Sequential Monte Carlo

Xiaohan Fei

Recursive Weight Update

We are interested in the posterior distribution $p(x_{1:t}|y_{1:t})$, apply Bayes' rule, we get:

$$p(x_{1:t}|y_{1:t}) = p(x_{1:t-1}|y_{1:t-1})\frac{p(y_t|x_{1:t}, y_{1:t-1})p(x_t|x_{1:t-1}, y_{1:t-1})}{p(y_t|y_{1:t-1})}.$$

Assuming the following conditional independences $x_t \perp x_{1:t-2}, y_{1:t-1}|x_{t-1}$ and $y_t|x_{1:t-1}, y_{t-1}|x_t$, we have:

$$p(x_{1:t}|y_{1:t}) \propto p(x_{1:t-1}|y_{1:t-1})p(y_t|x_t)p(x_t|x_{t-1}).$$

The weights read as:

$$w_t \propto p(x_{1:t-1}|y_{1:t-1})\frac{p(y_t|x_t)p(x_t|x_{t-1})}{q(x_{1:t})}.$$

Now assume the posterior $p(x_{1:t-1}|y_{1:t-1})$ is represented as a set of weighted samples $\left\{x_{t-1}^{(i)}, w_{t-1}^{(i)}\right\}_{i=1}^{N}$ such that

$$p(x_{1:t-1}|y_{1:t-1}) = \sum_{i=1}^{N} w_{t-1}^{(i)}\delta\left(x_{1:t-1} - x_{1:t-1}^{(i)}\right).$$

Plugin this into $p(x_{1:t}|y_{1:t})$ we got:

$$p(x_{1:t}|y_{1:t}) \propto \sum_{i=1}^{N} w_{t-1}^{(i)}\delta\left(x_{1:t-1} - x_{1:t-1}^{(i)}\right)p(y_t|x_t)p(x_t|x_{t-1})$$

And corresponding weights:

$$w_t \propto \sum_{i=1}^{N} w_{t-1}^{(i)}\delta\left(x_{1:t-1} - x_{1:t-1}^{(i)}\right)\frac{p(y_t|x_t)p(x_t|x_{t-1})}{q(x_{1:t})}.$$

Ideally, we should sample the whole trajectory $x_{1:t}$ from proposal distribution, however that's intractable due to the high dimension of trajectory. In practice, new trajectories are usually constructed by augmenting old trajectories one step before: $x_{1:t}^{(i)} = \left(x_{1:t-1}^{(i)}, x_t\right)$.

In other words, we first sample segments $x_{1:t-1}$ from distribution formed by normalized weights $\{v_{t-1}^{(i)}\}_{i=1}^{N}$ and then sample from conditional distribution $q(x_t|x_{1:t-1})$. The proposal distribution reads as:

$$q(x_{1:t}) = q(x_t|x_{1:t-1})\sum_{i=1}^{N} v_{t-1}^{(i)}\,\delta\left(x_{1:t-1} - x_{1:t-1}^{(i)}\right).$$

Plugin this term into the expression of weights:

$$w_t \propto \frac{\sum_{i=1}^{N} w_{t-1}^{(i)} \delta\left(x_{1:t-1} - x_{1:t-1}^{(i)}\right)}{\sum_{i=1}^{N} v_{t-1}^{(i)} \delta\left(x_{1:t-1} - x_{1:t-1}^{(i)}\right)} \frac{p(y_t|x_t)p(x_t|x_{t-1})}{q(x_t|x_{1:t-1})} = \frac{\mathrm{w}_{t-1}}{v_{t-1}} \frac{p(y_t|x_t)p(x_t|x_{t-1})}{q(x_t|x_{1:t-1})}$$

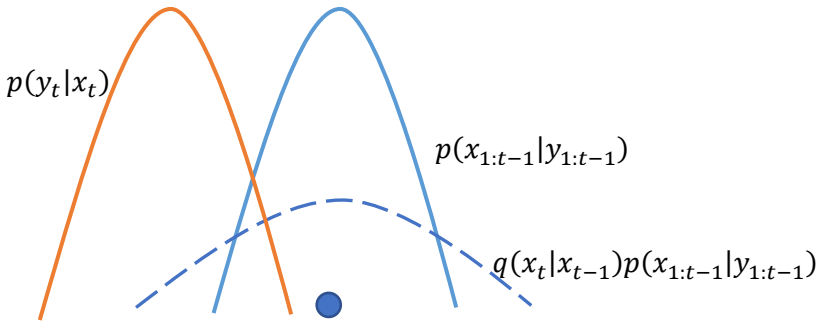
There are different ways to choose $\{v_{t-1}^{(i)}\}_{i=1}^{N}$. Here we discuss two popular choices:

1. **Sample from prior.** One can choose $v_{t-1}^{(i)} = w_{t-1}^{(i)}$, which is essentially sampling from prior distribution $p(x_{1:t-1}|y_{1:t-1})$ (this is posterior with data up to time t-1, but prior up to time t). In this case the first term in weight w_t cancels out, leaving us:

$$w_t \propto \frac{p(y_t|x_t)p(x_t|x_{t-1})}{q(x_t|x_{1:t-1})}.$$

which is equivalent to Bootstrap filter where resampling is performed immediately after weight update so that particles have uniform weights $\{w_{t-1}^{(i)} = 1\}_{i=1}^{N}$ and then in next iteration each particle is treated equally $\{v_{t-1}^{(i)} = 1\}_{i=1}^{N}$. If no resampling is performed, one has non-constant weights $\{w_{t-1}^{(i)} = 1\}_{i=1}^{N}$after previous iteration and needs to sample from it. Typically, once picks natural transition $q(x_t|x_{1:t-1}) = p(x_t|x_{t-1})$ and then $w_t \propto p(y_t|x_t)$.

2. **Sample from posterior.** Another choice is to sample from the posterior $v_{t-1}^{(i)} \propto p\left(x_{1:t-1}^{(i)}\middle|y_{1:t}\right)$, which is intractable (explained later) because of an inevitable integral in its computation. However, this scheme should perform better than blindly sampling from prior distribution $p(x_{1:t-1}|y_{1:t-1})$, since intuitively the posterior is informed by the new data. Also, high density region in prior distribution $p(x_{1:t-1}|y_{1:t-1})$ does not guarantee high density in $p(x_{1:t}|y_{1:t})$, since the likelihood term $p(y_t|x_t)$ might not heavily overlap with $p(x_{1:t-1}|y_{1:t-1})$ and samples from posterior then fall into tail of posterior.



Prior distribution (blue) and the proposal distribution (dashed) might not share high-density region with likelihood (orange), which makes a sample (dot) in high density region of prior distribution have very small density in posterior.

Now, let's address two issues. Why is sampling from posterior not tractable and how can we overcome this. Start with Bayes' rule

$$p(x_{1:t-1}|y_{1:t}) \propto p(y_t|x_{1:t-1}, y_{1:t-1})p(x_{1:t-1}|y_{1:t-1})$$

where we already have expression for the second term. The first expression can be obtained as follows:

$$p(y_t|x_{1:t-1}, y_{1:t-1}) = \int \mathrm{p(y_t, x_t|x_{1:t-1}, y_{1:t-1})dx_t} = \int \mathrm{p(y_t|x_t)p(x_t|x_{1:t-1})dx_t}$$

We need to evaluate an integral for each sample, which has quadratic complexity (number of actual samples times samples used in numerical integration) in total if there are no analytical solutions and numerical integration must be performed. A natural approximation is to assume all the probability mass concentrates at the mean/mode, i.e., $p(x_t|x_{1:t-1}) = \delta(x_t - \mu_t)$ where μ_t is the predicted mean/mode of $p(x_{1:t-1}|y_{1:t-1})$. Then the integral above becomes $p(y_t|x_{1:t-1}, y_{1:t-1}) = p(y_t|\mu_t)$ and the posterior reads as

$$p(x_{1:t-1}|y_{1:t}) \propto p(y_t|\mu_t)p(x_{1:t-1}|y_{1:t-1}).$$

The second term on RHS is just the set of particles we have resulted from the previous iteration. To summarize, we have the two-stage method known as **auxiliary method**:

The first step is for each sample $x_{1:t-1}^{(i)}$, compute the predicted mean/mode $\mu_t^{(i)} = \mathbb{E}_{\mathrm{p\left(x_t\middle|x_{1:t-1}^{(i)}\right)}}[\mathrm{x_t}]$ and its likelihood $p\left(y_t\middle|\mu_t^{(i)}\right)$. Then sample auxiliary index k as follows:

$$k \sim \mathrm{v_{t-1}^{(i)}} = p\left(y_t\middle|\mu_t^{(i)}\right) w_{t-1}^{(i)}.$$

The second step is for each sampled index k, make a move from proposal distribution $q\left(x_t\middle|x_{1:t-1}^{k}\right)$. Plugin these terms into weight update equation w_t:

$$w_t \propto \frac{\mathrm{w_{t-1}}}{p(y_t|\mu_t)w_{t-1}} \frac{p(y_t|x_t)p(x_t|x_{t-1})}{q(x_t|x_{1:t-1})} = \frac{p(y_t|x_t)p(x_t|x_{t-1})}{p(y_t|\mu_t)q(x_t|x_{1:t-1})}.$$

Typically, if one picks the natural transition probability $q(x_t|x_{1:t-1}) = p(x_t|x_{t-1})$ then

$$w_t \propto \frac{p(y_t|x_t)}{p(y_t|\mu_t)}.$$

Combined State and Parameter Estimation

In addition to the dynamic states of a system, we also want to estimate some constant but unknown parameters θ. The problem can be formulated as follows:

$$p(x_{1:t}, \theta|y_{1:t}) \propto p(y_t|x_t, \theta)p(x_t|x_{t-1}, \theta)p(\theta|y_{1:t-1}).$$

Since the parameters don't have dynamics, they will not update no matter how much data we have, which means they are and always will be what they were initialized (probably blindly from prior). One

common approach to handle this is to add artificial dynamics to constant parameters, such as small Brownian motion. Chapter 10 of the “Sequential Monte Carlo Methods in Practice” presented a method to reduce variance on constant parameter estimates.

MCMC Steps

As mentioned above, to sample a new trajectory up to time t, one typically fixes the portion of the trajectory up to time (t-1), conditioned on which a transition is made. This will cause a problem: Given the new data, one cannot change previous states. Intuitively, the estimates are too “stiff”. The problem is even more severe when we have constant but unknown parameters to estimate in addition to the dynamic states. One can, of course, add artificial dynamics to the fixed parameters and estimate them just as another dynamic state. Intuitively, this will lead to bad performance, in the sense that the estimates are wider-spread. To address this, one can optionally perform MCMC moves to the standard SIR (Sequential Importance Resampling) algorithm. If the stationary distribution of the MCMC move is the desired posterior, such move won’t change the latter.

However, rejection rate in MCMC steps can be very high and thus MCMC can take many iterations to converge. Chapter 7 of the “SMC in Practice” presented a sequential version of “annealed importance sampling” method, aiming to reduce iterations in MCMC moves embedded in a particle filter.