}(0,1)$ are independent Gaussian random variables. The observation model is given by

$$y_t = [1, 0, 0] x_t + \eta_t,$$

where $\eta_t \sim \mathcal{N}(0, \sigma_y^2)$ is a Gaussian random variable.

Another quantity BPF can estimate is the marginal likelihood:

$$p(y_{1:t}) = \int p(y_{1:t}, x_{0:t}) \mathrm{d}x_{0:t}.$$

This quantity is useful for model selection and model comparison.

Recall tbat we bave tbe factorisation:

$$p(y_{1:t}) = \prod_{k=1}^{t} p(y_k | y_{1:k-1}).$$

where

$$p(y_t | y_{1:t-1}) = \int g(y_t | x_t) \xi_t(x_t | y_{1:t-1}) \mathrm{d}x_t.$$

Recall that we can obtain the approximation of $\xi_t(x_t | y_{1:t-1})$ by the particle filter using predictive particles $\bar{x}_t^{(i)} \sim \tau(x_t | x_{t-1}^{(i)})$ as

$$p_t^N(\mathrm{d}x_t | y_{1:t-1}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{\bar{x}_t^{(i)}}(\mathrm{d}x_t).$$

Therefore, given

$$p_t^N(\mathrm{d}x_t|y_{1:t-1}) = \frac{1}{N}\sum_{i=1}^N \delta_{\bar{x}_t^{(i)}}(\mathrm{d}x_t),$$

we get

$$p^N(y_t|y_{1:t-1}) = \frac{1}{N}\sum_{i=1}^N g(y_t|\bar{x}_t^{(i)}).$$

As a result, we can approximate

$$p^N(y_{1:t}) = \prod_{k=1}^t p^N(y_k|y_{1:k-1}).$$

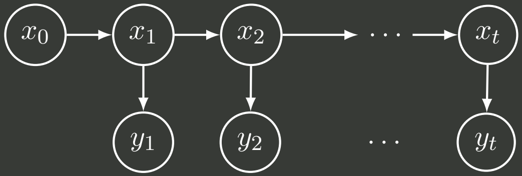
Remarkably, this estimate is unbiased:

$$\mathbb{E}[p^N(y_{1:t})] = p(y_{1:t}).$$

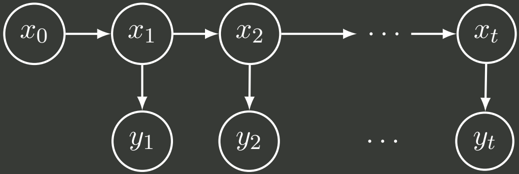
For general (bounded) test functions $\varphi(x_t)$ and filtering measures $\pi_t^N(\mathrm{d}x_t|y_{1:t})$, we have the following L_p bound

$$\|(\varphi, \pi_t^N) - (\varphi, \pi_t)\|_p \leq \frac{c_{t,p}\|\varphi\|_\infty}{\sqrt{N}}.$$

We have seen inference for

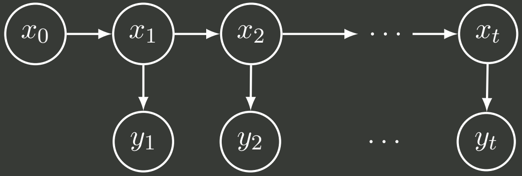


We have seen inference for

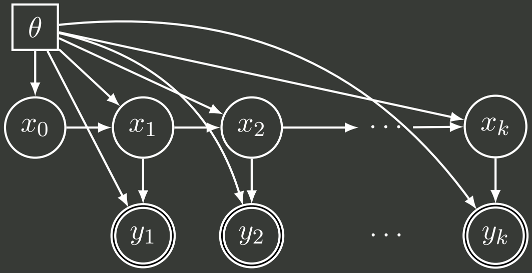


What if the model has parameters θ?

We have seen inference for

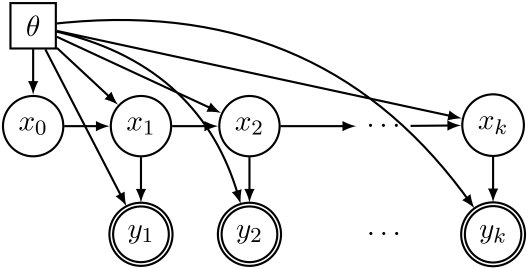


What if the model has parameters θ?



Problem definition

Recap – the model, the notation



We are given the model

$$x_0 \sim \mu_\theta(x_0),$$
$$x_t | x_{t-1} \sim \tau_\theta(x_t | x_{t-1}),$$
$$y_t | x_t \sim g_\theta(y_t | x_t).$$

We aim at estimating θ given $y_{1:T}$.

We are interested in solving the global optimization problem

$$\theta^\star = \underset{\theta \in \Theta}{\operatorname{argmax}} \log p_\theta(y_{1:T}),$$

where

$$p_\theta(y_{1:T}) = \int \gamma_\theta(x_{0:T}, y_{1:T}) \mathrm{d}x_{0:T}.$$

In this lecture, we are interested in gradient-based approaches for maximization of $\log p_\theta(y_{1:T})$.

The smoothing problem

For the maximum-likelihood parameter estimation methods, we often require an approximation of the smoothing distribution $\pi_\theta(x_{0:T}|y_{1:T})$.

For the maximum-likelihood parameter estimation methods, we often require an approximation of the smoothing distribution $\pi_\theta(x_{0:T}|y_{1:T})$.

Wait... Can't we obtain it via the joint sampler we described in the filtering lecture?

For the maximum-likelihood parameter estimation methods, we often require an approximation of the smoothing distribution $\pi_\theta(x_{0:T}|y_{1:T})$.

Wait... Can't we obtain it via the joint sampler we described in the filtering lecture?

Yes, but...

The smoothing problem

Recall how we do it: For $t \geq 2$,

- Sample:

$$\bar{x}_t^{(i)} \sim q_t(x_t | x_{t-1}^{(i)}),$$

- Weight

$$\mathsf{w}_t^{(i)} \propto \frac{\tau_\theta(\bar{x}_t^{(i)} | x_{t-1}^{(i)}) g_\theta(y_t | \bar{x}_t^{(i)})}{q_t(\bar{x}_t^{(i)} | x_{t-1}^{(i)})},$$

- Resample: Choose $a_t^{(i)}$ where $\mathbb{P}(a_t^{(i)} = j) \propto \mathsf{w}_t^j$ and set

$$x_{1:t}^{(i)} = (x_{1:t-1}^{a_t^{(i)}}, \bar{x}_t^{a_t^{(i)}})$$

The entire state history is resampled! What can go wrong?

If we do resampling every step (which is crucial), then we can only do it if we track the genealogy backwards. (?)

- After every resample, we throw away the killed particles' ancestors and replace them with the survivors' ancestors.

Path degeneracy is a big issue.

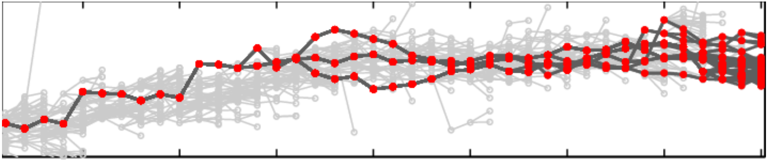


Figure: Source: Svensson, Andreas, Thomas B. Schön, and Manon Kok. "Nonlinear state space smoothing using the conditional particle filter." (2015).

Instead, we can consider the following decomposition

$$\pi_\theta(x_{0:T}|y_{1:T}) = \pi_\theta(x_T|y_{0:T}) \prod_{k=0}^{T-1} \pi_\theta(x_k|y_{0:T}, x_{k+1}),$$
$$= \pi_\theta(x_T|y_{0:T}) \prod_{k=0}^{T-1} \pi_\theta(x_k|y_{0:k}, x_{k+1}). \quad (11)$$

where

$$\pi_\theta(x_t|x_{t+1}, y_{1:t}) = \frac{\pi_\theta(x_t, x_{t+1}|y_{1:t})}{\xi_\theta(x_{t+1}|y_{1:t})}, \quad (12)$$
$$= \frac{\tau_\theta(x_{t+1}|x_t)\pi_\theta(x_t|y_{1:t})}{\xi_\theta(x_{t+1}|y_{1:t})}. \quad (13)$$

$$\pi_\theta(x_{0:T}|y_{1:T}) = \pi_\theta(x_T|y_{0:T}) \prod_{k=0}^{T-1} \pi_\theta(x_k|y_{0:k}, x_{k+1}).$$

This recursion suggests sampling $\pi_\theta(x_T|y_{1:T})$ from the filter and sample backwards from $\pi_\theta(x_k|y_{0:k}, x_{k+1})$ by conditioning on the x_{k+1}. This would provide us a sample $x_{0:T}^{(i)}$ from the smoother.

We approximate the backward distribution as

$$\pi_\theta(\mathrm{d}x_t|x_{t+1}, y_{1:t}) = \frac{\tau_\theta(x_{t+1}|x_t)\pi_\theta^N(\mathrm{d}x_t|y_{1:t})}{\xi_\theta^N(x_{t+1}|y_{1:t})}.$$

where π_θ^N and ξ_θ^N approximate filtering and predictive measures (see next slide).

The smoothing problem

An alternative: Forward filtering backward sampling

Imperial College London

$$\pi_\theta(\mathrm{d}x_t|x_{t+1}, y_{1:t}) = \frac{\tau_\theta(x_{t+1}|x_t)\pi_\theta^N(\mathrm{d}x_t|y_{1:t})}{\int \tau_\theta(x_{t+1}|x_t)\pi_\theta^N(\mathrm{d}x_t|y_{1:t})}$$

Plugging $\pi_\theta^N(\mathrm{d}x_t|y_{1:t}) = \sum_{i=1}^N \mathsf{w}_t^{(i)} \delta_{\bar{x}_t^{(i)}}(\mathrm{d}x_t)$ gives

$$\pi_\theta^N(\mathrm{d}x_t|x_{t+1}, y_{1:t}) = \frac{\sum_{i=1}^N \mathsf{w}_t^{(i)} \tau_\theta(x_{t+1}|\bar{x}_t^{(i)}) \delta_{\bar{x}_t^{(i)}}(\mathrm{d}x_t)}{\sum_{i=1}^N \mathsf{w}_t^{(i)} \tau_\theta(x_{t+1}|\bar{x}_t^{(i)})} \qquad (14)$$

If we use the weighted approximation then the FFBSa is given by

- At time T, sample $\tilde{x}_T \sim \pi_\theta^N(\mathrm{d}x_T | y_{1:T})$,
- t from $T-1$ to 1:
    - Compute smoothing weights

$$\mathsf{w}_{t+1|t}^{(i)} \propto \mathsf{w}_t^{(i)} \tau_\theta(\tilde{x}_{t+1} | \bar{x}_t^{(i)}).$$

    - Then sample

$$\tilde{x}_t \sim \sum_{i=1}^N \mathsf{w}_{t+1|t}^{(i)} \delta_{\bar{x}_t^{(i)}}(\mathrm{d}x_t).$$

The sample $\tilde{x}_{0:T}$ is a sample from the smoother. However, it is just a single sample!

Do the same N times. Reduces path degeneracy, but $\mathcal{O}(N^2(T+1))$.

Recall the original smoothing recursions we discussed:

$$
\begin{aligned}
\pi_\theta(x_t|y_{1:T}) &= \int \pi_\theta(x_t, x_{t+1}|y_{1:T})\mathrm{d}x_{t+1}, \\
&= \int \pi_\theta(x_t|x_{t+1}, y_{1:t})\pi_\theta(x_{t+1}|y_{1:T})\mathrm{d}x_{t+1}, \\
&= \int \frac{\tau_\theta(x_{t+1}|x_t)\pi_\theta(x_t|y_{1:t})}{\xi_\theta(x_{t+1}|y_{1:t})}\pi_\theta(x_{t+1}|y_{1:T})\mathrm{d}x_{t+1}.
\end{aligned}
$$

The smoothing problem

Another alternative: Forward filtering backward smoothing

Recall the original smoothing recursions we discussed:

$$\begin{aligned} \pi_\theta(x_t|y_{1:T}) &= \int \pi_\theta(x_t, x_{t+1}|y_{1:T}) \mathrm{d}x_{t+1}, \\ &= \int \pi_\theta(x_t|x_{t+1}, y_{1:t}) \pi_\theta(x_{t+1}|y_{1:T}) \mathrm{d}x_{t+1}, \\ &= \int \frac{\tau_\theta(x_{t+1}|x_t)\pi_\theta(x_t|y_{1:t})}{\xi_\theta(x_{t+1}|y_{1:t})} \pi_\theta(x_{t+1}|y_{1:T}) \mathrm{d}x_{t+1}. \end{aligned}$$

Can we use these to build a particle approximation?

Recall the original smoothing recursions we discussed:

$$\begin{aligned}\pi_\theta(x_t|y_{1:T}) &= \int \pi_\theta(x_t, x_{t+1}|y_{1:T})\mathrm{d}x_{t+1}, \\ &= \int \pi_\theta(x_t|x_{t+1}, y_{1:t})\pi_\theta(x_{t+1}|y_{1:T})\mathrm{d}x_{t+1}, \\ &= \int \frac{\tau_\theta(x_{t+1}|x_t)\pi_\theta(x_t|y_{1:t})}{\xi_\theta(x_{t+1}|y_{1:t})}\pi_\theta(x_{t+1}|y_{1:T})\mathrm{d}x_{t+1}.\end{aligned}$$

Can we use these to build a particle approximation? Recall measure theoretic form

$$\pi_\theta(\mathrm{d}x_t|y_{1:T}) = \pi_\theta(\mathrm{d}x_t|y_{1:t}) \int \frac{\tau_\theta(x_{t+1}|x_t)}{\xi_\theta(x_{t+1}|y_{1:t})}\pi_\theta(x_{t+1}|y_{1:T})\mathrm{d}x_{t+1}.$$

Backward recursion

$$\pi_\theta(\mathrm{d}x_t|y_{1:T}) = \pi_\theta(\mathrm{d}x_t|y_{1:t}) \int \frac{\tau_\theta(x_{t+1}|x_t)}{\int \tau_\theta(x_{t+1}|x_t)\pi_\theta(\mathrm{d}x_t|y_{1:t})} \pi_\theta(\mathrm{d}x_{t+1}|y_{1:T}).$$

Backward recursion

$$\pi_\theta(\mathrm{d}x_t|y_{1:T}) = \pi_\theta(\mathrm{d}x_t|y_{1:t}) \int \frac{\tau_\theta(x_{t+1}|x_t)}{\int \tau_\theta(x_{t+1}|x_t)\pi_\theta(\mathrm{d}x_t|y_{1:t})} \pi_\theta(\mathrm{d}x_{t+1}|y_{1:T}).$$

This means that we can use approximations $\{\pi_\theta^N(\mathrm{d}x_t|y_{1:t})\}_{t=1}^T$ again to recursively update the smoother backwards in time and construct the smoother update

$$\pi_\theta(\mathrm{d}x_{t+1}|y_{1:T}) \mapsto \pi_\theta(\mathrm{d}x_t|y_{1:T}).$$

Assume we have an approximation

$$\pi_\theta^N(\mathrm{d}x_{t+1}|y_{1:T}) = \sum_{i=1}^N \mathsf{w}_{t+1|T}^{(i)} \delta_{\bar{x}_{t+1}^{(i)}}(\mathrm{d}x_{t+1}).$$

where $\mathsf{w}_{T|T}^{(i)} = \mathsf{w}_T^{(i)}$. We can use the recursion in the previous slide to obtain

$$\pi_\theta(\mathrm{d}x_t|y_{1:T}) = \sum_{i=1}^N \mathsf{w}_{t|T}^{(i)} \delta_{\bar{x}_t^{(i)}}(\mathrm{d}x_t),$$

where

$$\mathsf{w}_{t|T}^{(i)} = \mathsf{w}_t^{(i)} \sum_{j=1}^N \frac{\mathsf{w}_{t+1|T}^{(j)} \tau_\theta(\bar{x}_{t+1}^{(j)}|\bar{x}_t^{(i)})}{\sum_{l=1}^N \mathsf{w}_t^{(l)} \tau_\theta(\bar{x}_{t+1}^{(j)}|\bar{x}_t^{(l)})}$$

Recall we are interested in solving the global optimization problem

$$\theta^\star = \operatorname*{argmax}_{\theta \in \Theta} \log p_\theta(y_{1:T}),$$

where

$$p_\theta(y_{1:T}) = \int \gamma_\theta(x_{0:T}, y_{1:T}) \mathrm{d}x_{0:T}.$$

A generic way to do this would be to run

$$\theta_{i+1} = \theta_i + \gamma \nabla \log p_\theta(y_{1:T}).$$

- Well understood gradient scheme,
- Can be also replaced by an adaptive gradient scheme. (Adam, your favourite one...)

However, the gradient is not computable...

How to compute the gradient?

For this maximization, we will be interested in computing

$$\nabla_\theta \log p_\theta(y_{1:T}).$$

For this, we use Fisher's identity.

How to compute the gradient?

Proposition 1 (Fisher's identity)

Under appropriate regularity conditions, we have

$$\nabla_\theta \log p_\theta(y_{1:T}) = \int \nabla_\theta \log \gamma_\theta(x_{0:T}, y_{1:T}) p_\theta(x_{0:T}|y_{1:T}) dx_{0:T}.$$

How to compute the gradient?

Proof.

Let us note that

$$
\begin{aligned}
\nabla_\theta \log p_\theta(y_{1:T}) &= \frac{\nabla_\theta p_\theta(y_{1:T})}{p_\theta(y_{1:T})}, \\
&= \frac{\nabla \int \gamma_\theta(x_{0:T}, y_{1:T}) \mathrm{d}x_{0:T}}{p_\theta(y_{1:T})}, \\
&= \int \frac{\nabla \gamma_\theta(x_{0:T}, y_{1:T})}{p_\theta(y_{1:T})} \mathrm{d}x_{0:T}, \\
&= \int \frac{\nabla \log \gamma_\theta(x_{0:T}, y_{1:T}) \gamma_\theta(x_{0:T}, y_{1:T})}{p_\theta(y_{1:T})} \mathrm{d}x_{0:T}, \\
&= \int \nabla \log \gamma_\theta(x_{0:T}, y_{1:T}) \pi_\theta(x_{0:T} | y_{1:T}) \mathrm{d}x_{0:T}.
\end{aligned}
$$

■

Given Fisher's identity,

$$\nabla_\theta \log \gamma_\theta(y_{1:T}) = \int \nabla_\theta \log \gamma_\theta(x_{0:T}, y_{1:T}) \pi_\theta(x_{0:T}|y_{1:T}) \mathsf{d}x_{0:T}.$$

and

$$\log p_\theta(x_{0:T}, y_{1:T}) = \log \mu_\theta(x_0) + \sum_{t=1}^{T} \log \tau_\theta(x_t|x_{t-1}) + \sum_{t=1}^{T} \log g_\theta(y_t|x_t),$$

How to compute the gradient?

Given

$$\log p_\theta(x_{0:T}, y_{1:T}) = \log \mu_\theta(x_0) + \sum_{t=1}^{T} \log \tau_\theta(x_t|x_{t-1}) + \sum_{t=1}^{T} \log g_\theta(y_t|x_t),$$

Some shortcut notation:

$$s_1^\theta(x_{-1}, x_0) = s_0^\theta(x_0) = \nabla \log \mu_\theta(x_0),$$
$$s_{\theta,t}(x_{t-1}, x_t) = \nabla \log g_\theta(y_t|x_t) + \nabla \log \tau_\theta(x_t|x_{t-1}).$$

So finally the gradient can be written as an expectation

$$\nabla_\theta \log p_\theta(y_{1:T}) = \int \nabla_\theta \log p_\theta(x_{0:T}, y_{1:T}) p_\theta(x_{0:T}|y_{1:T}) \mathsf{d}x_{0:T}.$$

We identify the marginal likelihood as an additive functional

$$\begin{aligned}\nabla_\theta \log p_\theta(y_{1:T}) &= S_T^\theta(x_{1:T}), \\ &= \int_{\mathsf{X}^{T+1}} \left(\sum_{t=1}^{T} s_t^\theta(x_{t-1}, x_t) \right) \pi_\theta(x_{0:T}|y_{1:T}) \mathrm{d}x_{0:T}.\end{aligned}$$

But how do we compute? Recall

$$s_t^\theta(x_{t-1}, x_t) = \nabla \log g_\theta(y_t|x_t) + \nabla \log \tau_\theta(x_t|x_{t-1}).$$

The BPF with parameter gradient computation. Fix θ and assume $\{X_{1:t-1}^{(i)}, \alpha_{t-1}^{(i)}\}$ are given.

- Sample: $\bar{x}_t^{(i)} \sim \tau_\theta(x_t|x_{t-1}^{(i)})$.
- Weight $\mathsf{w}_t^{(i)} \propto g(y_t|\bar{x}_t^{(i)})$.
- Resample:

$$x_t^{(i)} \sim \sum_{i=1}^N \mathsf{w}_t^{(i)} \delta_{\bar{x}_t^{(i)}}(\mathrm{d}x_t),$$

i.e. $x_t^{(i)} = \bar{x}_t^{a_t^{(i)}}$ with $\mathbb{P}(a_t^{(i)} = j) = \mathsf{w}_t^j$ and construct the estimate

$$\alpha_t^{(i)} = \alpha_{t-1}^{a_t^{(i)}} + s_t^\theta(x_{t-1}^{a_t^{(i)}}, x_t^{(i)})$$

How to compute the gradient?

Then

$$S_T^{\theta,N} = \frac{1}{N}\sum_{i=1}^{N}\alpha_T^{(i)}$$

However, as this naive "forward smoother" $\mathcal{O}(N)$ iteration complexity) suffers from path degeneracy as we discussed before, therefore the estimates will not be reliable.

Use FFBS described before however the computation won't be recursive (it is offline) and $\mathcal{O}(N^2)$ complexity - but has better properties.

There is a method called forward smoothing, which can build the smoothed additive functional expectations *online*. Let us go back and write, for $n < T$,

$$\begin{aligned}\nabla_\theta \log p_\theta(y_{1:n}) &= S_T^\theta(x_{1:n}),\\ &= \int_{\mathsf{X}^{n+1}} \left(\sum_{t=1}^{n} s_t^\theta(x_{t-1}, x_t)\right) \pi_\theta(x_{0:n}|y_{1:n}) \mathrm{d}x_{0:n},\\ &= \int V_n^\theta(x_n) \pi_\theta(x_n|y_{1:n}) \mathrm{d}x_n.\end{aligned}$$

where

$$V_n^\theta(x_n) = \int \left(\sum_{k=1}^{n} s_k(x_{k-1}, x_k)\right) p_\theta(x_{0:n-1}|y_{0:n-1}, x_n) \mathrm{d}x_{0:n-1}.$$

The key recursion, note that

$$V_{n+1}^{\theta}(x_{n+1}) = \int \left(\sum_{k=1}^{n+1} s_k(x_{k-1}, x_k) \right) p_\theta(x_{0:n}|y_{0:n}, x_{n+1}) \mathrm{d}x_{0:n},$$

$$= \int \left(\sum_{k=1}^{n} s_k(x_{k-1}, x_k) + s_n(x_{n-1}, x_n) \right)$$

$$p_\theta(x_{0:n-1}|y_{0:n-1}, x_n) \mathrm{d}x_{0:n-1} p_\theta(x_n|y_{0:n}, x_{n+1}) \mathrm{d}x_n,$$

$$= \int \left(V_n^{\theta}(x_n) + s_n(x_{n-1}, x_n) \right) p_\theta(x_n|y_{0:n}, x_{n+1}) \mathrm{d}x_n.$$

We have a recursion for $(V_n^{\theta})_{n \geq 1}$ that can be estimated online using $(x_t^{(i)}, x_{t+1}^{(i)})$.

How do compute things only forward pass? Recall FFBS

- At time T, sample $\tilde{x}_T \sim \pi_\theta^N(\mathrm{d}x_T|y_{1:T})$,
- t from $T-1$ to 1:
  - Compute smoothing weights

$$\mathsf{w}_{t+1|t}^{(i)} \propto \mathsf{w}_t^{(i)} \tau_\theta(\tilde{x}_{t+1}|\bar{x}_t^{(i)}).$$

  - Then sample

$$\tilde{x}_t \sim \sum_{i=1}^N \mathsf{w}_{t+1|t}^{(i)} \delta_{\bar{x}_t^{(i)}}(\mathrm{d}x_t).$$

How to compute the gradient?

Forward only smoothing: Assume we have a good approximation of $V_t^\theta(x_t^{(i)})$.

- Sample $\bar{x}_{t+1}^{(i)} \sim f(\cdot|x_t^{(i)})$,
- Use it to compute FFBS smoothing weights (with predictive particles)

$$\mathsf{w}_{t+1|t}^{(i)} \propto \mathsf{w}_t^{(i)} \tau_\theta(\bar{x}_{t+1}^{(i)}|x_t^{(i)}).$$

and

$$V_{t+1}^\theta(\bar{x}_{t+1}^{(i)}) = \sum_{j=1}^N \mathsf{w}_{t+1|t}^{(i)} \left(V_t^\theta(x_t^{(i)}) + s_{t+1}(x_t^{(i)}, x_{t+1}^{(i)}) \right).$$

and build

$$S_{t+1}^{\theta,N} = \sum_{j=1}^N \mathsf{w}_{t+1}^{(i)} V_t^\theta(x_{t+1}^{(i)}).$$

How to compute the gradient?

Forward only smoothing: Assume we have a good approximation of $V_t^\theta(x_t^{(i)})$.

- Sample $\bar{x}_{t+1}^{(i)} \sim f(\cdot|x_t^{(i)})$,
- Use it to compute FFBS smoothing weights (with predictive particles)

$$\mathsf{w}_{t+1|t}^{(i)} \propto \mathsf{w}_t^{(i)} \tau_\theta(\bar{x}_{t+1}^{(i)}|x_t^{(i)}).$$

and

$$V_{t+1}^\theta(\bar{x}_{t+1}^{(i)}) = \sum_{j=1}^N \mathsf{w}_{t+1|t}^{(i)} \left(V_t^\theta(x_t^{(i)}) + s_{t+1}(x_t^{(i)}, x_{t+1}^{(i)}) \right).$$

and build

$$S_{t+1}^{\theta,N} = \sum_{j=1}^N \mathsf{w}_{t+1}^{(i)} V_t^\theta(x_{t+1}^{(i)}).$$

Forward smoothing.

An approximation

$$\theta_{t+1} = \theta_t + \gamma \nabla \log p_{\theta_{0:t}}(y_t | y_{1:t-1})$$

where

$$\nabla \log p_{\theta_{0:t}}(y_t | y_{1:t-1}) = \nabla p_{\theta_{0:t}}(y_{1:t}) - \nabla p_{\theta_{0:t-1}}(y_{1:t-1}).$$

The definition

$$\nabla p_{\theta_{0:t}}(y_{1:t}) = \pi_{\theta,t} \left(\sum_{k=1}^{t} s_k^{\theta_k}(x_{t-1}, x_t) \right),$$

therefore

$$\nabla \log p_{\theta_{0:t}}(y_t | y_{1:t-1}) = \mathbb{E}\left[s_t^{\theta_t}(x_{t-1}, x_t) \right]$$

References I