CS281: Advanced ML **November 15, 2017**

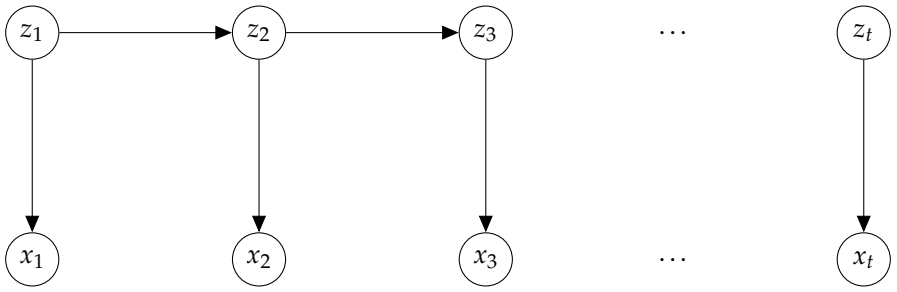
Lecture 20: Importance Sampling and Particle Filtering

Lecturer: Sasha Rush Scribes: Shangyan Li, Alex Wu, Nicolò Foppiani, Kevin Liu, Daniel Merchan, Milan Ravenell

20.1 State space models and Kalman Filter

20.1.1 Introduction on state space models

The first topic which has been covered during this lecture is a review about state space models and Kalman Filter. A state space model (SSM) is a Hidden Markov Model (HMM) with continuous hidden states.



The model is specified by

$$z_t = g_t(z_{t-1}, \epsilon_t)$$
$$x_t = h_t(z_t, \delta_t)$$

where ϵ and δ represent a noise which is added in the transition between two states. z_t is usually referred as the transition model and x_t as the observation model.
An important case is where the transition functions are linear-Gaussian, which means:

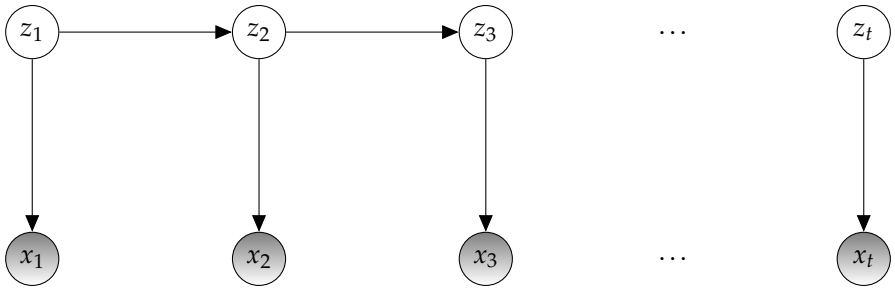
$$z_t = A_t(z_{t-1} + \epsilon_t) \text{ where } \epsilon_t \sim \mathcal{N}(0, Q_t)$$

$$x_t = C_t z_t + \delta_t \text{ where } \delta_t \sim \mathcal{N}(0, R_t)$$

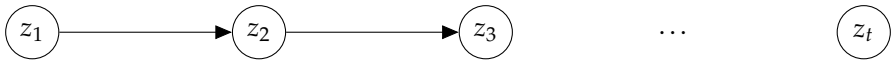
This case is called linear-Gaussian SSM (LG-SSM) or linear dynamical system (LDS).
In most situations we can assume that the process is stationary and considering the model functions independent of time label.
In principle this problem can be solved with the architecture developed for graphical models: this is a directed graphical model in which each variables is Gaussian distributed. Thus the whole model can be solved raising to a multivariate Gaussian for the whole model.
However this model is historically relevant and thus deserves a proper study.

20.1.2 Kalman Filter

In fact, if we know that $p(z_1)$ is MVN, then $p(z_t|x_{1:t})$ will be MVN. Suppose now to have observed $x_{1:t}$ and we want to compute $p(z_t|x_{1:t})$.



We can marginalize, obtaining a tree, on which we can apply exact belief propagation (BP).



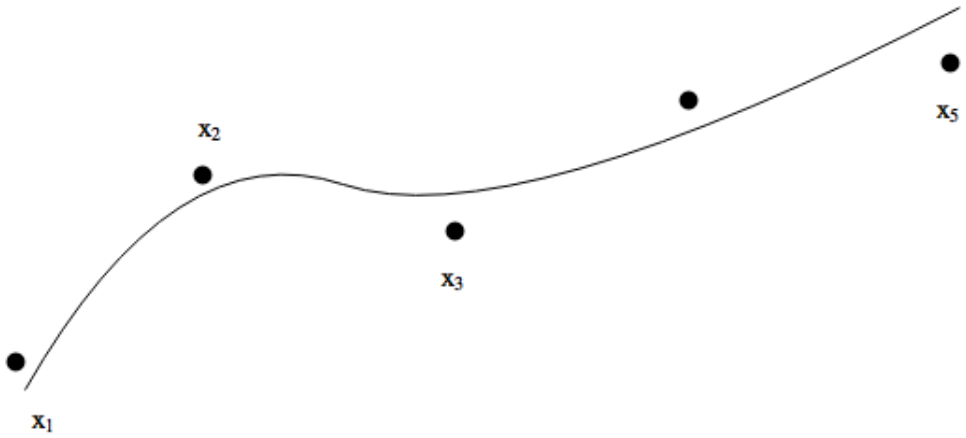
There is a difference with exact belief propagation in the discrete case:

- discrete: Bel(z_t) is an array
- MVN: Bel(z_t) is MVN with $\vec{\mu}$ and Σ

The inference algorithm based on MVN BP in this case is called **Kalman filter**. This is a crucial and popular algorithm, central to multiple technologies. However, at its core, it just runs the LG formula multiple times as part of BP.

20.1.3 Example: LG-SSM for tracking application

We have an object moving in the 2-D plane: the hidden state represents the position and velocity at every timestep, and the observed state is the observed position at the every timestep.



$$\vec{z_t} = [z_{1,t},\ z_{2,t},\ \dot{z}_{1,t},\ \dot{z}_{2,t}]$$

A_t updates $z_t \to z_{t+1}$
C_t generates x_t given z_t: in this case it sets $x_{1,t} = z_{1,t}$ and $x_{2,t} = z_{2,t}$
ϵ_t represents the noise related to physics
δ_t represents the noise generated by the sensors in the observations

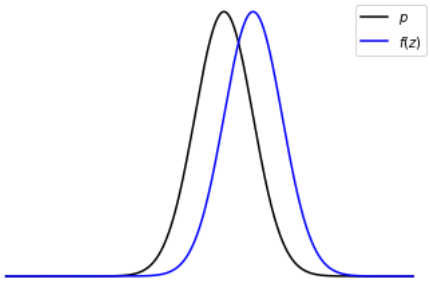
20.1.4 Kalman smoothing

Suppose we observed $x_{1:T}$ and we want to infer z_t. We can use another variant of BP in which we combine the messages from the past and from the future: this is called **Kalman smoothing**.

20.2 Review on importance sampling

20.2.1 Basic Idea

We aim to sample from p to approximate $f(z)$ using Monte Carlo sampling:



To approximate integrals of the form:

$$\mathbb{E}_p[f(z)] = \int q(z)\frac{p(z)}{q(z)}f(z)dz \approx \frac{1}{S}\sum_{i=1}^{S}\frac{p(z^s)}{q(z^s)}f(z^s) = \sum_{i=1}^{S} w^s f(z^s)$$

$w^s = \frac{p(z^s)}{q(z^s)}$ are importance weights associated with each sample z^s, where $z^s \sim q$, our importance distribution.

20.2.2 Unnormalized Distributions

Frequently will have unnormalized target distribution $\tilde{p}(z)$ but not its normalization constant Z_p. Similarly, may have unnormalized proposal $\tilde{q}(z)$ and its unknown normalization constant Z_q.

$$p(z) = \frac{\tilde{p}(z)}{Z_p} \text{ and } q(z) = \frac{\tilde{q}(z)}{Z_q}$$

Substituting into the desired expectation:

$$\mathbb{E}_p[f(z)] = \frac{Z_q}{Z_p}\int q(z)\frac{\tilde{p}(z)}{\tilde{q}(z)}f(z)dz \approx \frac{Z_q}{Z_p}\frac{1}{S}\sum_{i=1}^{S}\tilde{w}^s f(z^s)$$

where $\tilde{w}_s = \frac{\tilde{p}(z^s)}{\tilde{q}(z^s)}$.

Now, how to compute $\frac{Z_q}{Z_p}$? Can also importance sample from q.

$$\frac{Z_p}{Z_q} = \frac{1}{Z_q}\int \tilde{p}(z)dz = \frac{1}{Z_q}\int q(z)\frac{\tilde{p}(z)}{q(z)}dz = \int q(z)\frac{\tilde{p}(z)}{\tilde{q}(z)}dz \approx \frac{1}{S}\sum_{i=1}^{S}\tilde{w}^s$$

Substituting the sampled $\frac{Z_p}{Z_q}$ back:

$$\mathbb{E}_p[f(z)] \approx \sum_{s=1}^{S}\frac{\tilde{w}^s}{\sum_{s'}\tilde{w}^{s'}}f(z^s) = \sum_{s=1}^{S} w^s f(z^s)$$

Note: This form of importance sampling is convenient as it lets us work with unnormalized distributions. However, in many cases is much less convenient than, for instance, rejection sampling or Monte Carlo sampling because it is not fully clear how to use the weights.
Now consider getting samples from p. We need to define $p(x)$ in terms of $\mathbb{E}_p[f(x)]$ using a δ function that returns 1 if it's the value we want, 0 otherwise:

$$p(x = x') = \mathbb{E}_p[\delta_{x'}(x)]$$

Then, our estimate in terms of importance sampling is:

$$p(x) = \sum_{s=1}^{S} w_s \delta_{x^s}(x)$$

But we actually want unweighted samples, which can be obtained through **re-sampling**: pick x^s with probability w^s. In other words, we create a discrete set that approximates the original distribution and then we draw samples from that discrete set based on a categorical distribution weighted by the w_s.

20.2.3 Example

Recall from last class, we have:

$$p(\theta|Data) = \frac{p(Data|\theta)p(\theta)}{p(Data)}$$
$$\tilde{p}(\theta|Data) = p(Data|\theta)p(\theta)$$

Notice we drop the normalization term for $\tilde{p}$. Then $q(\theta) = p(\theta)$, which implies that we importance sample based on the prior. This gives normalized w_s:

$$w_s = \frac{\tilde{p}(\theta_s)/q(\theta_s)}{\sum_{s'} \tilde{p}(\theta_s)/q(\theta_s)} = \frac{p(Data|\theta_s)}{\sum_{s'} p(Data|\theta_{s'})}$$

In summary, we run importance sampling to compute the posterior of the parameters by sampling different parameters from the priors and then by assigning weights to each of those different parameters based on likelihood in the data. The resulting samples are unbiased samples from the distribution of interest.

Exercise 20.1. Importance sampling can be thought of as a form of variance reduction, where we are estimating $\mu = \mathbb{E}_p[f(z)]$ with

$$\mu_q = \frac{1}{S} \sum_{i=1}^{S} \frac{p(z^s)}{q(z^s)} f(z^s), \qquad z^s \sim q$$

We hope that by choosing a "good" q, we will have $\text{Var}[\mu_q] < \text{Var}[\mu]$. Here we will compute this variance reduction and thus find what a "good" q means.

1. Show that the importance-sampled variance is given by

$$\text{Var}[\mu_q] = \int \frac{(f(z)p(z) - \mu q(z))^2}{q(z)} dz$$

2. What choice of $q(z)$ will minimize this variance? With this choice, how many samples do you need to converge?

3. Why is this choice not practical (i.e. implementable)?

Solution

1. From the definition of variance we have

$$\begin{aligned}\text{Var}[\mu_q] &= \int \frac{(f(z)p(z))^2}{q^2(z)} q(z)dz - \mu^2 \\ &= \int \frac{(f(z)p(z))^2}{q(z)} dz - \mu^2 \\ &= \int \frac{(f(z)p(z) - \mu q(z))^2}{q(z)} dz\end{aligned}$$

   The extra factor of $q(z)$ in the first line comes from the fact that $z^s \sim q$.

2. The variance goes to zero if $q(z) = f(z)p(z)/\mu$. With zero variance, every sample has the mean value, thus only one sample is needed.

3. We must be able to sample from $q(z) \sim f(z)p(z)$; however, this implies that we can sample from $f(z)p(z)$ efficiently. But if we could, we wouldn't need to do importance sampling in the first place! So our best bet is to sample from a distribution similar to $f(z)p(z)$ as possible.

20.3 Particle Filtering/Sequential Monte Carlo

There might be cases of space-state models in which it might not be possible to derive closed-form expressions based on LG for the update equations. In those cases, a sampling-based approaches offer an alternative method.
In order to approximate the belief state of an entire sequence, we can use a weighted set of particles as follows:

$$p(z_{1:t} = z'_{1:t}|x_{1:t}) = \int p(z_1, ..., z_t|x_{1:t})\delta_{z'}(z)dz \approx \sum_{s=1}^{S} \hat{w}_t^s \delta_{z'}(z_{1:t}^s)$$

where $\hat{w}_t^s$ represents the normalized weight of sample s at time t.

We update the belief state using importance sampling, with the following importance weights.

$$w_t^s \sim \frac{p(z_{1:t}|x_{1:t})}{q(z_{1:t}|x_{1:t})}$$

The numerator can be expressed as follows:

$$p(z_{1:t}|x_{1:t}) \propto p(z_{1:t}|x_{t-1})p(x|t|z_{1:t}, y_{t-1}) \propto p(x_t|z_t)p(z_t|z_{t-1})p(z_{1:t-1}|x_{1:t})$$

where $p(x_t|z_t)$, $p(z_t|z_{t-1})$, and $p(z_{1:t-1}|x_{1:t})$ correspond to the observation, transition, and recursion of the sequence, respectively.

Meanwhile, we can express the denominator as

$$q(z_{1:t}|x_{1:t}) \propto q(z_t|z_{1:t-1}, y_{1:t})q(z_{1:t-1}|x_{1:t-1})$$

Our importance weights, therefore, simplify to

$$w_t^s \propto \frac{p(x_t|z_t)p(z_t|z_{t-1})p(z_{1:t-1}|x_{1:t})}{q(z_t|z_{1:t-1}, y_{1:t})q(z_{1:t-1}|y_{1:t-1})} = \frac{p(x_t|z_t)p(z_t|z_{t-1})}{q(z_t|z_{1:t-1}, y_{1:t})} w_{t-1}^s$$

Using this expression, we have the following algorithm to approximate the belief state.
1. Start with particles $(w_t, z_t)^{(s)}$.
2. At each time step t for particle x, calculate $w_t^s = \frac{p(x_t|z_t)p(z_t|z_{t-1})}{q(z_t|z_{1:t-1}, y_{1:t})} w_{t-1}^s$ and sample $z_t^s \sim q(z_t|z_{1:t-1}^s, x_t)$.
3. Compute $p(z_t, x_{1:t})$

References

Owen, Art, *Importance Sampling*, 2013, https://statweb.stanford.edu/~owen/mc/Ch-var-is.pdf