Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.438 Algorithms for Inference
Fall 2020

24 Particle Methods

The stochastic methods for approximate inference we have developed thus far—i.e., Markov Chain Monte Carlo (MCMC)—have been based on rejection sampling concepts. We now turn to *particle methods*, which are stochastic methods based on what is termed *importance sampling.* Both classes of methods are useful for efficiently sampling from distributions over both large finite alphabets and continuous-valued ones. However, particle methods are particularly popular for applications involving the latter, so our development will emphasize this scenario.

Before proceeding, recall that to this point, the only continuous-valued variables we know how to handle efficiently for inference are Gaussian ones. As we saw, for jointly Gaussian variables, conditioning and marginalization preserve Gaussianity, and messages in belief propagation also (essentially) share the Gaussian (i.e., exponential-quadratic) form. As a result, we were able to avoid the explicit integrations that inference with such distributions would otherwise require, and instead represent all quantities of interest by their finite dimensional parameterizations (such as in terms of means and covariances). In this respect, Gaussian models, while highly effective and widely used, are very special.

For nonGaussian models, we must find other ways to avoid the explicit integrations that arise with continuous variables. One conceptually straightforward approach would involve discretization of the underlying variables. However, in practice, this often leads to discrete variables over very large alphabets in order to minimize the impact of quantization on the ensuing inference. The methods we now develop can be viewed as a different and often much more efficient way to pursue such quantization.

24.1 Sampling as Discrete Approximation of Distributions

Consider a continuous-valued random variable $\mathsf{x} \in \mathbb{R}$ with probability density function $p(x)$.[1] Next, we generate sample $x^1, \dots, x^K$ from p, and define

$$\hat{p}(x) \triangleq \frac{1}{K} \sum_{k=1}^{K} \delta(x - x^k), \tag{1}$$

where $\delta(\cdot)$ denotes the usual Dirac delta (generalized) function, i.e., impulse, which has the defining "sifting" property that for all g that are continuous at 0,

$$\int_{-\infty}^{+\infty} \delta(u)\, g(u)\, \mathrm{d}u = g(0). \tag{2}$$

[1] To simplify the initial exposition, we restrict our attention for the moment to the case $\mathcal{X} = \mathbb{R}$. However, in general we are interested in the vector case in which $\mathcal{X} = \mathbb{R}^M$ for some M that may be quite large.

Then, in an appropriate sense, $\hat{p}$ is a discrete approximation to $p(x)$. Obviously $\hat{p}$, which is discrete, doesn't "look" like any continuous distribution. However, expectations and other integrations with respect to $\hat{p}$ can be close to those with respect to p. Indeed,

$$\begin{aligned}
\mathbb{E}_{\hat{p}}\left[g(\mathsf{x})\right] &= \int_{-\infty}^{+\infty} \hat{p}(x)\, g(x)\, \mathrm{d}x \\
&= \int_{-\infty}^{+\infty} \frac{1}{K} \sum_{k=1}^{K} \delta(x - x^k)\, g(x)\, \mathrm{d}x \\
&= \frac{1}{K} \sum_{k=1}^{K} \int_{-\infty}^{+\infty} \delta(x - x^k)\, g(x)\, \mathrm{d}x \\
&= \frac{1}{K} \sum_{k=1}^{K} g(x^k) \\
&\xrightarrow{\text{a.s.}} \mathbb{E}_p\left[g(\mathsf{x})\right], \qquad K \to \infty \\
&= \int_{-\infty}^{+\infty} p(x)\, g(x)\, \mathrm{d}x,
\end{aligned}$$

where $\xrightarrow{\text{a.s.}}$ denotes almost sure convergence, i.e., convergence with probability one, which follows from the strong law of large numbers.

Hence, for sufficiently large K we have $\mathbb{E}_{\hat{p}}\left[g(\mathsf{x})\right] \cong \mathbb{E}_p\left[g(\mathsf{x})\right]$, so $\hat{p}$ is a convenient tool for approximate inference. For example, if $g(x) = \mathbb{1}_{x \in \mathcal{S}}$ for some set $\mathcal{S} \subset \mathbb{R}$, then

$$\mathbb{E}_{\hat{p}}\left[g(\mathsf{x})\right] = \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}_{x^k \in \mathcal{S}} \cong \mathbb{P}\left(\mathsf{x} \in \mathcal{S}\right) = \mathbb{E}_p\left[g(\mathsf{x})\right].$$

We say that (1) is an approximation to p based on K *particles*, each of which are represented by an impulse, and the *weight* of each impulse is $1/K$. The distribution $\hat{p}$ is depicted in Fig. 1, which indicates the weights explicitly. However, particle representations with other weightings are possible, and quite useful, as we develop next.

24.2 Importance Sampling and Weighted Approximations

Suppose we are given $\tilde{p}(x)$ such that

$$p(x) = \frac{\tilde{p}(x)}{Z}. \tag{3}$$

In practice, p can be hard to sample from, because the integration involved in computing the partition function, viz.,

$$Z = \int_{-\infty}^{+\infty} \tilde{p}(x)\, \mathrm{d}x \tag{4}$$

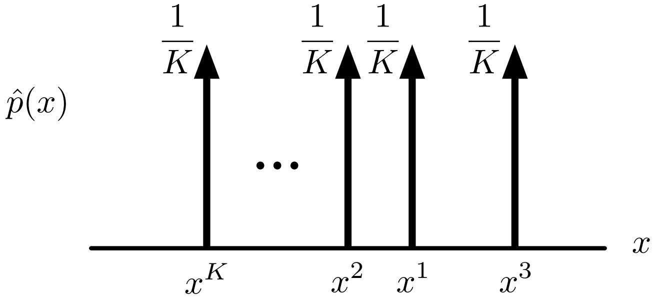

Figure 1: An approximation $\hat{p}(\cdot)$ to distribution $p(\cdot)$ based on samples $x^1, \ldots, x^K$ from p.

can be computationally infeasible.[2]

However, if we have some *proposal* distribution q that we can sample from,[3] then we can construct a suitable $\hat{p}$ from samples $x^1, \ldots, x^K$ generated from q. In particular, if we choose (normalized) weights

$$w^k \triangleq \frac{\tilde{w}^k(x^k)}{\sum_{k'=1}^K \tilde{w}^{k'}(x^{k'})}, \qquad \tilde{w}^k(x^k) \triangleq \frac{\tilde{p}(x^k)}{q(x^k)}. \tag{5}$$

Then

$$\hat{p}(x) = \sum_{k=1}^K w^k \, \delta(x - x^k) \tag{6}$$

is also an arbitrarily good approximation to p in the same sense of expectation. In

[2] In this case, since $\mathcal{X} = \mathbb{R}$, this is need not be so. However, more generally, when $\mathcal{X} = \mathbb{R}^M$ for sufficiently large M, it can be so.

[3] We also require that the support of p be included in that of q, i.e., $q(x) > 0$ for any x such that $p(x) > 0$.

particular, note that

$$\begin{aligned}
\mathbb{E}_{\hat{p}}\left[g(\mathsf{x})\right] &= \int \hat{p}(x)\, g(x)\mathrm{d}x \\
&= \int \sum_{k=1}^{K} w^k\, \delta(x - x^k)\, g(x)\mathrm{d}x \\
&= \sum_{k=1}^{K} w^k \int \delta(x - x^k)\, g(x)\mathrm{d}x \\
&= \sum_{k=1}^{K} w^k g(x^k) \\
&= \sum_{k=1}^{K} \frac{\tilde{p}(x^k)/q(x^k)}{\sum_{k'=1}^{K} \tilde{p}(x^{k'})/q(x^{k'})} g(x^k) \\
&= \frac{\dfrac{1}{K}\displaystyle\sum_{k=1}^{K} g(x^k)\, \tilde{p}(x^k)/q(x^k)}{\dfrac{1}{K}\displaystyle\sum_{k'=1}^{K} \tilde{p}(x^{k'})/q(x^{k'})} \qquad (7)
\end{aligned}$$

But the numerator of (7) satisfies

$$\begin{aligned}
\frac{1}{K}\sum_{k=1}^{K} g(x^k)\, \tilde{p}(x^k)/q(x^k) \xrightarrow{\text{a.s.}} & \ \mathbb{E}_q\left[g(\mathsf{x})\, \tilde{p}(\mathsf{x})/q(\mathsf{x})\right], \qquad K \to \infty \\
&= \int_{-\infty}^{+\infty} q(x)\, \big[g(x)\, Z\, p(x)/q(x)\big]\, \mathrm{d}x \\
&= Z\, \mathbb{E}_p\left[g(\mathsf{x})\right],
\end{aligned}$$

and the denominator similarly satisfies

$$\begin{aligned}
\frac{1}{K}\sum_{k'=1}^{K} \tilde{p}(x^{k'})/q(x^{k'}) \xrightarrow{\text{a.s.}} & \ \mathbb{E}_q\left[\tilde{p}(\mathsf{x})/q(\mathsf{x})\right], \qquad K \to \infty \\
&= \int_{-\infty}^{+\infty} q(x)\, \big[Z\, p(x)/q(x)\big]\, \mathrm{d}x \\
&= Z.
\end{aligned}$$

Hence, we have the (7) converges to

$$\mathbb{E}_{\hat{p}}\left[g(\mathsf{x})\right] \xrightarrow{\text{a.s.}} \frac{Z\, \mathbb{E}_p\left[g(\mathsf{x})\right]}{Z} = \mathbb{E}_p\left[g(\mathsf{x})\right], \qquad (8)$$

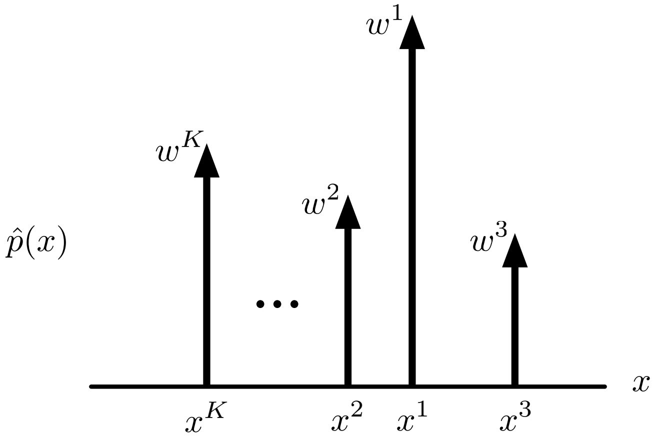

Figure 2: An approximation $\hat{p}(\cdot)$ to distribution $p(\cdot)$ based on samples $x^1, \dots, x^K$ from q, where the weights are defined via (5).

as desired.

The distribution (6) is depicted in Fig. 2, with the weights indicated. The weights w^k compensate for the fact that q is producing samples that are not correctly distributed (for p). In particular, in regions of $\mathcal{X} = \mathbb{R}$ where q has higher probability that p, so sampling from q produces too many particles, the effect of these particles is reduced by assigning collectively smaller weights to them, in the appropriate proportions, and vice-versa.

In general, the rate of convergence of $\mathbb{E}_{\hat{p}}\left[g(\mathsf{x})\right]$ to $\mathbb{E}_{p}\left[g(\mathsf{x})\right]$ with K depends on how different q and p are, and in particular how it takes before there are enough samples from q in regions of p of high probability. As a rough rule, the smaller the domain for p (i.e., the smaller M is for $\mathcal{X} = \mathbb{R}^M$), the better the convergence behavior, though obviously the choice of q has a significant effect.

24.3 Marginalization with Importance Sampling

It is straightforward to generate approximations for marginal distributions from those of a joint distribution. We illustrate the key ideas for the special case $\mathbf{x} = (\mathsf{x}_1, \mathsf{x}_2) \in \mathbb{R}^2$.

Suppose that $p_{\mathsf{x}_1,\mathsf{x}_2}$ is the joint distribution of interest, from which we want approximate the marginal

$$p_{\mathsf{x}_1}(x_1) = \int_{-\infty}^{+\infty} p_{\mathsf{x}_1,\mathsf{x}_2}(x_1, x_2)\,\mathrm{d}x_2.$$

In this case, we generate samples

$$\mathbf{x}^k = (x_1^k, x_2^k), \qquad k = 1, \dots, K \tag{9}$$

from a distribution with density $q(x_1, x_2)$, and generate an approximating density

$$\hat{p}_{\mathsf{x}_1,\mathsf{x}_2}(x_1, x_2) = \sum_{k=1}^{K} w^k \, \delta(x_1 - x_1^k) \, \delta(x_2 - x_2^k) \tag{10}$$

where

$$w^k \triangleq \frac{\tilde{p}(x_1^k, x_2^k)/q(x_1^k, x_2^k)}{\sum_{k'=1}^{K} \tilde{p}(x_1^{k'}, x_2^{k'})/q(x_1^{k'}, x_2^{k'})} \tag{11}$$

are the weights.

Then the marginal for x_1 is simply

$$\begin{align} \hat{p}_{\mathsf{x}_1}(x_1) &= \int_{-\infty}^{+\infty} \hat{p}_{\mathsf{x}_1,\mathsf{x}_2}(x_1, x_2) \, \mathrm{d}x_2 \tag{12} \\ &= \sum_{k=1}^{K} w^k \, \delta(x_1 - x_1^k) \tag{13} \end{align}$$

where we emphasize the weights w^k are unchanged. Note that this approximation retains the qualities of the joint, viz., for any g_1 of interest,

$$\mathbb{E}_{\hat{p}_{\mathsf{x}_1}}\left[g_1(\mathsf{x}_1)\right] \xrightarrow{\text{a.s.}} \mathbb{E}_{p_{\mathsf{x}_1}}\left[g_1(\mathsf{x}_1)\right], \qquad K \to \infty. \tag{14}$$

Indeed it suffices to use that

$$\mathbb{E}_{\hat{p}_{\mathsf{x}_1,\mathsf{x}_2}}\left[g(\mathsf{x}_1, \mathsf{x}_2)\right] \xrightarrow{\text{a.s.}} \mathbb{E}_{p_{\mathsf{x}_1,\mathsf{x}_2}}\left[g(\mathsf{x}_1, \mathsf{x}_2)\right], \qquad K \to \infty \tag{15}$$

with $g(x_1, x_2) = g_1(x_1)$.

The marginal for x_2 is obtained similarly, as are extensions to distributions over N variables.

24.4 Posterior Approximation via Importance Sampling

Importance sampling is often useful for sampling from posterior distributions. To illustrate this application, suppose that $\mathsf{x} \in \mathbb{R}$ represents a variable of interest, and $\mathsf{y} \in \mathbb{R}$ represents an observed variable. Then the posterior density can be expressed in the form

$$p_{\mathsf{x}|\mathsf{y}}(x|y) = \frac{1}{Z} \, p_{\mathsf{x}}(x) \, p_{\mathsf{y}|\mathsf{x}}(y|x),$$

where the observation model $p_{\mathsf{y}|\mathsf{x}}$ is typically specified directly or easily computed, and where the prior p_{x} is often chosen to be easy to sample from. However, when the partition function

$$Z = \int_{-\infty}^{+\infty} p_{\mathsf{x}}(x) \, p_{\mathsf{y}|\mathsf{x}}(y|x) \, \mathrm{d}x$$

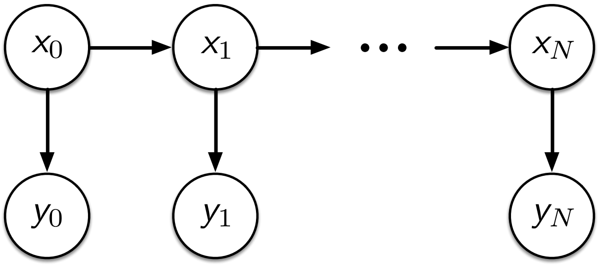

Figure 3: The representation of a general hidden Markov model as a directed graphical model.

is hard to compute, it is difficult to sample from the posterior directly. In such cases, we can use importance sampling with

$$\tilde{p}(x) = p_{\mathsf{x}}(x)\, p_{\mathsf{y}|\mathsf{x}}(y|x),$$

and the prior as the proposal density, i.e., $q = p_{\mathsf{x}}$. In this case, the associated weights for the particles generated from p_{x} are

$$w^k \propto \tilde{w}^k(x^k) = p_{\mathsf{y}|\mathsf{x}}(y|x^k). \tag{16}$$

This example extends to higher dimensional latent and observed variables in the obvious way, and is especially useful in such scenarios.

24.5 Posterior Beliefs in Continuous Hidden Markov Models

In this section we consider the general problem of posterior belief computation in continuous-state HMMs. In our development, we use the factorization

$$\begin{aligned} &p_{\mathsf{y}_0,\ldots,\mathsf{y}_N,\mathsf{x}_0,\ldots,\mathsf{x}_N}(y_0,\ldots,y_N,x_0,\ldots,x_N) \\ &\qquad = p_{\mathsf{x}_0,\ldots,\mathsf{x}_N}(x_0,\ldots,x_N)\, p_{\mathsf{y}_0,\ldots,\mathsf{y}_N|\mathsf{x}_0,\ldots,\mathsf{x}_N}(y_0,\ldots,y_N|x_0,\ldots,x_N) \end{aligned} \tag{17a}$$

with

$$p_{\mathsf{x}_0,\ldots,\mathsf{x}_N}(x_0,\ldots,x_N) = p_{\mathsf{x}_0}(x_0) \prod_{i=1}^{N} p_{\mathsf{x}_i|\mathsf{x}_{i-1}}(x_i|x_{i-1}) \tag{17b}$$

$$p_{\mathsf{y}_0,\ldots,\mathsf{y}_N|\mathsf{x}_0,\ldots,\mathsf{x}_N}(y_0,\ldots,y_N|x_0,\ldots,x_N) = \prod_{i=0}^{N} p_{\mathsf{y}_i|\mathsf{x}_i}(y_i|x_i), \tag{17c}$$

which corresponds to the (directed) graphical representation depicted of Fig. 3.

As in the case of discrete hidden Markov models (HMMs), there are two types of posterior beliefs often of most interest:

1. *Causal beliefs* $p_{\mathsf{x}_i|\mathsf{y}_0,\ldots,\mathsf{y}_i}(\cdot|y_0,\ldots,y_i)$, for $i=0,\ldots,N$, which are often of interest when i is a time index and the target application involves real-time inference, so future observations are not available. The process of generating causal beliefs is sometimes referred to as *filtering.*

2. *Noncausal beliefs* $p_{\mathsf{x}_i|\mathsf{y}_0,\ldots\, \mathsf{y}_N}(\cdot|y_0,\ldots,y_N)$, for $i=0,\ldots,N$, which take into account all the available observations. The process of generating noncausal beliefs is sometimes referred to as *smoothing.*

An in the case of these discrete HMM models, we developed the forward-backward algorithm as an efficient procedure for obtaining both types of posterior beliefs. For example, the (α,β) form of the algorithm provides the causal beliefs via the forward pass through the data, and then the noncausal beliefs via the backward pass.

Our derivation of this algorithm did not depend on the discrete nature of the HMM (excepting our use of summations instead of integrations). In the remainder of this section, we carry out this extension to illustrate the computation involved.

24.5.1 Causal Belief Computation

To begin, referring to the forward-backward algorithm equations, starting from $\alpha_0(x_0)=p_{\mathsf{x}_0,\mathsf{y}_0}(x_0,y_0)$, we have the iteration

$$\alpha_{i+1}(x_{i+1})=\int_{-\infty}^{+\infty} p_{\mathsf{x}_{i+1}|\mathsf{x}_i}(x_{i+1}|x_i)\,p_{\mathsf{y}_{i+1}|\mathsf{x}_{i+1}}(y_{i+1}|x_{i+1})\,\alpha_i(x_i)\,\mathrm{d}x_i,\quad 0=1,\ldots,N-1, \tag{18}$$

where

$$\alpha_i(x_i)=p_{\mathsf{x}_i,\mathsf{y}_0,\ldots,\mathsf{y}_i}(x_i,y_0,\ldots,y_i).$$

But since the causal posteriors are proportional to the forward messages, we have

$$\begin{aligned}
&p_{\mathsf{x}_{i+1}|\mathsf{y}_0,\ldots,\mathsf{y}_{i+1}}(x_{i+1}|y_0,\ldots,y_{i+1})\\
&\quad\propto \alpha_{i+1}(x_{i+1}) &(19)\\
&\quad\propto \int_{-\infty}^{+\infty} p_{\mathsf{x}_{i+1}|\mathsf{x}_i}(x_{i+1}|x_i)\,p_{\mathsf{y}_{i+1}|\mathsf{x}_{i+1}}(y_{i+1}|x_{i+1})\,p_{\mathsf{x}_i|\mathsf{y}_0,\ldots,\mathsf{y}_i}(x_i|y_0,\ldots,y_i)\,\mathrm{d}x_i &(20)\\
&\quad= p_{\mathsf{y}_{i+1}|\mathsf{x}_{i+1}}(y_{i+1}|x_{i+1})\underbrace{\int_{-\infty}^{+\infty} p_{\mathsf{x}_{i+1}|\mathsf{x}_i}(x_{i+1}|x_i)\,p_{\mathsf{x}_i|\mathsf{y}_0,\ldots,\mathsf{y}_i}(x_i|y_0,\ldots,y_i)\,\mathrm{d}x_i,}_{=p_{\mathsf{x}_{i+1}|\mathsf{y}_0,\ldots,\mathsf{y}_i}(x_{i+1}|y_0,\ldots,y_i)} &(21)
\end{aligned}$$

where we have recognized that the integral in (21) is a one-step prediction of the state from the current observations. As a result, we see we can describe the forward pass (i.e., filtering procedure) as consisting of two steps:

1. *Prediction step:*

$$p_{\mathsf{x}_{i+1}|\mathsf{y}_0,\ldots,\mathsf{y}_i}(x_{i+1}|y_0,\ldots,y_i)=\int_{-\infty}^{+\infty} p_{\mathsf{x}_{i+1}|\mathsf{x}_i}(x_{i+1}|x_i)\,p_{\mathsf{x}_i|\mathsf{y}_0,\ldots,\mathsf{y}_i}(x_i|y_0,\ldots,y_i)\,\mathrm{d}x_i$$

2. *Update step:*

$$p_{\mathsf{x}_{i+1}|\mathsf{y}_0,\dots,\mathsf{y}_{i+1}}(x_{i+1}|y_0,\dots,y_{i+1}) = \frac{1}{Z}\, p_{\mathsf{y}_{i+1}|\mathsf{x}_{i+1}}(y_{i+1}|x_{i+1})\, p_{\mathsf{x}_{i+1}|\mathsf{y}_0,\dots,\mathsf{y}_i}(x_{i+1}|y_0,\dots,y_i)$$

where

$$Z = \int_{-\infty}^{+\infty} p_{\mathsf{y}_{i+1}|\mathsf{x}_{i+1}}(y_{i+1}|x_{i+1})\, p_{\mathsf{x}_{i+1}|\mathsf{y}_0,\dots,\mathsf{y}_i}(x_{i+1}|y_0,\dots,y_i)\, \mathrm{d}x_{i+1}$$

From this form, we see exact computation of causal posterior beliefs involves two integrations, one for each step.

24.5.2 Noncausal Belief Computation

We can similarly derive the backward pass (i.e., smoothing procedure) for continuous HMMs by a minor extension of the discrete case. In particular, referring again to the forward-backward equations, starting from $\beta_N(x_N) = 1$, we have the iteration

$$\beta_i(x_i) = \int_{-\infty}^{+\infty} p_{\mathsf{x}_{i+1}|\mathsf{x}_i}(x_{i+1}|x_i)\, p_{\mathsf{y}_{i+1}|\mathsf{x}_{i+1}}(y_{i+1}|x_{i+1})\, \beta_{i+1}(x_{i+1})\, \mathrm{d}x_{i+1}, \quad i = 1,\dots,N-1, \tag{22}$$

where

$$\beta_i(x_i) = p_{\mathsf{y}_{i+1},\dots,\mathsf{y}_N|\mathsf{x}_i}(y_{i+1},\dots,y_N|x_i),$$

and, in turn,

$$p_{\mathsf{x}_i|\mathsf{y}_0,\dots,\mathsf{y}_N}(x_i|y_0,\dots,y_N) \propto \alpha_i(x_i)\, \beta_i(x_i), \qquad i = 1,\dots,N. \tag{23}$$

Hence, we can also describe the backward pass as consisting of two steps (which, by contrast, do not need to be executed in alternation):

1.

$$\begin{aligned} &p_{\mathsf{y}_{i+1},\dots,\mathsf{y}_N|\mathsf{x}_i}(y_{i+1},\dots,y_N|x_i) \\ &\quad = \int_{-\infty}^{+\infty} p_{\mathsf{x}_{i+1}|\mathsf{x}_i}(x_{i+1}|x_i)\, p_{\mathsf{y}_{i+1}|\mathsf{x}_{i+1}}(y_{i+1}|x_{i+1})\, p_{\mathsf{y}_{i+2},\dots,\mathsf{y}_N|\mathsf{x}_{i+1}}(y_{i+2},\dots,y_N|x_{i+1})\, \mathrm{d}x_{i+1} \end{aligned}$$

2.

$$p_{\mathsf{x}_i|\mathsf{y}_0,\dots,\mathsf{y}_N}(x|y_0,\dots,y_N) = \frac{1}{Z}\, p_{\mathsf{x}_i|\mathsf{y}_0,\dots,\mathsf{y}_i}(x_i|y_0,\dots,y_i)\, p_{\mathsf{y}_{i+1},\dots,\mathsf{y}_N|\mathsf{x}_i}(y_{i+1},\dots,y_N|x_i)$$

where

$$Z = \int p_{\mathsf{x}_i|\mathsf{y}_0,\dots,\mathsf{y}_i}(x_i|y_0,\dots,y_i)\, p_{\mathsf{y}_{i+1},\dots,\mathsf{y}_N|\mathsf{x}_i}(y_{i+1},\dots,y_N|x_i)\, \mathrm{d}x_i.$$

Again, from this form we see that computation of noncausal posterior beliefs involves two further integrations, one for each step.

Among other reconfigurations of the backward pass is that corresponding to the (α, γ) version of the forward-backward algorithm:

1. $$\begin{aligned} & p_{\mathsf{x}_i|\mathsf{y}_0,\ldots,\mathsf{y}_N}(x_i|y_0,\ldots,y_N) \\ & \qquad = \int_{-\infty}^{+\infty} p_{\mathsf{x}_i|\mathsf{x}_{i+1},\mathsf{y}_0,\ldots,\mathsf{y}_i}(x_i|x_{i+1},y_0,\ldots,y_i)\, p_{\mathsf{x}_{i+1}|\mathsf{y}_0,\ldots,\mathsf{y}_N}(x_{i+1}|y_0,\ldots,y_N)\,\mathrm{d}x_{i+1} \end{aligned}$$

2. $$p_{\mathsf{x}_i|\mathsf{x}_{i+1},\mathsf{y}_0,\ldots,\mathsf{y}_i}(x_i|x_{i+1},y_0,\ldots,y_i) = \frac{p_{\mathsf{x}_{i+1}|\mathsf{x}_i}(x_{i+1}|x_i)\, p_{\mathsf{x}_i|\mathsf{y}_0,\ldots,\mathsf{y}_i}(x_i|y_0,\ldots,y_i)}{p_{\mathsf{x}_{i+1}|\mathsf{y}_0,\ldots,\mathsf{y}_i}(x_{i+1}|y_0,\ldots,y_i)}$$

   which we can compute from quantities obtained during the forward pass.

Evidently, this saves one integration.

Regardless of the formulation, exact computation of both causal and noncausal posterior beliefs involves integrations that are typically prohibitive in practice. However, these beliefs can be approximated by using particle methods, as we now develop. As a first step, we begin by summarizing how to sample from Markov models.

24.6 Sampling from Markov Models

Sampling from Markov models—whether discrete-state or continuous-state—can be accomplished quite efficiently in practice, as our development of MCMC exploited. Here we summarize the sampling procedure for the case in which the states are scalars ($\mathcal{X} = \mathbb{R}$). In particular, suppose $p_{\mathsf{x}_0,\ldots,\mathsf{x}_N}$ represents a Markov model, i.e., factorizes according to (17b). Then to generate a sample $(x'_0,\ldots,x'_N)$, we use the following prodedure:

1. Generate a sample x'_0 from $p_{\mathsf{x}_0}(\cdot)$ and set $i = 1$.
2. Generate a sample x'_i from $p_{\mathsf{x}_i|\mathsf{x}_{i-1}}(\cdot|x'_{i-1})$ using the previous sample x'_{i-1}.
3. Increment i and return to step 2; stop after generating x'_N.

This generates a single sample from $p_{\mathsf{x}_0,\ldots,\mathsf{x}_N}$. By repeating this procedure, we can generate K samples

$$(x_0^1,\ldots,x_N^1),\ldots,(x_0^K,\ldots,x_N^K).$$

24.7 Particle Filtering and Smoothing

The key to efficient approximation of posterior beliefs in HMMs is applying the the importance sampling approach of Section 24.4.

Considering first the approximation of causal beliefs, note that after generating K independent samples from p_{x_0}, viz.,

$$x_0^k \sim p_{\mathsf{x}_0}(\cdot), \quad k = 1, \dots, K,$$

we can approximate $p_{\mathsf{x}_0|\mathsf{y}_0}(\cdot|y_0)$ via [cf. (16)]

$$\hat{p}_{\mathsf{x}_0|\mathsf{y}_0}(x_0|y_0) \propto \sum_{k=1}^{K} \tilde{w}_0^k(x_0^k)\, \delta(x_0 - x_0^k), \qquad \tilde{w}_0^k(x_0^k) \triangleq p_{\mathsf{y}_0|\mathsf{x}_0}(y_0|x_0^k).$$

In turn, we can generate K independent samples

$$x_1^k \sim p_{\mathsf{x}_1|\mathsf{x}_0}(\cdot|x_0^k), \quad k = 1, \dots, K,$$

and similarly approximate $p_{\mathsf{x}_0,\mathsf{x}_1|\mathsf{y}_0,\mathsf{y}_1}(\cdot,\cdot|y_0,y_1)$ via

$$\hat{p}_{\mathsf{x}_0,\mathsf{x}_1|\mathsf{y}_0,\mathsf{y}_1}(x_0,x_1|y_0,y_1) \propto \sum_{k=1}^{K} \tilde{w}_1^k(x_0^k,x_1^k)\, \delta(x_0 - x_0^k)\, \delta(x_1 - x_1^k),$$

with

$$\tilde{w}_1^k(x_0^k,x_1^k) \triangleq p_{\mathsf{y}_1|\mathsf{x}_1}(y_1|x_1)\, \tilde{w}_0^k(x_0^k),$$

which when we marginalize, as discussed in Section 24.3, yields

$$\hat{p}_{\mathsf{x}_1|\mathsf{y}_0,\mathsf{y}_1}(x_1|y_0,y_1) \propto \sum_{k=1}^{K} \tilde{w}_1^k(x_0^k,x_1^k)\, \delta(x_1 - x_1^k).$$

Proceeding in this manner, the following iterative procedure becomes apparent for generating approximations to the causal posterior beliefs: starting with the initialization above, for $i = 1, \dots, N$ we generate particles

$$x_i^k \sim p_{\mathsf{x}_i|\mathsf{x}_{i-1}}(\cdot|x_{i-1}^k), \quad i = 1, \dots, K \tag{24}$$

from which we obtain the causal posterior belief approximation

$$\hat{p}_{\mathsf{x}_i|\mathsf{y}_0,\dots,\mathsf{y}_i}(x_i|y_0,\dots,y_i) \propto \sum_{k=1}^{K} \tilde{w}_i^k(x_0^k,\dots,x_i^k)\, \delta(x_i - x_i^k) \tag{25}$$

with (unnormalized) weights

$$\tilde{w}_i^k(x_0^k,\dots,x_i^k) \propto p_{\mathsf{y}_i|\mathsf{x}_i}(y_i|x_i)\, \tilde{w}_{i-1}^k(x_0^k,\dots,x_{i-1}^k). \tag{26}$$

This procedure is generally referred to as *particle filtering.*

The noncausal posteriors are similarly straightforward to compute. In particular, note that if we were to have avoided the intermediate marginalizations, at the end of the particle filtering pass we have the belief approximation

$$\hat{p}_{\mathsf{x}_0,\ldots,\mathsf{x}_N|\mathsf{y}_0,\ldots,\mathsf{y}_N}(x_0,\ldots,x_N|y_0,\ldots,y_N) \propto \sum_{k=1}^{K} \tilde{w}_N^k(x_0^k,\ldots,x_N^k) \prod_{i=0}^{N} \delta(x_i - x_i^k).$$

Hence, again applying the results of Section 24.3, the posterior marginals immediately follow as

$$\hat{p}_{\mathsf{x}_i|\mathsf{y}_0,\ldots,\mathsf{y}_N}(x_i|y_0,\ldots,y_N) \propto \sum_{k=1}^{K} \tilde{w}_N^k(x_0^k,\ldots,x_N^k)\, \delta(x_i - x_i^k).$$

This procedure can be viewed as *particle smoothing.*

Particle filtering and smoothing is sometimes referred to as *sequential Monte Carlo (SMC)* or *sequential importance sampling (SIS).* Some aspects of using this approach to posterior belief approach are worth emphasizing.

First, note that we never explicitly need to compute or use approximations to the joint posterior $p_{\mathsf{x}_0,\ldots,\mathsf{x}_N|\mathsf{y}_0,\ldots,\mathsf{y}_N}$, which an MCMC approach based on Metropolis Hastings, such as Gibbs sampling, would. Instead, we only ever generate and use marginals. As a result, particle filtering and smoothing require many fewer samples to obtain good belief approximations compared with more general MCMC approaches, even though both exploit the factorization structure in the HMM.

Second, for some classes of continuous-state Markov processes, other approaches to approximate posterior belief computation can be used. For example, when the Markov process corresponds to a *nonlinear* dynamical system, one can use linearizations around current casual belief mean approximations to generate subsequent ones. The resulting procedure is typically referred to as an *extended Kalman filter (EKF).* This is the version of the Kalman filter that was used in the Apollo space program to navigate to the Moon. And while a rich topic in its own right, its analysis is challenging and beyond the scope of our subject.

24.8 Importance Resampling

In practice effective "particle management" with importance sampling can improve the computational efficiency of particle filtering and other applications. In particular, when there are many particles with small weights, the contributions from such particles is small despite their contribution to the computational burden of particle-based inference. Accordingly, particle methods are typically combined with *resampling.*

In a simple form of such resampling, the K particles $x^1,\ldots,x^K$ are replaced with K independent samples $x_*^1,\ldots,x_*^K$ drawn from $\hat{p}$ in (6), which of course is a function of the previous drawn samples. This is referred to as *multinomial resampling*, and can be implemented with $O(K)$ complexity. The resulting "new" distribution approximation is

$$\hat{p}_*(x) = \sum_{k=1}^{K} w_*^k \, \delta(x - x_*^k), \tag{27}$$

where

$$w_*^k = \frac{1}{K} \sum_{j=1}^{K} \mathbb{1}_{x_*^j = x_*^k}.$$

Observe that by construction, with high probability resampling will remove samples with comparatively low weights. While this can significantly reduce the computational complexity of the associated approximate inference, there is typically also at least a small accompanying degradation in the quality of such inference. More elaborate resampling methods can mitigate such effects.

To decide when to resample, one could use, e.g.,

$$K_{\text{eff}} \triangleq \frac{1}{\sum_{k=1}^{K} (w^k)^2}$$

as a measure of how good the current weights are, since $1 \le K_{\text{eff}} \le K$, where the upper bound is attained when all weights are equal. Hence, we could decide to resample whenever K_{eff} falls below a threshold—e.g., $K/2$.

24.9 Particle-Based Approximations to Belief Propagation

Particle filtering and smoothing can be viewed as approximate belief propagation on HMMs. From this perspective, we are also interested in such approximations over more general graphs, such as trees, and even graphs with cycles. Indeed, one can construct particle-based methods for approximating the sum-product algorithm on trees, and for approximating loopy BP. There are a couple of issues and subtleties that arise in these more general settings that do not in the case of HMMs.

For more details on particle-based belief propagation, see, e.g.,

E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky, "Nonparametric Belief Propagation," in *Proceedings of CVPR*, June 2003.

A. T. Ihler, E. B. Sudderth, W. T. Freeman, and A. S. Willsky, "Efficient Multiscale Sampling From Products of Gaussian Mixtures," *Proceedings of NIPS*, 2003.

R. van Handel, "Monte Carlo Methods: Interacting Particles," in *Hidden Markov Models*, Lecture Notes, July 2008.