MFC CDT Probability and Statistics
Week 9

O. Deniz Akyildiz

Mathematics for our Future Climate: Theory, Data and Simulation (MFC CDT).

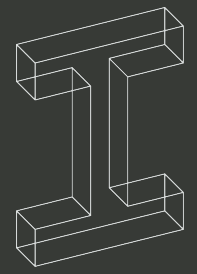
December 2, 2024

IMPERIAL

https://akyildiz.me/

𝕏: @odakyildiz

Recall our basic task:



Recall our basic task:

- We wanted to sample from distributions $\pi(x) \propto \gamma(x)$ given only the knowledge of $\gamma(x)$ and use these samples to estimate an integral

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

for some function φ.

Recall our basic task:

- We wanted to sample from distributions $\pi(x) \propto \gamma(x)$ given only the knowledge of $\gamma(x)$ and use these samples to estimate an integral

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

  for some function φ.

We will now consider *dynamic* settings.

- We will have a *sequence* of distributions $(\pi)_{t\geq 0}$ and we will want to estimate the integrals

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

  for some function φ.

Recall our basic task:

- We wanted to sample from distributions $\pi(x) \propto \gamma(x)$ given only the knowledge of $\gamma(x)$ and use these samples to estimate an integral

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

  for some function φ.

We will now consider *dynamic* settings.

- We will have a *sequence* of distributions $(\pi)_{t\geq 0}$ and we will want to estimate the integrals

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

  for some function φ.

This will include the filtering problem.

State-space models

problem definition

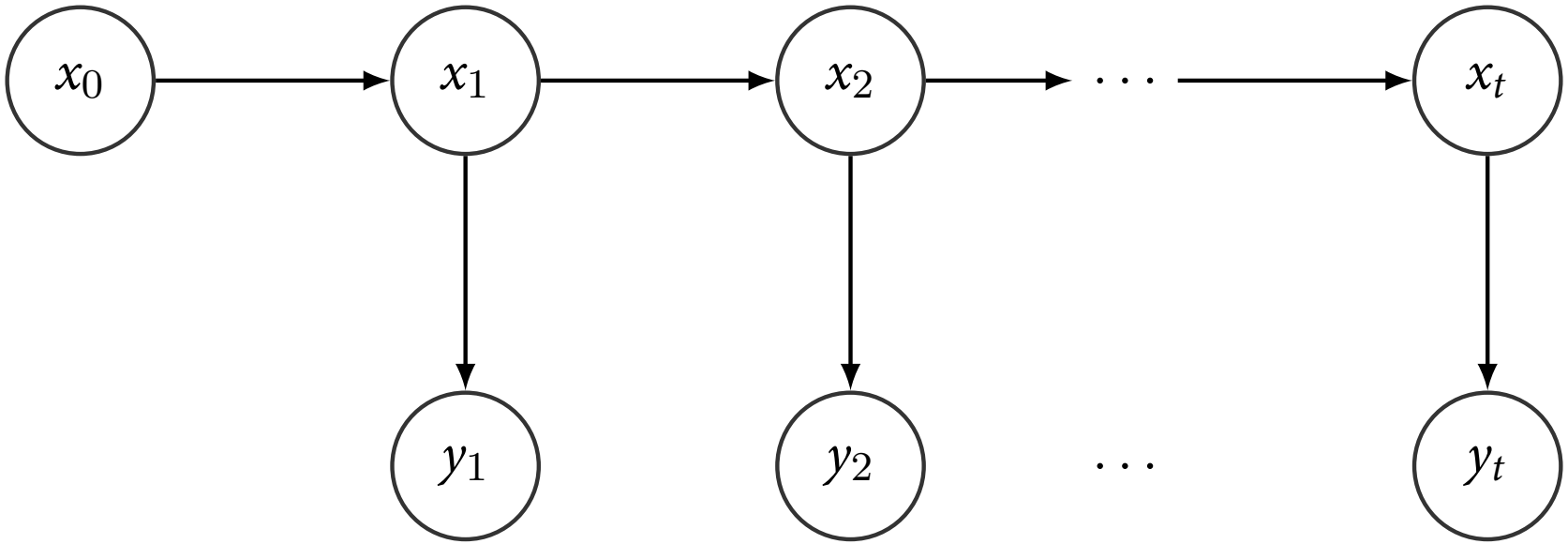


Figure: The conditional independence structure of a state-space model.

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

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

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

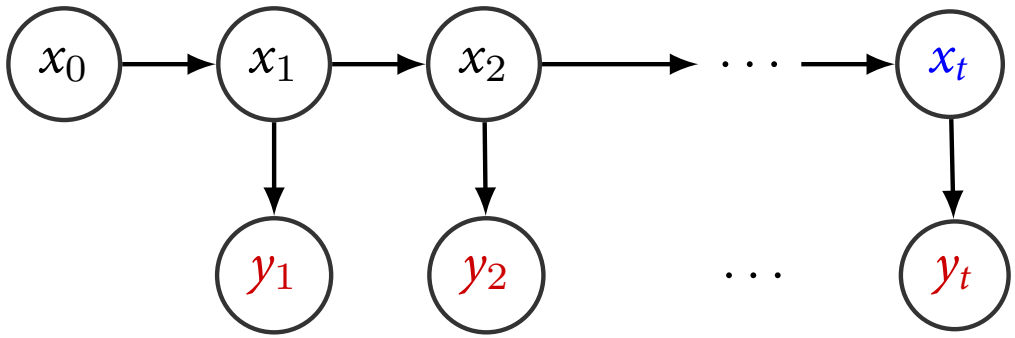
State-space models

problem definition

We are interested estimating expectations,

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

sequentially as new data arrives. This problem is known as *the filtering problem*.



A simpler problem

Sequential inference

Let us first consider a generic probabilistic setting,

$$\pi_0(x) \qquad \text{and} \qquad g_t(y_t|x).$$

for $(y_t)_{t\in\mathbb{N}_+}$ a sequence of observations.

A simpler problem

Sequential inference

Let us first consider a generic probabilistic setting,

$$\pi_0(x) \qquad \text{and} \qquad g_t(y_t|x).$$

for $(y_t)_{t\in\mathbb{N}_+}$ a sequence of observations. We are interested in estimating expectations,

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

A simpler problem

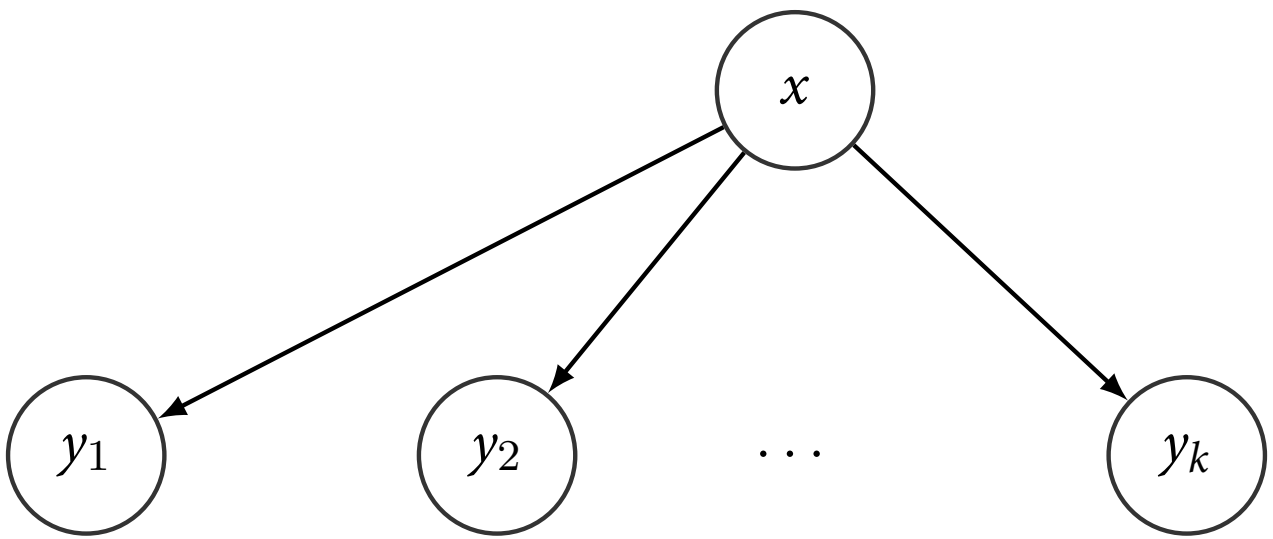
Sequential inference

Let us first consider a generic probabilistic setting,

$$\pi_0(x) \qquad \text{and} \qquad g_t(y_t|x).$$

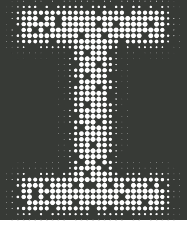
for $(y_t)_{t\in\mathbb{N}_+}$ a sequence of observations. We are interested in estimating expectations,

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



A simpler problem

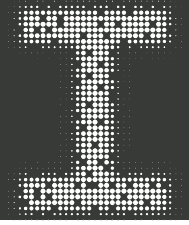
Sequential inference



Mini-quiz: How would you obtain $\pi(x|y_{1:t})$?

A simpler problem

Sequential inference



Mini-quiz: How would you obtain $\pi(x|y_{1:t})$?

We can use Bayes' rule iteratively

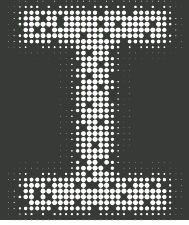
$$\begin{aligned}\pi(x|y_{1:t}) &= \frac{\gamma(x, y_{1:t})}{p(y_{1:t})}, \\ &= \frac{g_t(y_t|x)\gamma(x, y_{1:t-1})}{p(y_t|y_{1:t-1})p(y_{1:t-1})}, \\ &= \frac{g_t(y_t|x)\pi(x|y_{1:t-1})}{p(y_t|y_{1:t-1})}.\end{aligned}$$

where

$$p(y_t|y_{1:t-1}) = \int g_t(y_t|x)\pi(x|y_{1:t-1})\mathrm{d}x.$$

A simpler problem

Sequential inference



Mini-quiz: How would you obtain $\pi(x|y_{1:t})$?

We can use Bayes' rule iteratively

$$\begin{aligned}\pi(x|y_{1:t}) &= \frac{\gamma(x, y_{1:t})}{p(y_{1:t})}, \\ &= \frac{g_t(y_t|x)\gamma(x, y_{1:t-1})}{p(y_t|y_{1:t-1})p(y_{1:t-1})}, \\ &= \frac{g_t(y_t|x)\pi(x|y_{1:t-1})}{p(y_t|y_{1:t-1})}.\end{aligned}$$

where

$$p(y_t|y_{1:t-1}) = \int g_t(y_t|x)\pi(x|y_{1:t-1})\mathrm{d}x.$$

The previous posterior $\pi(x|y_{1:t-1})$ is used as the prior for the next step.

A simpler problem

Sequential inference: the Gaussian case

Let us assume that

$$\pi_0(x) = \mathcal{N}(x; \mu_0, V_0),$$
$$g_t(y_t|x) = \mathcal{N}(y_t; H_t x, R_t).$$

A simpler problem

Sequential inference: the Gaussian case

Let us assume that

$$\pi_0(x) = \mathcal{N}(x; \mu_0, V_0),$$
$$g_t(y_t|x) = \mathcal{N}(y_t; H_t x, R_t).$$

Can we compute $\pi(x|y_{1:t})$ analytically?

A simpler problem

Sequential inference: the Gaussian case

Let us assume that

$$\pi_0(x) = \mathcal{N}(x; \mu_0, V_0),$$
$$g_t(y_t|x) = \mathcal{N}(y_t; H_t x, R_t).$$

Can we compute $\pi(x|y_{1:t})$ analytically?

Lemma 1

We obtain $\pi(x|y_{1:t}) = \mathcal{N}(x; \mu_t, V_t)$ *where,*

$$\mu_t = \mu_{t-1} + V_{t-1}H_t^\top (R_t + H_t V_{t-1} H_t^\top)^{-1}(y_t - H_t \mu_{t-1}),$$
$$V_t = V_{t-1} - V_{t-1}H_t^\top (R_t + H_t V_{t-1} H_t^\top)^{-1} H_t V_{t-1},$$

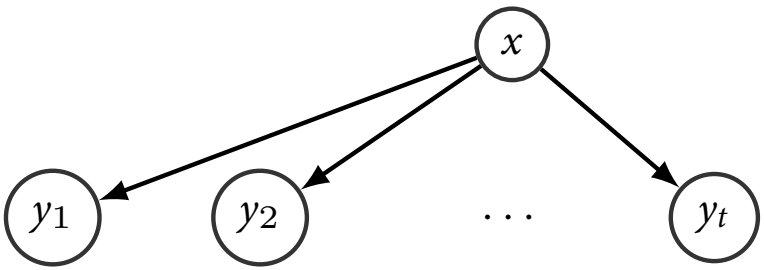
for $t \geq 1$.

Static vs. dynamic setting

Static inference: Given a probability model,

$$x \sim \pi_0(\mathrm{d}x),$$
$$y_t|x_t \sim g_t(y_t|x),$$

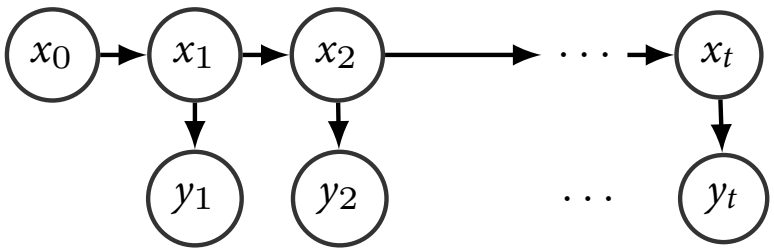
we are interested in static inference: Estimating $\pi(x|y_{1:t})$ sequentially.



Dynamic inference: Given a SSM,

$$x_0 \sim \pi_0(\mathrm{d}x_0),$$
$$x_t|x_{t-1} \sim \tau_t(\mathrm{d}x_t|x_{t-1}),$$
$$y_t|x_t \sim g_t(y_t|x_t),$$

we are interested in *the stochastic filtering problem*: Estimating $\pi_t(x_t|y_{1:t})$.



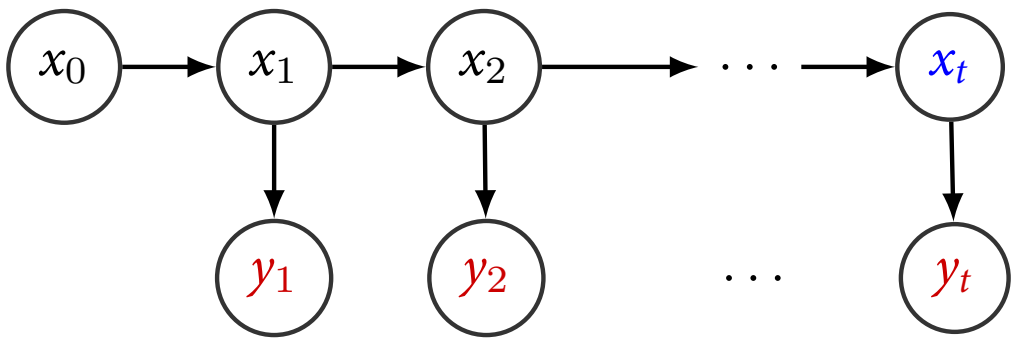
State-space models

Algorithmic principle

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

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

Let us look in detail to these steps:

Prediction: Given $\pi_{t-1}(\mathrm{d}x_{t-1}|y_{1:t-1})$, we want to compute $\pi_t(\mathrm{d}x_t|y_{1:t-1})$.

Let us look in detail to these steps:

Prediction: Given $\pi_{t-1}(\mathrm{d}x_{t-1}|y_{1:t-1})$, we want to compute $\pi_t(\mathrm{d}x_t|y_{1:t-1})$.

$$\pi_t(\mathrm{d}x_t|y_{1:t-1}) = \int \pi_{t-1}(\mathrm{d}x_{t-1}|y_{1:t-1})\tau_t(\mathrm{d}x_t|x_{t-1}).$$

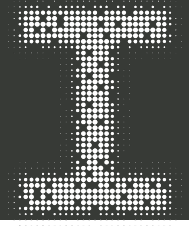
Let us look in detail to these steps:

Prediction: Given $\pi_{t-1}(\mathrm{d}x_{t-1}|y_{1:t-1})$, we want to compute $\pi_t(\mathrm{d}x_t|y_{1:t-1})$.

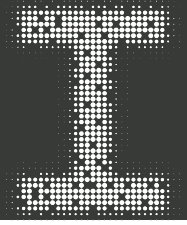
$$\pi_t(\mathrm{d}x_t|y_{1:t-1}) = \int \pi_{t-1}(\mathrm{d}x_{t-1}|y_{1:t-1})\tau_t(\mathrm{d}x_t|x_{t-1}).$$

In terms of densities

$$\pi_t(x_t|y_{1:t-1}) = \int \pi_{t-1}(x_{t-1}|y_{1:t-1})\tau_t(x_t|x_{t-1})\mathrm{d}x_{t-1}.$$



We have already seen the update rule, but we modify this in the dynamic setting: Our prior will now be the predictive distribution $\pi_t(\mathrm{d}x_t|y_{1:t-1})$.



We have already seen the update rule, but we modify this in the dynamic setting: Our prior will now be the predictive distribution $\pi_t(\mathrm{d}x_t|y_{1:t-1})$.

Update: Given $\pi_t(\mathrm{d}x_t|y_{1:t-1})$, we want to compute $\pi_t(\mathrm{d}x_t|y_{1:t})$.

We have already seen the update rule, but we modify this in the dynamic setting: Our prior will now be the predictive distribution $\pi_t(\mathrm{d}x_t|y_{1:t-1})$.

Update: Given $\pi_t(\mathrm{d}x_t|y_{1:t-1})$, we want to compute $\pi_t(\mathrm{d}x_t|y_{1:t})$.

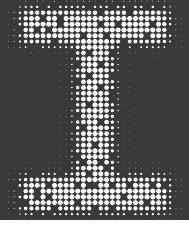
$$\begin{aligned}\pi_t(x_t|y_{1:t}) &= \frac{\gamma(x_t, y_{1:t})}{p(y_{1:t})}, \\ &= \frac{g_t(y_t|x_t)\pi_t(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}.\end{aligned}$$

where

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

Let us assume that

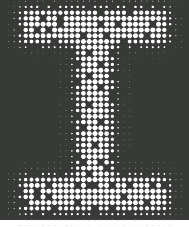
$$\begin{aligned}\pi_0(x) &= \mathcal{N}(x; \mu_0, V_0),\\ \tau_t(x_t|x_{t-1}) &= \mathcal{N}(x_t; A_t x_{t-1}, Q_t),\\ g_t(y_t|x_t) &= \mathcal{N}(y_t; H_t x_t, R_t).\end{aligned}$$



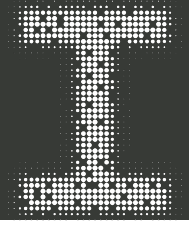
Let us assume that

$$\begin{aligned}\pi_0(x) &= \mathcal{N}(x; \mu_0, V_0),\\ \tau_t(x_t|x_{t-1}) &= \mathcal{N}(x_t; A_t x_{t-1}, Q_t),\\ g_t(y_t|x_t) &= \mathcal{N}(y_t; H_t x_t, R_t).\end{aligned}$$

Can we compute $\pi(x|y_{1:t})$ analytically?



Yes, we can!



Yes, we can!

Lemma 2

Given the optimal filter $\pi_{t-1}(x_{t-1}|y_{1:t-1}) = \mathcal{N}(x_{t-1}; \mu_{t-1}, V_{t-1})$ *at time* $t-1$ *the predictive distribution* $\xi_t(x_t|y_{1:t-1})$ *is given by*

$$\xi_t(x_t|y_{1:t-1}) = \mathcal{N}(x_t; \tilde{\mu}_t, \tilde{V}_t),$$

where,

$$\tilde{\mu}_t = A_t \mu_{t-1}, \tag{1}$$

$$\tilde{V}_t = A_t V_{t-1} A_t^\top + Q_t. \tag{2}$$

Lemma 3

Finally, the optimal filter $\pi_t(x_t|y_{1:t})$ *is given by*

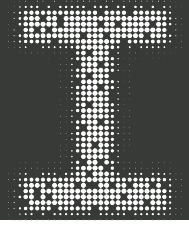
$$\pi_t(x_t|y_{1:t-1}) = \mathcal{N}(x_t; \mu_t, V_t),$$

where,

$$\mu_t = \tilde{\mu}_t + \tilde{V}_t H_t^\top (R_t + H_t \tilde{V}_t H_t^\top)^{-1}(y_t - H_t \tilde{\mu}_t), \tag{3}$$

$$V_t = \tilde{V}_t - \tilde{V}_t H_t^\top (R_t + H_t \tilde{V}_t H_t^\top)^{-1} H_t \tilde{V}_t, \tag{4}$$

from Lemma 1.



What if nonlinearities exist in Gaussian models?

$$\begin{aligned}\pi_0(x) &= \mathcal{N}(x; \mu_0, V_0),\\ \tau_t(x_t|x_{t-1}) &= \mathcal{N}(x_t; a_t(x_{t-1}), Q_t),\\ g_t(y_t|x_t) &= \mathcal{N}(y_t; h_t(x_t), R_t).\end{aligned}$$

Can we still do analytical computations?

What if nonlinearities exist in Gaussian models?

$$\begin{aligned}\pi_0(x) &= \mathcal{N}(x; \mu_0, V_0),\\ \tau_t(x_t|x_{t-1}) &= \mathcal{N}(x_t; a_t(x_{t-1}), Q_t),\\ g_t(y_t|x_t) &= \mathcal{N}(y_t; h_t(x_t), R_t).\end{aligned}$$

Can we still do analytical computations?

Yes! We can use the *extended Kalman filter* (EKF) or the *unscented Kalman filter* (UKF).

Assume that we are given the SSM

$$\begin{aligned}\pi_0(x_0) &= \mathcal{N}(x_0; \mu_0, V_0),\\ \tau_t(x_t|x_{t-1}) &= \mathcal{N}(x_t; a_t(x_{t-1}), Q_t)\\ g_t(y_t|x_t) &= \mathcal{N}(y_t; h_t(x_t), R_t).\end{aligned}$$

where $a_t : \mathrm{X} \to \mathrm{X}$, $h_t : \mathrm{X} \to \mathrm{Y}$, $Q_t \in \mathbb{R}^{d_x \times d_x}$, and $R_t \in \mathbb{R}^{d_y \times d_y}$. Assume that the approximate posterior distribution at time $t-1$ is $\pi^E_{t-1}(x_{t-1}) = \mathcal{N}(x_{t-1}; \mu^E_{t-1}, V^E_{t-1})$.

If the model is approximately locally linear, one can linearize $a_t(x_t)$ around μ^E_{t-1} and obtain the dynamical model

$$\bar{a}_t(x_t) = a_t(\mu^E_{t-1}) + A_t(x_t - \mu^E_{t-1}) = a_t(\mu^E_{t-1}) + A_t x_t - A_t \mu^E_{t-1}, \tag{5}$$

where

$$A_t = \left.\frac{\partial a_t(x)}{\partial x}\right|_{x=\mu^E_{t-1}}.$$

We can see (5) as a linear model with control inputs. Hence, the prediction step with this linearized model simply becomes

$$\tilde{\mu}^E_t = a_t(\mu^E_{t-1}).$$

The uncertainty is propagated also as in the KF, since (5) is a linear model, hence we obtain

$$\tilde{V}_t^E = A_t V_{t-1}^E A_t^\top + Q_t.$$

Similarly, given $\tilde{\mu}_t^E$, in order to proceed with the observation model we can linearize h_t around $\tilde{\mu}_t^E$, i.e., we construct

$$\bar{h}_t(x_t) = h_t(\tilde{\mu}_t^E) + H_t(x_t - \tilde{\mu}_t^E),$$

where

$$H_t = \left.\frac{\partial h_t(x)}{\partial x}\right|_{x=\tilde{\mu}_t}.$$

Given the linearization, the EKF update step now becomes

$$\mu_t^E = \tilde{\mu}_{t-1}^E + \tilde{V}_t^E H_t^\top (R_t + H_t \tilde{V}_t^E H_t^\top)^{-1}(y_t - h_t(\tilde{\mu}_t^E)),$$
$$V_t^E = \tilde{V}_t^E - \tilde{V}_t^E H_t^\top (R_t + H_t \tilde{V}_t^E H_t^\top)^{-1} H_t \tilde{V}_t^E.$$

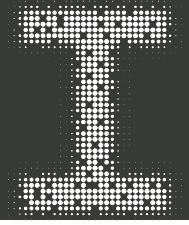
Finally, one can compactly summarize the EKF as follows. Given $\pi^E_{t-1}(x_{t-1}) = \mathcal{N}(x_{t-1}; \mu^E_{t-1}, V^E_{t-1})$, the new posterior pdf $\pi^E_t(x_t) = \mathcal{N}(x_t; \mu^E_t, V^E_t)$ is obtained via

$$\tilde{\mu}^E_t = a_t(\mu^E_{t-1}), \tag{6}$$

$$\tilde{V}^E_t = A_t V^E_{t-1} A_t^\top + Q_t, \tag{7}$$

$$\mu^E_t = \tilde{\mu}^E_{t-1} + \tilde{V}^E_t H_t^\top (R_t + H_t \tilde{V}^E_t H_t^\top)^{-1}(y_t - h_t(\tilde{\mu}^E_t)), \tag{8}$$

$$V^E_t = \tilde{V}^E_t - \tilde{V}^E_t H_t^\top (R_t + H_t \tilde{V}^E_t H_t^\top)^{-1} H_t \tilde{V}^E_t. \tag{9}$$



Other kinds of Gaussian approximations are very popular:

- *Unscented Kalman filter* (UKF): The UKF uses a deterministic sampling technique called the *unscented transform* to obtain a Gaussian approximation of the posterior distribution.

Other kinds of Gaussian approximations are very popular:

- *Unscented Kalman filter* (UKF): The UKF uses a deterministic sampling technique called the *unscented transform* to obtain a Gaussian approximation of the posterior distribution.
- *Gaussian sum filter* (GSF): The GSF uses a Gaussian mixture approximation of the posterior distribution.

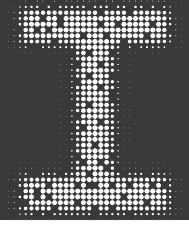
Other kinds of Gaussian approximations are very popular:

- *Unscented Kalman filter* (UKF): The UKF uses a deterministic sampling technique called the *unscented transform* to obtain a Gaussian approximation of the posterior distribution.
- *Gaussian sum filter* (GSF): The GSF uses a Gaussian mixture approximation of the posterior distribution.
- *ensemble Kalman filter* (EnKF): The EnKF uses a Monte Carlo approximation of the posterior distribution.

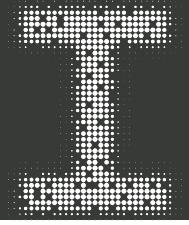
Other kinds of Gaussian approximations are very popular:

- *Unscented Kalman filter* (UKF): The UKF uses a deterministic sampling technique called the *unscented transform* to obtain a Gaussian approximation of the posterior distribution.
- *Gaussian sum filter* (GSF): The GSF uses a Gaussian mixture approximation of the posterior distribution.
- *ensemble Kalman filter* (EnKF): The EnKF uses a Monte Carlo approximation of the posterior distribution.

Many other variants, very popular in fields like robotics, navigation, guidance, aerospace, finance, vision, etc.

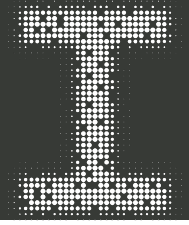


We can use the *unscented transform* to obtain a Gaussian approximation of the posterior distribution.



We can use the *unscented transform* to obtain a Gaussian approximation of the posterior distribution.

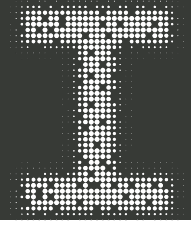
Let X be a Gaussian random variable $X \sim \mathcal{N}(x; \mu, \Sigma)$.



We can use the *unscented transform* to obtain a Gaussian approximation of the posterior distribution.

Let X be a Gaussian random variable $X \sim \mathcal{N}(x; \mu, \Sigma)$.

Say we would like to compute moments of $g(X)$ where g is a nonlinear function.



The unscented transform precisely approximates the moments of $g(X)$ by the moments of a Gaussian random variable.

The unscented transform precisely approximates the moments of $g(X)$ by the moments of a Gaussian random variable.

The unscented transform is based on the idea of *sigma points*, which are chosen deterministically:

$$\begin{aligned}
\sigma_0 &= \mu, \\
\sigma_i &= \mu + \left(\sqrt{(n+\lambda)\Sigma}\right)_i, \quad i = 1, \ldots, n, \\
\sigma_i &= \mu - \left(\sqrt{(n+\lambda)\Sigma}\right)_{i-n}, \quad i = n+1, \ldots, 2n,
\end{aligned}$$

where $\lambda = \alpha^2(n+\kappa) - n$ and α and κ are parameters that can be chosen freely.

How is this idea used given:

$$\begin{aligned}
\pi_0(x_0) &= \mathcal{N}(x_0; \mu_0, V_0), \\
\tau_t(x_t|x_{t-1}) &= \mathcal{N}(x_t; a_t(x_{t-1}), Q_t) \\
g_t(y_t|x_t) &= \mathcal{N}(y_t; h_t(x_t), R_t).
\end{aligned}$$

How is this idea used given:

$$\begin{aligned}\pi_0(x_0) &= \mathcal{N}(x_0; \mu_0, V_0),\\ \tau_t(x_t|x_{t-1}) &= \mathcal{N}(x_t; a_t(x_{t-1}), Q_t)\\ g_t(y_t|x_t) &= \mathcal{N}(y_t; h_t(x_t), R_t).\end{aligned}$$

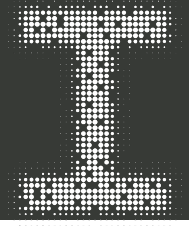
Given μ_{t-1}, V_{t-1}, use the nonlinearity $a_t(\cdot)$ using unscented transform to compute the moments of prediction.

How is this idea used given:

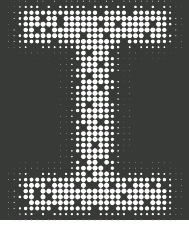
$$\begin{aligned}\pi_0(x_0) &= \mathcal{N}(x_0; \mu_0, V_0),\\ \tau_t(x_t|x_{t-1}) &= \mathcal{N}(x_t; a_t(x_{t-1}), Q_t)\\ g_t(y_t|x_t) &= \mathcal{N}(y_t; h_t(x_t), R_t).\end{aligned}$$

Given μ_{t-1}, V_{t-1}, use the nonlinearity $a_t(\cdot)$ using unscented transform to compute the moments of prediction.

Given μ_t, V_t, use the nonlinearity $h_t(\cdot)$ using unscented transform to compute the moments of updated posterior.

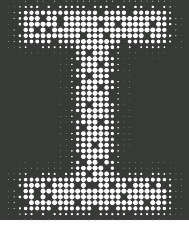


We have been looking at the filtering problem, i.e., estimating $\pi_t(x_t|y_{1:t})$.



We have been looking at the filtering problem, i.e., estimating $\pi_t(x_t|y_{1:t})$.

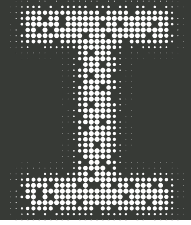
What if we want to estimate $\pi_t(x_t|y_{1:T})$ for $T > t$?



We have been looking at the filtering problem, i.e., estimating $\pi_t(x_t|y_{1:t})$.

What if we want to estimate $\pi_t(x_t|y_{1:T})$ for $T > t$?

This is called the *smoothing problem*. These methods are usually implemented backwards in time.



We have smoothing recursions

$$\pi(x_{t+1}|y_{1:t}) = \int \tau(x_{t+1}|x_t)\pi(x_t|y_{1:t})\mathrm{d}x_t,$$

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

Proof: Let us notice

$$\begin{aligned} p(x_t|x_{t+1}, y_{1:T}) &= p(x_t|x_{t+1}, y_{1:t}), \\ &= \frac{p(x_t, x_{t+1}|y_{1:t})}{p(x_{t+1}|y_{1:t})}, \\ &= \frac{\pi(x_t|y_{1:t})\tau(x_{t+1}|x_t)}{\pi(x_{t+1}|y_{1:t})}, \end{aligned}$$

where the last equality follows from the Markov property.

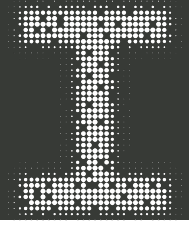
Proof: Let us notice

$$\begin{aligned} p(x_t|x_{t+1}, y_{1:T}) &= p(x_t|x_{t+1}, y_{1:t}), \\ &= \frac{p(x_t, x_{t+1}|y_{1:t})}{p(x_{t+1}|y_{1:t})}, \\ &= \frac{\pi(x_t|y_{1:t})\tau(x_{t+1}|x_t)}{\pi(x_{t+1}|y_{1:t})}, \end{aligned}$$

where the last equality follows from the Markov property. Now we construct the joint

$$\begin{aligned} p(x_{t+1}, x_t|y_{1:T}) &= p(x_t|x_{t+1}, y_{1:T})p(x_{t+1}|y_{1:T}), \\ &= \frac{\pi(x_t|y_{1:t})\tau(x_{t+1}|x_t)}{\pi(x_{t+1}|y_{1:t})}\pi(x_{t+1}|y_{1:T}). \end{aligned}$$

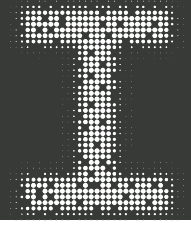
By integrating out x_{t+1}, the result follows.



Let us consider our linear-Gaussian model again

$$\begin{aligned}\pi_0(x) &= \mathcal{N}(x; \mu_0, V_0),\\ \tau_t(x_t|x_{t-1}) &= \mathcal{N}(x_t; A_t x_{t-1}, Q_t),\\ g_t(y_t|x_t) &= \mathcal{N}(y_t; H_t x_t, R_t).\end{aligned}$$

In this setting, smoothing can be exactly implemented too.

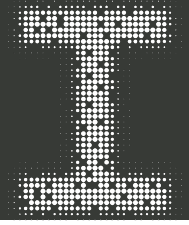


Let us consider our linear-Gaussian model again

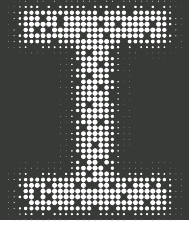
$$\begin{aligned}\pi_0(x) &= \mathcal{N}(x; \mu_0, V_0),\\ \tau_t(x_t|x_{t-1}) &= \mathcal{N}(x_t; A_t x_{t-1}, Q_t),\\ g_t(y_t|x_t) &= \mathcal{N}(y_t; H_t x_t, R_t).\end{aligned}$$

In this setting, smoothing can be exactly implemented too.

The resulting algorithm is called the *Rauch-Tung-Striebel* (RTS) smoother.



Assume we have computed filter moments $(\mu_t, V_t)_{t=0}^T$.

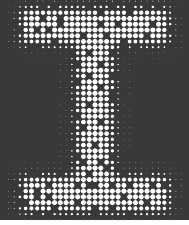


Assume we have computed filter moments $(\mu_t, V_t)_{t=0}^T$. The smoother is then given as

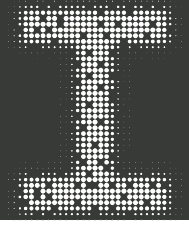
$$\begin{aligned}
\mu_T^s &= \mu_T, \\
V_T^s &= V_T, \\
\mu_t^s &= \mu_t + J_t(\mu_{t+1}^s - A_t\mu_t), \\
V_t^s &= V_t + J_t(V_{t+1}^s - V_t)J_t^\top,
\end{aligned}$$

where

$$J_t = V_t A_t^\top \hat{V}_{t+1}^{-1}.$$



Assume we have computed filter moments $(\mu_t, V_t)_{t=0}^T$.

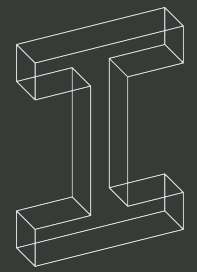


Assume we have computed filter moments $(\mu_t, V_t)_{t=0}^T$. The smoother is then given as

$$\begin{aligned}
\mu_T^s &= \mu_T, \\
V_T^s &= V_T, \\
\mu_t^s &= \mu_t + J_t(\mu_{t+1}^s - A_t\mu_t), \\
V_t^s &= V_t + J_t(V_{t+1}^s - V_t)J_t^\top,
\end{aligned}$$

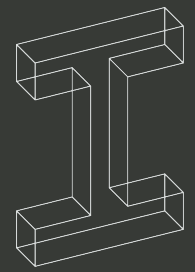
where

$$J_t = V_t A_t^\top \hat{V}_{t+1}^{-1}.$$



Deterministic approximations are only useful in certain settings where we can ensure

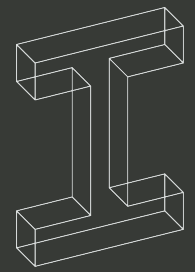
- Exact or approximate linearity
- Gaussianity



Deterministic approximations are only useful in certain settings where we can ensure

- Exact or approximate linearity
- Gaussianity

We will now introduce a general Monte Carlo approach to estimate posterior distributions $\pi_t^N(\mathrm{d}x_t|y_{1:t})$.



Deterministic approximations are only useful in certain settings where we can ensure

- Exact or approximate linearity
- Gaussianity

We will now introduce a general Monte Carlo approach to estimate posterior distributions $\pi_t^N(\mathrm{d}x_t|y_{1:t})$.

Particle filters.

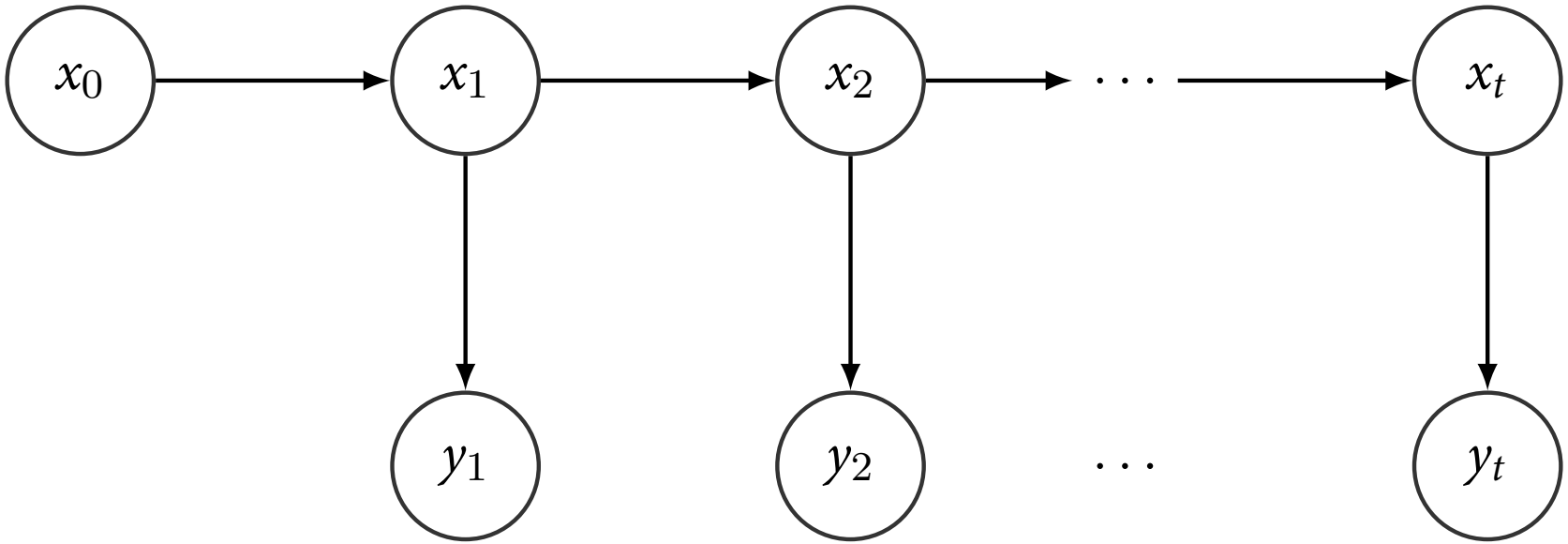
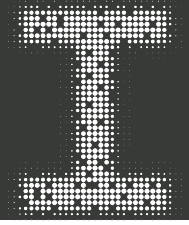


Figure: The conditional independence structure of a state-space model.

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

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

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



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

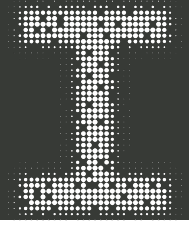
Before we go into the details of the derivation, let us directly look at the algorithm. A general algorithm to estimate expectations of any test function $\varphi(x_t)$ given $y_{1:t}$.

- 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,
$$x_t^{(i)} \sim \tilde{\pi}_t(\mathrm{d}x) := \sum_i w_t^{(i)} \delta_{\bar{x}_t^{(i)}}(\mathrm{d}x) \quad \text{for } i = 1, \ldots, N.$$

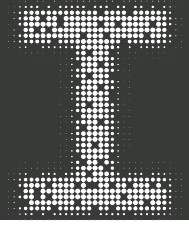
$$\pi_{t-1}^N \underbrace{\rightarrow}_{\text{sampling}} \xi_t^N \underbrace{\rightarrow}_{\text{weighting}} \tilde{\pi}_t^N \underbrace{\rightarrow}_{\text{resampling}} \pi_t^N.$$

Particle filters

Derivation

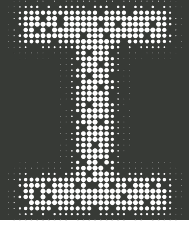


Where does the algorithm come from?



Where does the algorithm come from?

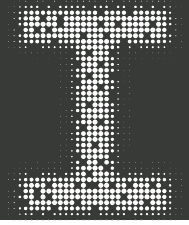
Surprisingly, we will not use the prediction-update recursions directly unlike in the Kalman filter.



Where does the algorithm come from?

Surprisingly, we will not use the prediction-update recursions directly unlike in the Kalman filter.

We will instead develop an importance sampler on the path space.



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

$$\begin{aligned}(\varphi, \pi) &= \int \varphi(x)\pi(x)\mathrm{d}x, \\ &= \frac{\int \varphi(x)\frac{\gamma(x)}{q(x)}q(x)\mathrm{d}x}{\int \frac{\gamma(x)}{q(x)}q(x)\mathrm{d}x}, \\ &\approx \frac{\frac{1}{N}\sum_{i=1}^{N}\varphi(x^{(i)})\frac{\gamma(x^{(i)})}{q(x^{(i)})}}{\frac{1}{N}\sum_{i=1}^{N}\frac{\gamma(x^{(i)})}{q(x^{(i)})}}, \qquad x^{(i)} \sim q, \quad i = 1, \ldots, N. \quad (10)\end{aligned}$$

Let us now introduce the unnormalised weight function

$$W(x) = \frac{\gamma(x)}{q(x)}. \tag{11}$$

With this, the Eq. (10) becomes

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

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

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

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

by introducing the *normalised importance weights*

$$\mathrm{w}^{(i)} = \frac{\mathrm{w}^{(i)}}{\sum_{i=1}^{N} \mathrm{w}^{(i)}}, \tag{12}$$

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

$$\pi^N(\mathrm{d}x) = \sum_{i=1}^{N} \mathrm{w}^{(i)} \delta_{x^{(i)}}(\mathrm{d}x). \tag{13}$$

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

$$\gamma(x_{0:t}, y_{1:t}) = \mu(x_0) \prod_{k=1}^{t} \tau(x_k|x_{k-1}) g(y_k|x_k). \tag{14}$$

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

$$\gamma(x_{0:t}, y_{1:t}) = \mu(x_0) \prod_{k=1}^{t} \tau(x_k|x_{k-1}) g(y_k|x_k). \tag{14}$$

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 $\mathrm{W}_{0:t}^{(i)} = W_{0:t}(x_{0:t}^{(i)})$ for $i = 1, \ldots, N$ and building a particle approximation

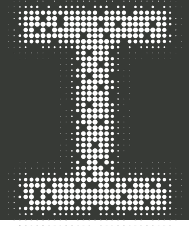
$$\pi^N(\mathrm{d}x_{0:t}) = \sum_{i=1}^{N} \mathrm{W}_{0:t}^{(i)} \delta_{x_{0:t}^{(i)}}(\mathrm{d}x_{0:t}).$$

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

$$\begin{aligned} W_{0:t}(x_{0:t}) &= \frac{\gamma(x_{0:t}, y_{1:t})}{q(x_{0:t})}, \\ &= \frac{\gamma(x_{0:t-1}, y_{1:t-1})}{q(x_{0:t-1})} \frac{\tau(x_t|x_{t-1})g(y_t|x_t)}{q(x_t|x_{0:t-1})}, \\ &= W_{0:t-1}(x_{0:t-1}) \frac{\tau(x_t|x_{t-1})g(y_t|x_t)}{q(x_t|x_{0:t-1})}, \\ &= W_{0:t-1}(x_{0:t-1}) W_t(x_{0:t}). \end{aligned} \tag{15}$$



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

$$\begin{aligned} W_{0:t}(x_{0:t}) &= \frac{\gamma(x_{0:t}, y_{1:t})}{q(x_{0:t})}, && (16)\\ &= \frac{\gamma(x_{0:t-1}, y_{1:t-1})}{q(x_{0:t-1})} \frac{\tau(x_t|x_{t-1})g(y_t|x_t)}{q(x_t|x_{t-1})}, && (17)\\ &= W_{0:t-1}(x_{0:t-1}) \frac{\tau(x_t|x_{t-1})g(y_t|x_t)}{q(x_t|x_{t-1})}, && (18)\\ &= W_{0:t-1}(x_{0:t-1}) W_t(x_t, x_{t-1}), && (19) \end{aligned}$$

Let us implement it. Assume that we have computed the unnormalised weights $\mathrm{W}_{0:t-1}^{(i)} = W(x_{0:t-1}^{(i)})$ 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 (19). And also note that $\mathrm{W}_{0:t-1}^{(i)}$ for $i = 1, \ldots, N$ are just numbers, they do not need 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

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

where

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

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

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

and then compute

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

and update

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

These are unnormalised weights and we normalise them to obtain,

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

which finally leads to the empirical measure,

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

Particle filters

Sequential Importance Sampling (SIS)

- 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:

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

and update

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

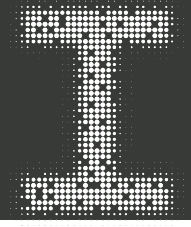
Normalise weights,

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

  - Report

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

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



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

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

Particle filters

Sequential Importance Sampling - Resampling (SISR)

- 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:

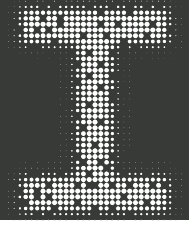
$$\mathrm{W}_t^{(i)} = \frac{\tau(\bar{x}_t^{(i)} | x_{t-1}^{(i)}) g(y_t | \bar{x}_t^{(i)})}{q(\bar{x}_t^{(i)} | x_{t-1}^{(i)})}.$$

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

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

  - Resample:

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



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

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

Particle filters

Bootstrap particle filter

- 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:

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

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

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

  - Resample:

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

References I