Feynman-Kac models & HMMs

nicolas.chopin@ensae.fr
(based on a previous PG course with O. Papaspiliopoulos)

Outline

Summary

- Tool: change of measure
- Define FK models via Markov and CoM
- FK formalism of given probabilistic models
- Explore properties of FK models: recursion, marginalisation, Markovianity
- Apply the machinery on specific SSMs: HMMs.

Change of measure

Definition

Let $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ be a measurable space, and $\mathbb{M}$ and $\mathbb{Q}$ two probability measures defined on this space. We then say that $\mathbb{Q}$ is absolutely continuous with respect to $\mathbb{M}$, if for any $A \in \mathcal{B}(\mathcal{X})$ for which $\mathbb{M}(A) = 0$, $\mathbb{Q}(A) = 0$. In this case, we also say that $\mathbb{M}$ dominates $\mathbb{Q}$.

In fact, $\mathbb{Q}$ is a.c. wrt to $\mathbb{M}$ iff $\exists$

$$w(x) = \frac{\mathbb{Q}(\mathrm{d}x)}{\mathbb{M}(\mathrm{d}x)}$$

(Radon-Nikodym)

Lemma

Suppose that $\mathbb{Q}$ and $\mathbb{M}$ are probability mesures on a space $\mathcal{X}$, and $w(x) \propto \mathbb{Q}(\mathrm{d}x)/\mathbb{M}(\mathrm{d}x)$. Then, for any test function ϕ,

$$\mathbb{M}(\phi w) = \mathbb{Q}(\phi)\mathbb{M}(w)\,.$$

The other way around

Alternatively, if you give me:

- A probability measure $\mathbb{M}$;
- a function G such that

$$L := \mathbb{M}(G) \in (0, \infty)$$

Then I can define:

$$\mathbb{Q}(\mathrm{d}x) = \frac{1}{L}\mathbb{M}(\mathrm{d}x)G(x)$$

The components of a Feynman-Kac model

- Markov measure:

$$\mathbb{M}_T(\mathrm{d}x_{0:T}) = \mathbb{M}_0(\mathrm{d}x_0) \prod_{t=1}^{T} M_t(x_{t-1}, \mathrm{d}x_t) \,.$$

- Potential functions, $G_0 : \mathcal{X} \to \mathbb{R}^+$, and $G_t : \mathcal{X}^2 \to \mathbb{R}^+$, for $1 \leq t \leq T$
- Change of measure: for $t \leq T$

$$\mathbb{Q}_t(\mathrm{d}x_{0:T}) = \frac{1}{L_t} G_0(x_0) \left\{ \prod_{s=1}^{t} G_s(x_{s-1}, x_s) \right\} \mathbb{M}_T(\mathrm{d}x_{0:T})$$

Components: $T, G_0, \mathbb{M}_0, \; G_t(x_{t-1}, x_t), M_t(x_{t-1}, \mathrm{d}x_t)$

Partition function/evidence/marginal likelihood

$$L_t = \int_{\mathcal{X}^{t+1}} G_0(x_0) \prod_{s=1}^{t} G_s(x_{s-1}, x_s) \, \mathbb{M}_t(\mathrm{d}x_{0:t})$$
$$= \mathbb{E}_{\mathbb{M}_t} \left[G_0(X_0) \prod_{s=1}^{t} G_s(X_{s-1}, X_s) \right] .$$

and assume that G_t's such that $0 < L_t < \infty$ for all t
Normalising factors: $\ell_t = L_t / L_{t-1}$

The "bootstrap" Feynman-Kac formalism of a state-space model

Consider a state-space model with signal transition kernels $P_t(x_{t-1}, \mathrm{d}x_t)$ and observation densities $f_t(y_t|x_t)$. We define its "bootstrap" Feynman-Kac formalism to be a Feynman-Kac model with the following components

$$\mathbb{M}_0(\mathrm{d}x_0) = \mathbb{P}_0(\mathrm{d}x_0), \quad G_0(x_0) = f_0(y_0|x_0)$$
$$M_t(x_{t-1}, \mathrm{d}x_t) = P_t(x_{t-1}, \mathrm{d}x_t), \quad G_t(x_{t-1}, x_t) = f_t(y_t|x_t).$$

Then

- $\mathbb{Q}_{t-1}(\mathrm{d}x_{0:t}) = \mathbb{P}_t(X_{0:t} \in \mathrm{d}x_{0:t} | Y_{0:t-1} = y_{0:t-1})$
- $\mathbb{Q}_t(\mathrm{d}x_{0:t}) = \mathbb{P}_t(X_{0:t} \in \mathrm{d}x_{0:t} | Y_{0:t} = y_{0:t})$

Is this the only one? And what is this formalism useful for?

The "guided" Feynman-Kac formalism of a state-space model

Consider a state-space model with signal transition kernels $P_t(x_{t-1}, \mathrm{d}x_t)$ and observation densities $f_t(y_t|x_t)$. We define its "guided" Feynman-Kac formalism to be a Feynman-Kac model with the following components

$$G_0(x_0)\mathbb{M}_0(\mathrm{d}x_0) = f_0(y_0|x_0)\mathbb{P}_0(\mathrm{d}x_0)\,,$$
$$G_t(x_{t-1}, x_t)M_t(x_{t-1}, \mathrm{d}x_t) = f_t(y_t|x_t)P_t(x_{t-1}, \mathrm{d}x_t)\,.$$

meaning of equalities, special case

The "auxiliary" Feynman-Kac formalism of a state-space model

Consider a state-space model with signal transition kernels $P_t(x_{t-1}, \mathrm{d}x_t)$ and observation densities $f_t(y_t|x_t)$. Additionally, let $\eta_t(x_t)$ be user-chosen, "auxiliary" functions, such that $\mathbb{E}_{\mathbb{P}_t}[\eta_t(X_t)|Y_{0:t} = y_{0:t}] < \infty$ for all t. We define its "auxiliary" Feynman-Kac formalism to be a Feynman-Kac model with the following components

$$G_0(x_0)\mathbb{M}_0(\mathrm{d}x_0) = f_0(y_0|x_0)\mathbb{P}_0(\mathrm{d}x_0)\eta_0(x_0)$$

$$G_t(x_{t-1}, x_t)M_t(x_{t-1}, \mathrm{d}x_t) = f_t(y_t|x_t)P_t(x_{t-1}, \mathrm{d}x_t)\frac{\eta_t(x_t)}{\eta_{t-1}(x_{t-1})}$$

Use of formalism

- Decouple a statistical model (the state-space model) from its mathematical representation $\rightarrow$ unified treatment of theory (recursions) and numerics (particle filters)
- Feynman-Kac models share the same fundamental structure: the specific change of measure from a Markov measure $\rightarrow$ common set of recursions regardless of the details of components
- Feynman-Kac representation and modularity
- Feynman-Kac outside state-space models

Forward recursion (Feynman-Kac formalism) pt1

Initialise with $\mathbb{Q}_{-1}(\mathrm{d}x_0) = \mathbb{M}_0(\mathrm{d}x_0)$, then, for $t = 0 : T$,

- Extension:

$$\mathbb{Q}_{t-1}(\mathrm{d}x_{t-1:t}) = \mathbb{Q}_{t-1}(\mathrm{d}x_{t-1}) M_t(x_{t-1}, \mathrm{d}x_t)$$

  Recall ▸ definition

- Change of measure:

$$\mathbb{Q}_t(\mathrm{d}x_{t-1:t}) = \frac{1}{\ell_t} G_t(x_{t-1}, x_t) \mathbb{Q}_{t-1}(\mathrm{d}x_{t-1:t})$$

Forward recursion (Feynman-Kac formalism) pt2

with

$$\ell_0 = L_0 = \int_{\mathcal{X}} G_0(x_0) M_0(\mathrm{d}x_0)$$

and

$$\ell_t = \frac{L_t}{L_{t-1}} = \int_{\mathcal{X}^2} G_t(x_{t-1}, x_t) \mathbb{Q}_{t-1}(\mathrm{d}x_{t-1:t})$$

for $t \geq 1$.

- Marginalisation:

$$\begin{aligned} \mathbb{Q}_t(\mathrm{d}x_t) &= \int_{\mathcal{X}} \mathbb{Q}_t(\mathrm{d}x_{t-1:t}) \\ &= \frac{1}{\ell_t} \int G_t(x_{t-1}, x_t) M_t(x_{t-1}, \mathrm{d}x_t) \mathbb{Q}_{t-1}(\mathrm{d}x_{t-1}) \end{aligned}$$

Implications for the "b"-fm: recursion for filter, prediction, likelihood

$$\mathbb{P}_{t-1}(X_t \in \mathrm{d}x_t | Y_{0:t-1} = y_{0:t-1})$$
$$= \int_{\mathcal{X}} P_t(x_{t-1}, \mathrm{d}x_t) \mathbb{P}_t(X_{t-1} \in \mathrm{d}x_{t-1} | Y_{0:t-1} = y_{0:t-1}),$$

$$\mathbb{P}_t(X_t \in \mathrm{d}x_t | Y_{0:t} = y_{0:t}) = \frac{1}{p_t(y_t | y_{0:t-1})} f_t(y_t | x_t)$$
$$\mathbb{P}_{t-1}(X_t \in \mathrm{d}x_t | Y_{0:t-1} = y_{0:t-1}).$$

$$p_t(y_t | y_{0:t-1}) = \int_{\mathcal{X}^2} f_t(y_t | x_t) \mathbb{P}_{t-1}(X_{t-1:t} \in \mathrm{d}x_{t-1:t} | Y_{0:t-1} = y_{0:t-1}).$$

Feynman-Kac model as a Markov measure - cost-to-go functions

$$H_{T:T}(x_T) = 1\,,$$

$$H_{t:T}(x_t) = \int_{\mathcal{X}^{T-t}} \prod_{s=t+1}^{T} G_s(x_{s-1}, x_s) M_s(x_{s-1}, \mathrm{d}x_s)\,, \quad t < T\,.$$

Hence

$$H_{t:T}(x_t) = \int_{\mathcal{X}} G_{t+1}(x_t, x_{t+1}) H_{t+1:T}(x_{t+1}) M_{t+1}(x_t, \mathrm{d}x_{t+1})$$

but also

$$H_{t:T}(x_t) = \mathbb{E}_{\mathbb{M}_T}\left[\prod_{s=t+1}^{T} G_s(X_{s-1}, X_s) \middle| X_t = x_t\right]$$

$$= \mathbb{E}_{\mathbb{M}_T}\left[G_{t+1}(X_t, X_{t+1}) H_{t+1:T}(X_{t+1}) \middle| X_t = x_t\right]$$

Proposition

$\mathbb{Q}_T$ is the law of a Markov process with state-space $\mathcal{X}$, initial distribution

$$Q_{0|T}(\mathrm{d}x_0) = \frac{H_{0:T}(x_0)}{L_T} G_0(x_0) \mathbb{M}_0(\mathrm{d}x_0) ,$$

forward transition kernels $Q_{t|T}(x_{t-1}, \mathrm{d}x_t)$ given by:

$$Q_{t|T}(x_{t-1}, \mathrm{d}x_t) = \frac{H_{t:T}(x_t)}{H_{t-1:T}(x_{t-1})} G_t(x_{t-1}, x_t) M_t(x_{t-1}, \mathrm{d}x_t) ,$$

and backward kernels given by:

$$\overleftarrow{Q}_{t-1|T}(x_t, \mathrm{d}x_{t-1}) = \frac{Q_{t|T}(x_{t-1}, \mathrm{d}x_t)}{\mathbb{Q}_T(\mathrm{d}x_t)} \mathbb{Q}_T(\mathrm{d}x_{t-1}) .$$

Implications for the "b"-fm: POMP

By immediate translation:

$$H_{t:T}(x_t) = \frac{\mathbb{P}_T(Y_{t+1:T} \in \mathrm{d}y_{t+1:T} | X_t = x_t)}{\nu^{T-t}(\mathrm{d}y_{t+1:T})}, t < T.$$

$$p_t(y_{t+1:T}|x_t) = \int_{\mathcal{X}} f(y_{t+1}|x_{t+1}) p(y_{t+2:T}|x_{t+1}) P_t(x_t, \mathrm{d}x_{t+1}).$$

Hence, the conditioned Markov process is also Markov with

$$P_{0|T}(\mathrm{d}x_0) = \frac{p(y_{1:T}|x_0)}{p(y_{0:T})} f_0(y_0|x_0) \mathbb{P}_0(\mathrm{d}x_0),$$

$$P_{t|T}(x_{t-1}, \mathrm{d}x_t) = \frac{p(y_{t+1:T}|x_t)}{p(y_{t:T}|x_{t-1})} f_t(y_t|x_t) P_t(x_{t-1}, \mathrm{d}x_t).$$

Stability properties

Forward-backward recursions in Feynman-Kac models

Recall that $\mathbb{M}_T(\mathrm{d}x_{0:t}) = \mathbb{M}_t(\mathrm{d}x_{0:t})$. For the Feynman-Kac model we have:

Proposition

For any $t < T$,

$$\mathbb{Q}_T(\mathrm{d}x_{0:t}) = \frac{L_t}{L_T} H_{t:T}(x_t) \mathbb{Q}_t(\mathrm{d}x_{0:t}) \,.$$

Ideas for proof?

Proof: use the Markov property of the Q process

Corollary

$$\mathbb{Q}_T(\mathrm{d}x_t) = \frac{L_t}{L_T} H_{t:T}(x_t) \mathbb{Q}_t(\mathrm{d}x_t) \,.$$

& from ▸ Proposition .3 and the result above we get:

Corollary

$$\overleftarrow{Q}_{t-1|T}(x_t, \mathrm{d}x_{t-1}) = \frac{1}{\ell_t} G_t(x_{t-1}, x_t) \frac{M_t(x_{t-1}, \mathrm{d}x_t)}{\mathbb{Q}_t(\mathrm{d}x_t)} \mathbb{Q}_{t-1}(\mathrm{d}x_{t-1})$$

Implications for the "b"-fm: forward filtering/backward smoothing

$$\mathbb{P}(X_t \in \mathrm{d}x_t | Y_{0:T} = y_{0:T}) = \frac{1}{p(y_{t+1:T}|y_{0:t})} p(y_{t+1:T}|x_t) \mathbb{P}(X_t \in \mathrm{d}x_t | Y_{0:t} = y_{0:t})$$

$$\overleftarrow{P}_{t-1|T}(x_t, \mathrm{d}x_{t-1}) = \frac{1}{p(y_t|y_{0:t-1})} f_t(y_t|x_t) \frac{P_t(x_{t-1}, \mathrm{d}x_t)}{\mathbb{P}_t(X_t \in \mathrm{d}x_t | Y_{0:t} = y_{0:t})} \mathbb{P}_{t-1}(X_{t-1} \in \mathrm{d}x_{t-1} | Y_{0:t-1} = y_{0:t-1}) .$$

Forward-backward simulation

How generate draws from $\mathbb{Q}_T(\mathrm{d}x_{0:T})$?

Then, we should know how to generate from

$$\mathbb{P}_T(X_{0:T} \in \mathrm{d}x_{0:T} | Y_{0:T} = y_{0:T})$$

ideas?

Further reading

- The "Don Quixote" of SMC: *Feynman-Kac formulae* (Del Moral, Springer)

Outline

HMMs

- $\mathcal{X} = \{1, \ldots, K\}$
- Integrals $\rightarrow$ sums; measures $\rightarrow$ vectors; kernels $\rightarrow$ matrices
- Following based on "bootstrap" formalism

Recursions for hidden Markov models

The following are understood for all $k \in 1 : \mathcal{D}$

- **Prediction:**

$$\mathbb{P}_{t-1}(X_t = k | Y_{0:t-1} = y_{0:t-1}) = \sum_{l=1}^{\mathcal{D}} \mathbb{P}_{t-1}(X_{t-1} = l | Y_{0:t-1} = y_{0:t-1}) p_t(k|l)$$

- **Filter:**

$$\mathbb{P}_t(X_t = k | Y_{0:t} = y_{0:t}) = \frac{1}{p(y_t | y_{0:t-1})} \mathbb{P}_{t-1}(X_t = k | Y_{0:t-1} = y_{0:t-1}) f_t(y_t|k)$$

- **Likelihood factors:**

$$p_t(y_t | y_{0:t-1}) = \sum_k \mathbb{P}_{t-1}(X_t = k | Y_{0:t-1} = y_{0:t-1}) f_t(y_t|k)$$

- **Likelihood of future observations given current state:**

$$p_T(y_{t+1:T}|k) = \sum_l f(y_{t+1}|l) p(y_{t+2:T}|l) p_t(l|k)$$

Complexity

- Predictive probabilities: $O(K^2)$ (unless sparse transition matrix, e.g. change point models)
- Given those, filter & likelihood factors obtained at $O(K)$
- Overall cost: $O(TK^2)$ as opposed to $O(K^T)$
- Still, K might be large...

Particle filtering

nicolas.chopin@ensae.fr

(based on a previous PG course with O. Papaspiliopoulos)

Objectives

Objectives

- introduce a generic PF algorithm for a given Feynman-Kac model $\{(M_t, G_t)\}_{t=0}^{T}$
- discuss the different algorithms one may obtain for a given state-space model, by using different Feynman-Kac formalisms.
- give more details on the implementation, complexity, and so on of the algorithm.

The algorithm

Input

- A Feynman-Kac model $\{(M_t, G_t)\}_{t=0}^{T}$ such that:
  - the weight function G_t may be evaluated pointwise (for all t);
  - it is possible to simulate from $M_0(\mathrm{d}x_0)$ and from $M_t(x_{t-1}, \mathrm{d}x_t)$ (for any x_{t-1} and t)
- The number of particles N

Algorithm 1 Basic PF algorithm

All operations to be performed for all $n \in 1:N$.
At time 0:

(a) Generate $X_0^n \sim M_0(\mathrm{d}x_0)$.

(b) Compute $w_0^n = G_0(X_0^n)$, $W_0^n = w_0^n / \sum_{m=1}^N w_0^m$, and $L_0^N = N^{-1} \sum_{n=1}^N w_0^n$.

Recursively, for $t = 1, \ldots, T$:

(a) Generate ancestor variables $A_t^n \in 1:N$ independently from $\mathcal{M}(W_{t-1}^{1:N})$.

(b) Generate $X_t^n \sim M_t(X_{t-1}^{A_t^n}, \mathrm{d}x_t)$.

(c) Compute $w_t^n = G_t(X_{t-1}^{A_t^n}, X_t^n)$, $W_t^n = w_t^n / \sum_{m=1}^N w_t^m$, and $L_t^N = L_{t-1}^N \{N^{-1} \sum_{n=1}^N w_t^n\}$.

Output

the algorithm delivers the following approximations at each time t:

$$\frac{1}{N}\sum_{n=1}^{N}\delta_{X_t^n} \quad \text{approximates } \mathbb{Q}_{t-1}(\mathrm{d}x_t)$$

$$\mathbb{Q}_t^N(\mathrm{d}x_t) = \sum_{n=1}^{N} W_t^n \delta_{X_t^n} \quad \text{approximates } \mathbb{Q}_t(\mathrm{d}x_t)$$

$$L_t^N \quad \text{approximates } L_t$$

some comments

- by *approximates*, we mean: for any test function φ, the quantity

$$\mathbb{Q}_t^N(\varphi) = \sum_{n=1}^{N} W_t^n \varphi(X_t^n)$$

converges to $\mathbb{Q}_t(\varphi)$ as $N \to +\infty$ (at the standard Monte Carlo rate $O_P(N^{-1/2})$).

some comments

- by *approximates*, we mean: for any test function φ, the quantity

$$\mathbb{Q}_t^N(\varphi) = \sum_{n=1}^{N} W_t^n \varphi(X_t^n)$$

converges to $\mathbb{Q}_t(\varphi)$ as $N \to +\infty$ (at the standard Monte Carlo rate $O_P(N^{-1/2})$).

- complexity is $O(N)$ per time step.

Particle algorithms for a given state-space model

Principle

We now consider a given state-space model:

- with initial law $P_0(\mathrm{d}x_0)$ and Markov kernel $P_t(x_{t-1}, \mathrm{d}x_t)$ for $\{X_t\}$;
- with conditional probability density $f_t(y_t|x_t)$ for $Y_t|X_t$

and discuss how the choice of a particular Feynman-Kac formalism leads to more or less efficient particle algorithms.

The bootstrap filter

Bootstrap Feynman-Kac formalism:

$$M_t(x_{t-1}, \mathrm{d}x_t) = P_t(x_{t-1}, \mathrm{d}x_t), \quad G_t(x_{t-1}, x_t) = f_t(y_t|x_t)$$

then $\mathbb{Q}_t$ is the filtering distribution, L_t is the likelihood of $y_{0:t}$, and so on.

The resulting algorithm is called the **boostrap filter**, and is particularly simple to interpret: we sample particles from Markov transition $P_t(x_{t-1}, \mathrm{d}x_t)$, and we reweight particles according to how compatible they are with the data.

The boostrap filter: algorithm

All operations to be performed for all $n \in 1:N$.

At time 0:

(a) Generate $X_0^n \sim P_0(\mathrm{d}x_0)$.

(b) Compute $w_0^n = f_0(y_0|X_0^n)$, $W_0^n = w_0^n / \sum_{m=1}^N w_0^m$, and $L_0^N = N^{-1} \sum_{n=1}^N w_0^n$.

Recursively, for $t = 1, \ldots, T$:

(a) Generate ancestor variables $A_t^n \in 1:N$ independently from $\mathcal{M}(W_{t-1}^{1:N})$.

(b) Generate $X_t^n \sim P_t(X_{t-1}^{A_t^n}, \mathrm{d}x_t)$.

(c) Compute $w_t^n = f_t(y_t|X_t^n)$, $W_t^n = w_t^n / \sum_{m=1}^N w_t^m$, and $L_t^N = L_{t-1}^N \{N^{-1} \sum_{n=1}^N w_t^n\}$.

The bootstrap filter: output

$$\frac{1}{N}\sum_{n=1}^{N}\varphi(X_t^n) \quad \text{approximates } \mathbb{E}[\varphi(X_t)|Y_{0:t-1}=y_{0:t-1}]$$

$$\sum_{n=1}^{N}W_t^n\varphi(X_t^n) \quad \text{approximates } \mathbb{E}[\varphi(X_t)|Y_{0:t}=y_{0:t}]$$

$$L_t^N \quad \text{approximates } p(y_{0:t})$$

The bootstrap filter: pros and cons

Pros:

- particularly simple
- does not require to compute the density $X_t|X_{t-1}$: we can apply it to models with **intractable dynamics**

Cons:

- We simulate particles *blindly*: if $Y_t|X_t$ is very informative, few particles will get a non-negligible weight.

The guided PF

Guided Feynman-Kac formalism: M_t is a user-chosen **proposal** kernel such that $M_t(x_{t-1}, \mathrm{d}x_t)$ dominates $P_t(x_{t-1}, \mathrm{d}x_t)$, and

$$G_t(x_{t-1}, x_t) = \frac{f_t(y_t|x_t) P_t(x_{t-1}, \mathrm{d}x_t)}{M_t(x_{t-1}, \mathrm{d}x_t)}$$
$$= f_t(y_t|x_t) \frac{p_t(x_t|x_{t-1})}{m_t(x_t|x_{t-1})}$$

(assuming in the second line that both kernels admit a density wrt a common measure). We still have that $\mathbb{Q}_t(\mathrm{d}x_t)$ is the filtering distribution, and L_t is the likelihood.

We call the resulting algorithm the **guided particle filter**, as in practice we would like to choose M_t so as to **guide** particles to regions of high likelihood.

The guided PF: choice of M_t (local optimality)

Suppose that (G_s, M_s) have been chosen to satisfy (**??**) for $s \leq t-1$. Among all pairs (M_t, G_t) that satisfy (**??**), the Markov kernel

$$M_t^{\mathrm{opt}}(x_{t-1}, \mathrm{d}x_t) = \frac{f_t(y_t|x_t)}{\int_{\mathcal{X}} f(y_t|x')\, P_t(x_{t-1}, \mathrm{d}x')} P_t(x_{t-1}, \mathrm{d}x_t)$$

minimises the variance of the weights, $\mathrm{Var}\left[G_t(X_{t-1}^{A_t^n}, X_t^n)\right]$.

Interpretation and discussion of this result

- M_t^{opt} is simply the law of X_t given X_{t-1} and Y_t. In a sense it is the perfect compromise between the information brought by $P_t(x_{t-1}, \mathrm{d}x_t)$ and by $f_t(y_t|x_t)$.
- In most practical cases, M_t^{opt} is not tractable, hence this result is mostly indicative (on how to choose M_t).
- Note also that the local optimality criterion is debatable. For instance, we do not consider the effect of *future* datapoints.

A first example: stochastic volatility

There, the log-density of $X_t|X_{t-1}, Y_t$ is (up to a constant):

$$-\frac{1}{2\sigma^2}\left\{x_t - \mu - \phi(x_{t-1} - \mu)\right\}^2 - \frac{x_t}{2} - \frac{e^{-x_t}}{2}y_t^2$$

We can use $e^{x-x_0} \approx 1 + (x - x_0) + (x - x_0)^2/2$ to get a Gaussian approximation.

A second example: bearings-only tracking

In that case, $P_t(x_{t-1}, \mathrm{d}x_t)$ imposes deterministic constraints:

$$X_t(k) = X_{t-1}(k) + X_{t-1}(k+2), \quad k = 1, 2$$

We can choose a M_t that imposes the same constraints. However, in this case, we find that

$$M_t^{\mathrm{opt}}(x_{t-1}, \mathrm{d}x_t) = P_t(x_{t-1}, \mathrm{d}x_t).$$

Discuss.

Guided particle filter pros and cons

Pro:

- may work much better that bootstrap filter when $Y_t|X_t$ is informative (provided we are able to derive a good proposal).

Cons:

- requires to be able to compute density $p_t(x_t|x_{t-1})$.
- sometimes local optimality criterion is not so sound.

The auxiliary particle filter

In the auxiliary Feynman-Kac formalism, an extra degree of freedom is gained by introducing **auxiliary** function η_t, and set:

$$G_0(x_0) = f_0(y_0|x_0)\frac{P_0(\mathrm{d}x_0)}{M_0(\mathrm{d}x_0)}\eta_0(x_0),$$

$$G_t(x_{t-1}, x_t) = f_t(y_t|x_t)\frac{P_t(x_{t-1}, \mathrm{d}x_t)}{\mathbb{M}_t(x_{t-1}, \mathrm{d}x_t)}\frac{\eta_t(x_t)}{\eta_{t-1}(x_{t-1})}.$$

so that

$$\mathbb{Q}_t(\mathrm{d}x_{0:t}) \propto \mathbb{P}(\mathrm{d}x_{0:t}|Y_{0:t} = y_{0:t})\eta_t(x_t)$$

and we recover the filtering distribution by reweighting by $1/\eta_t$.

Idea: choose η_t so that G_t is as constant as possible.

Output of APF

Let $\tilde{w}_t^n := w_t^n / \eta_t(X_t^n)$, $\tilde{W}_t^n := \tilde{w}_t^n / \sum_{m=1}^N \tilde{w}_t^m$, then

$$\frac{1}{\sum_{m=1}^N \frac{\tilde{W}_t^m}{f(y_t|X_t^m)}} \sum_{n=1}^N \frac{\tilde{W}_t^n}{f_t(y_t|X_t^n)} \varphi(X_t^n) \quad \text{approx.} \quad \mathbb{E}[\varphi(X_t)|Y_{0:t-1} = y_{0:t-1}]$$

$$\sum_{n=1}^N \tilde{W}_t^n \varphi(X_t^n) \quad \text{approx.} \quad \mathbb{E}[\varphi(X_t)|Y_{0:t} = y_{0:t}]$$

$$L_t^N \times N^{-1} \sum_{n=1}^N \tilde{w}_t^n \quad \text{approx.} \quad p(y_{0:t})$$

Local optimality for M_t and η_t

For a given state-space model, suppose that (G_s, M_s) have been chosen to satisfy (**??**) for $s \leq t-2$, and M_{t-1} has also been chosen. Among all pairs (M_t, G_t) that satisfy (**??**) and functions η_{t-1}, the Markov kernel

$$M_t^{\mathrm{opt}}(x_{t-1}, \mathrm{d}x_t) = \frac{f_t(y_t|x_t)}{\int_{\mathcal{X}} f(y_t|x')\, P_t(x_{t-1}, \mathrm{d}x')} P_t(x_{t-1}, \mathrm{d}x_t)$$

and the function

$$\eta_{t-1}^{\mathrm{opt}}(x_{t-1}) = \int_{\mathcal{X}} f(y_t|x')\, P_t(x_{t-1}, \mathrm{d}x')$$

minimise $\mathrm{Var}\left[G_t(X_{t-1}^{A_t^n}, X_t^n)/\eta_t(X_t^n)\right]$.

Interpretation and discussion

- We find again that the optimal proposal is the law of X_t given X_{t-1} and Y_t. In addition, the optimal auxiliary function is the probability density of Y_t given X_{t-1}.
- For this ideal algorithm, we would have

$$G_t(x_{t-1}, x_t) = \eta_t^{\mathrm{opt}}(x_t);$$

the density of Y_{t+1} given $X_t = x_t$; not constant, but intuitively less variable than $f_t(y_t|x_t)$ (as in the bootstrap filter).

Example: stochastic volatility

We use the same ideas as for the guided PF: Taylor expansion of log-density, then we integrate wrt x_t.

APF pros and cons

Pros:

- usually gives some extra performance.

Cons:

- a bit difficult to interpret and use;
- they are some (contrived) examples where the auxiliary particle filter actually performs worse than the bootstrap filter.

Note on the generality of APF

From the previous descriptions, we see that:

- the guided PF is a particular instance of the auxiliary particle filter (take $\eta_t = 1$);
- the bootstrap filter is a particular instance of the guided PF(take $M_t = P_t$).

This is why some recent papers focus on the APF.

When to resample?

Resampling or not resampling, that is the question

For the moment, we resample every time. When we introduced resampling, we explained that the decision to resample was based on a trade-off: adding noise at time $t-1$, while hopefully reducing noise at time t (assuming that $\{X_t\}$ forgets its past).

Resampling or not resampling, that is the question

For the moment, we resample every time. When we introduced resampling, we explained that the decision to resample was based on a trade-off: adding noise at time $t-1$, while hopefully reducing noise at time t (assuming that $\{X_t\}$ forgets its past).

We do know that never resample would be a bad idea: consider $M_t(x_{t-1}, \mathrm{d}x_t)$ defined such that the X_t are IID $\mathcal{N}(0,1)$, $G_t(x_t) = \mathbb{1}(x_t > 0)$. (More generally, recall the curse of dimensionality of importance sampling.)

The ESS recipe

Trigger the resampling step whenever the variability of the weights is too large, as measured by e.g. the ESS (effective sample size):

$$\mathrm{ESS}(W_t^{1:N}) := \frac{1}{\sum_{n=1}^N (W_t^n)^2} = \frac{\{\sum_{n=1}^N w_t(X^n)\}^2}{\sum_{n=1}^N w_t(X^n)^2}.$$

Recall that $\mathrm{ESS}(W_t^{1:N}) \in [1, N]$, and that if k weights equal one, and $N - k$ weights equal zero, then $\mathrm{ESS}(W_t^{1:N}) = k$.

PF with adaptive resampling

(Same operations at $t = 0$.)

Recursively, for $t = 1, \ldots, T$:

(a) **If** $\mathrm{ESS}(W_{t-1}^{1:N}) < \gamma N$
generate ancestor variables $A_{t-1}^{1:N}$ from resampling distribution $\mathcal{RS}(W_{t-1}^{1:N})$, and set $\hat{W}_{t-1}^n = W_{t-1}^{A_t^n}$;
Else (no resampling)
set $A_{t-1}^n = n$ and $\hat{W}_{t-1}^n = 1/N$

(b) Generate $X_t^n \sim M_t(X_{t-1}^{A_t^n}, \mathrm{d}x_t)$.

(c) Compute $w_t^n = (N\hat{W}_{t-1}^n) \times G_t(X_{t-1}^{A_t^n}, X_t^n)$,
$L_t^N = L_{t-1}^N \{N^{-1} \sum_{n=1}^N w_t^n\}$, $W_t^n = w_t^n / \sum_{m=1}^N w_t^m$.