10-708: Probabilistic Graphical Models 10-708, Spring 2014

1 : Introduction

Lecturer: Eric P. Xing *Scribes: Nobody*

1 Introduction

For simple models, exact inference is possible through algorithms elimination, message passing, and junction trees. However, these are often computationally infeasible for larger models and it is necessary to use approximate inference techniques. This lecture focuses on Monte Carlo methods which approach inference using stochastic simulation through sampling.

To motivate this idea, recall that the distribution of a random variable X can be represented by its pdf $p(x)$. In simple cases, this expression can be written in closed form. However, it can be difficult to compute due to the normalizing constant, and quantities that require integrating over the distribution may be computationally infeasible. For example, computing the expectation of some function $f(x)$ of the random variable requires integrating over all possible assignments to X.

$$\mathbb{E}_p(f(x)) = \int f(x)p(x)dx \tag{1}$$

Alternatively, we can approximate the distribution of X by maintaining a collection of samples $\{x_1, \ldots, x_m\}$ drawn from $p(x)$. Then, the expectation of $f(x)$ can be approximated using the sample mean.

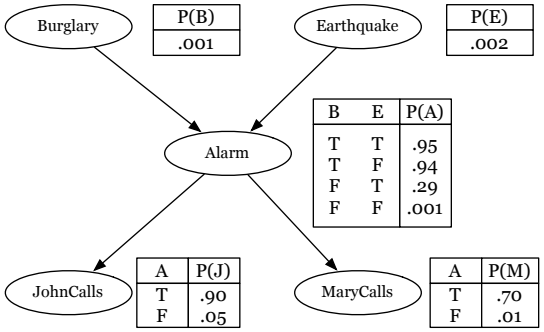
$$\mathbb{E}_p(f(x)) \approx \frac{1}{m}\sum_{i=1}^{m} f(x_i) \tag{2}$$

Monte Carlo methods work by drawing samples from the desired distribution, which can be accomplished even when it is not possible to write out the pdf. This results in a stochastic representation of the distribution in which quantities of interest may be calculated simply using sample averages. By the Law of Large Numbers, it is clear that this approximation is asymptotically exact and it is easy to apply to arbitrary models.

Despite its simplicity, there are are a number of challenges that are addressed in the remainder of the lecture, including problems associated with naive sampling in high dimensions, describing how to draw samples from a given distribution when it is not possible to do so trivially, and introducing methods that improve efficiency by making better use of samples.

2 Naive Sampling

Consider the binary, 5-dimensional probability distribution defined by the Bayesian Network in Figure 1. Due to the local conditional dependencies of the model, samples can be easily drawn by following the ordering of the edges and conditioning locally on parent samples. For example, B and E will be sampled independently, and the probability of A will be determined by the values of these samples. This process was repeated ten times for all variables and the resulting samples are shown on the right, where 1 and 0 denote true and false



E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J1
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E1	B0	A1	M1	J1
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0

Figure 1: An example Bayesian network along with samples from the distribution.

assignments respectively. These samples approximate the joint distribution of the model with probabilities determined by the frequency counts of each assignment appearing in the samples. Conditional and marginal probabilities can be similarly computed. For example, $P(J = 1|A = 0) = 1/9$ since 9 of the samples had $A = 0$ but only one of those also had $J = 1$.

Note that this sampling procedure implicitly approximates the distribution as a multinomial with 2^5 dimensions, one for every possible variable assignment, some of which may correspond to very rare events. For example, our estimate of $P(J = 1|A = 1)$ would be zero since there was only one sample with $A = 1$ but $J \neq 1$. Similarly, $P(J = 1|B = 1)$ is not even defined since there were no samples that satisfied $B = 1$ due to it's low probability of occurrence. Thus, a good approximation of high-dimensional distributions using naive sampling might require an extremely large number of samples and could become computationally infeasible.

3 Rejection Sampling

Rejection sampling is a method for sampling from a distribution $p(x) = \frac{1}{Z}p'(x)$ that is difficult to sample from, but whose unnormalized pdf $p'(x)$ is easy to evaluate. This is accomplished by first sampling from a simpler distribution $q(x)$. This sample is then accepted or rejected so that the samples follow the unknown distribution $p(x)$. Specifically, $q(x)$ must be chosen along with some constant k so that $kq(x) > p'(x)$ for all x. Then, a sample x^* from $q(x)$ is accepted with probability $p'(x^*)/kq(x^*)$. The correctness of this approach is shown in Equation 3.

$$\begin{aligned} P(x^*) &= \frac{\left[p'(x^*)/kq(x^*)\right] q(x^*)}{\int \left[p'(x)/kq(x)\right] q(x)dx} \\ &= \frac{p'(x^*)}{\int p'(x)dx} \\ &= p(x^*) \end{aligned} \tag{3}$$

While this method is guaranteed to generate samples from the desired distribution $p(x)$, it can be very inefficient, particularly in high dimensions. If the shapes of $p'(x)$ and $kq(x)$ are very different, then the probability of rejection will be high and most of the samples will be wasted. For example, consider the d-dimensional target distribution $p(x) = \mathcal{N}(x; \mu, \sigma_p^{2/d})$ and the proposal distribution $q(x) = \mathcal{N}(x; \mu, \sigma_q^{2/d})$. Note that the optimal acceptance rate can be accomplished with $k = (\sigma_q/\sigma_p)^d$. With $d = 1000$ and σ_q exceeding σ_p by only 1%, $k \approx 1/20000$ resulting in a large waste in samples. While this can by remedied by using adaptive rejection sampling in which $q(x)$ is defined by piece-wise envelope functions that are generated during sampling, this is not very generic and is only really viable in low dimensions.

4 Importance Sampling

In importance sampling, samples are independently drawn from a *proposal density* $Q(x)$, which is designed to be close to the true density $P(x)$. The contribution of each sample x to the Monte Carlo summation is weighted by an *importance* $\frac{P(x)}{Q(x)}$ so that, the estimator is unbiased. Depending on whether it is possible to compute the true density $P(x)$ or a scaled version $P*(x) = \alpha P(x)$ of the true density, we have two versions of importance sampling called *unnormalized* importance sampling and *normalized* importance sampling.

4.1 Unnormalized Importance sampling

Assume that we are equipped with a way to compute the true density $P(x)$ and the proposal density $Q(x)$ at any given point x. Further, assume that Q dominates P, that is, $Q(x) > 0$ whenever $P(x) > 0$. In other words, the support of Q contains the support of P. For an arbitrary function f, the procedure to compute $\mathbb{E}[f(x)]$ is as follows:

1. Sample $x^m \sim Q(x)$ for $m = 1, 2, \cdots, M$
2. Compute $\widehat{f} = \frac{1}{M} \sum_{m=1}^{M} f(x^m) \frac{P(x^m)}{Q(x^m)}$

It is easy to show that $\widehat{f}$ is an unbiased estimator of $\mathbb{E}_P[f(x)]$ in the measure defined by Q

$$\begin{aligned}
\mathbb{E}_Q[\widehat{f}] &= \mathbb{E}_Q[\frac{1}{M} \sum_{m=1}^{M} f(x^m) \frac{P(x^m)}{Q(x^m)}] \\
&= \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}_Q[f(x^m) \frac{P(x^m)}{Q(x^m)}] \\
&= \mathbb{E}_{x \sim Q}[f(x) \frac{P(x)}{Q(x)}] \qquad \text{as } x^m \text{ are i.i.d drawn from } Q \\
&= \int f(x) \frac{P(x)}{Q(x)} Q(x)\, dx \\
&= \int f(x) P(x)\, dx \\
&= \mathbb{E}_P[f(x)]
\end{aligned}$$

Eventhough $\widehat{f}$ is unbiased, it is in general hard to compute its variance and hence it is hard to decide when to stop sampling.

4.2 Normalized Importance sampling

In normalized importance sampling, we assume that, in addition to proposal density $Q(x)$, we are only equipped with a way to compute $P'(x) = \alpha P(x)$ for some unknown scaling factor $\alpha > 0$. The sampling procedure is similar to the above method, except for the fact that we need to eliminate the α. This is done by observing that, for $r(x) := \frac{P'(x)}{Q(x)}$

$$\mathbb{E}_Q[r(x)] = \mathbb{E}_Q[\frac{P'(x)}{Q(x)}] = \int \frac{P'(x)}{Q(x)} Q(x)\, dx = \int P'(x)\, dx = \alpha$$

For an arbitrary function f, the procedure to compute $\mathbb{E}[f(x)]$ is as follows:

1. Sample $x^m \sim Q(x)$ for $m = 1, 2, \cdots, M$
2. Compute scaling factor estimate $\widehat{\alpha} = \frac{1}{M}\sum_{m=1}^{M} r(x^m)$
3. Compute

$$\widehat{f} = \frac{1}{\widehat{\alpha}}\frac{1}{M}\sum_{m=1}^{M} f(x^m)\frac{P'(x^m)}{Q(x^m)} = \frac{\sum_{m=1}^{M} f(x^m)r(x^m)}{\sum_{m=1}^{M} r(x^m)} \tag{4}$$

The estimator $\widehat{f}$ is not unbiased. To show this, suppose we sampled just once, that is, $M = 1$. Then

$$\widehat{f} = \frac{f(x_1)r(x_1)}{r(x_1)} = f(x_1)$$
$$\mathbb{E}_Q[\widehat{f}] = \mathbb{E}_Q[f(x_1)] \neq \mathbb{E}_P[f(x_1)] \text{ in general}$$

However, in practice, the variance of the estimator in the normalized case is usually lower than that in the unnormalized case. Moreover, it is common to have $P'(x)$ available instead of $P(x)$. For example, In MRFs, it is more reasonable to assume that the unnormalized density can be computed, rather than the normalized density, as the normalizing constant Z is generally hard to compute in $P(x) = \frac{P'(x)}{Z}$. In Bayes nets, again it is more reasonable to assume that $P'(x, e) = P(x|e)P(e)$ is computable, where $P(e)$ is the scaling factor. Following is an example illustrating this idea.

Applying normalized importance sampling on Bayes Nets
The objective is to estimate the conditional probability of a variable given some evidence, which is of the form $P(X_i = x_i|e)$. For example, in Figure 3, the evidence is $e = (G = g^2, I = i^1)$ and we want to find the conditional probability of $X_i = x_i$ where X_i is one of the unobserved variables. Note that a subscript indexes a variable whereas a superscript denotes a sample number. We estimate the probability $P(X_i = x_i|e)$ by normalized importance sampling.

In importance sampling, we estimate expectations. We rewrite the probability $P(X_i = x_i|e)$ as the expectation $\mathbb{E}_{P(X_i|e)}[f(X_i)]$ where $f(X_i) := \delta(X_i = x_i)$. Construct a proposal density as follows: Clamp down the evidence nodes at the evidence values and cut off their incoming edges. Figure 3 gives an illustration of

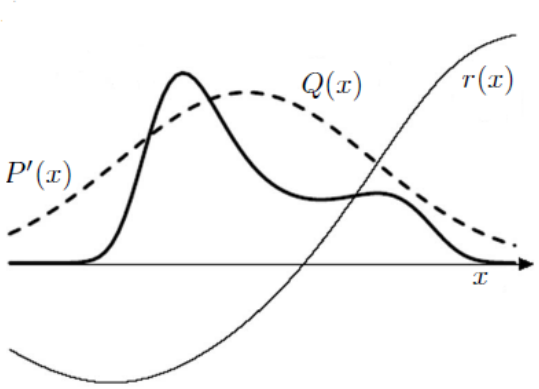


Figure 2: Normalized importance sampling

this procedure. Define the proposal density $Q(X) = P_M(X)$ to be the density of the remaining Bayes net. Define $P'(x) = P(x, e)$ so that it is proportional to $P(x|e)$.

Now we can use Equation (4), to compute the estimate

$$\widehat{P}(X_i = x_i|e) = \frac{\sum_{m=1}^{M} \delta(x_i^m = x_i) r(x^m)}{\sum_{m=1}^{M} r(x^m)}$$

where $r(x^m) = \frac{P'(x^m)}{P_M(x^m)}$.

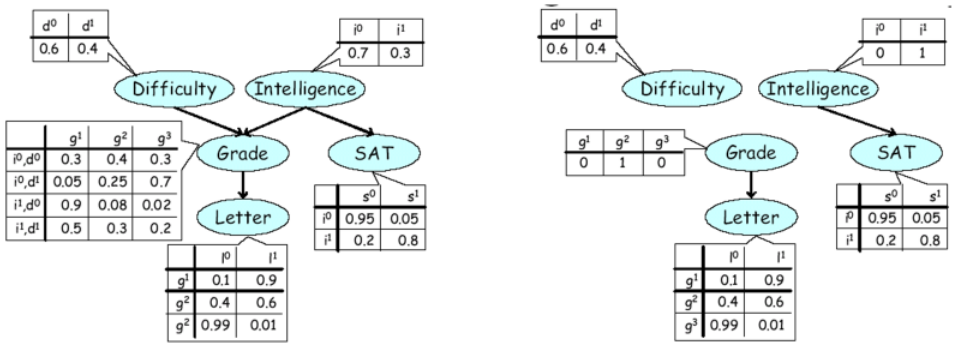


Figure 3: Illustration of how the proposal density is constructed in likelihood weighting. The evidence consists of e=(G=g2, I=i1)

In this example, we discussed how to estimate a probability $P(X_i = x_i|e)$ using normalized importance sampling, in a Bayes net. In the following section, we describe a technique to sample all the variables from a Bayes net.

4.3 Likelihood Weighting

Likelihood weighting is a special case of normalized importance sampling used to sample from a Bayes net. Suppose $\mathcal{X}$ is the set of variables in the Bayes net. Suppose the variables $E \subset \mathcal{X}$ are observed. A tuple x of size $\mathcal{X}$ is sampled as follows. For each observed variable $X_i \in E$, set x_i to the observed value of X_i. Otherwise, sample from $P(X_i|\pi_i)$ where π_i denotes the parents of X_i. The likelihood weighting algorithm carries out these steps of sampling efficiently by doing a topological sort on the variables upfront. The weight for the sample x is given by $w = \prod_{X_i \notin E} P(X_i = x_i|\pi_i)$. Each term $P(X_i = x_i|\pi_i)$ in this product can be computed when the algorithm visits X_i using the sampled assignments for its parents, as the topological visiting order makes sure that the parent variables π_i are assigned before X_i.

4.4 Weighted resampling(Sampling-Importance-Sampling)

In this procedure, the final samples are resampled according to importance weights from the samples drawn from Q. The steps are as follows:

1. Draw $x^1, x^2, \cdots, x^N$ from Q
2. Compute the weights $w^m = \frac{r^m}{\sum_{m=1}^{N} r^m}$ for $m = 1, 2, \cdots, N$
3. Resample M times from $(x^1, x^2, \cdots, x^N)$ according to the weights $(w^1, w^2, \cdots, w^N)$.

The idea is that the resampling is equivalent to a drawing from a fat tailed modification of Q.

5 Particle Filters

Particle Filters, or sequential Monte Carlo, is a method to find approximate the distribution $P(X_t|\mathbf{Y_{1:t}})$ in an state space model, SSM, such as the model in figure .

The distribution of interest is:

$$P(X_t|\mathbf{Y_{1:t}}) = P(X_t|Y_t, \mathbf{Y_{1:t}})$$

By definition of conditional probability:

$$P(X_t|Y_t, \mathbf{Y_{1:t}}) = \frac{p(X_t, Y_t|\mathbf{Y_{1:t-1}})}{p(Y_t|\mathbf{Y_{1:t-1}})}$$

The denominator can be replaced as a marginal probability:

$$P(X_t|Y_t, \mathbf{Y_{1:t}}) = \frac{p(X_t, Y_t|\mathbf{Y_{1:t-1}})}{\int p(X_t, Y_t|\mathbf{Y_{1:t-1}})dX_t}$$

By applying the chain rule,:

$$P(X_t|Y_t, \mathbf{Y_{1:t}}) = \frac{p(X_t|\mathbf{Y_{1:t-1}})p(Y_t|X_t, \mathbf{Y_{1:t-1}})}{\int p(X_t|\mathbf{Y_{1:t-1}})p(Y_t|X_t, \mathbf{Y_{1:t-1}})dX_t}$$

Because of the independencies of the SSM $p(Y_t|X_t, \mathbf{Y_{1:t-1}}) = p(Y_t|X_t)$, thus:

$$P(X_t|Y_t, \mathbf{Y_{1:t}}) = \frac{p(X_t|\mathbf{Y_{1:t-1}})p(Y_t|X_t)}{\int p(X_t|\mathbf{Y_{1:t-1}})p(Y_t|X_t)dX_t}$$

Since there is no closed from solution for the previous equation, Particle Filters uses an approximation represented by:

$$X_t^m \sim p(X_t|\mathbf{Y_{1:t-1}}), w_t^m = \frac{p(Y_t|X_t^m)}{\sum_{m=1}^{M} p(Y_t|X_t^m)}$$

where w_t^m is the weight for sample m. The initial weights, w_0^m, are initialized to be equal for each sample m.

The sampling procedure at each time t is done with a Time Update and Measurement Update.

5.1 Time Update

At this step, the distribution of interest is:

$$p(X_{t+1}|\mathbf{Y_{1:t}}) = \int p(X_{t+1}|X_t)p(X_t|\mathbf{Y_{1:t}})dX_t$$

which is approximated by:

$$p(X_{t+1}|\mathbf{Y_{1:t}}) = \sum_m w_t^m p(X_{t+1}|X_t^m)$$

Specifically, new m particles, X_{t+1}^m, are sampled from the old particles, X_t^m, using the given transition model $p(X_{t+1}|X_t)$.

5.2 Measurement Update

At this step, the weights, w_{t+1}^m, are updated for the new m particles, X_{t+1}^m, using the given emission model $p(Y_t|X_t)$. Thus, for each particle, X_{t+1}^m,

$$w_{t+1}^m = \frac{p(Y_{t+1}|X_{t+1}^m)}{\sum_{m=1}^M p(Y_{t+1}|X_{t+1}^m)}$$

The denominator only ensures that $\sum_{m=1}^M w_{t+1}^m = 1$.

Then, a new set of particles are sampled using weighted sampling:

$$X_{t+1}^m \sim p(X_{t+1}|\mathbf{Y_{1:t}}), w_{t+1}^m = \frac{p(Y_{t+1}|X_{t+1}^m)}{\sum_{m=1}^M p(Y_{t+1}|X_{t+1}^m)}$$

And this distribution was the goal of Particle Filters.

6 Particle Filters for switching State Space Models

Particles filters can be used to estimate $P(X_t|\mathbf{Y_{1:t}}, S_{1:t})$ in a SSM.

The overview of the method, at each time t, would be:

- Sample a new particle, S_t^m, from each old particle m using the transition parameters $P(S_t|S_{t-1}^m)$.
- Apply Kalman Filter to the old belief state, ($\hat{x}_{t-1|t-1}^m, P_{t-1|t-1}^m$), for each particle m.
- The result will be the new belief $P(X_t|\mathbf{Y_{1:t}}, S_{1:t})$.

More detailed information can be found in slides 19 and 20 of lecture 16.

7 Rao-Blackwellised sampling

Rao-Blackwellised sampling is a sampling method, similar to Particle Filters, for high dimensional spaces. The main difference with Particle Filters is that instead of sampling all the variables, $\mathbf{X}$, Rao-Blackwellised only samples from a subset of variables, $\mathbf{X_p}$. The m samples for the variables $\mathbf{X_p}$ are sampled using Particle Filters. Finally, there is an additional step in which the expected value of the distribution is calculated with:

$$E_{p(X|e)}[f(X)] = \frac{1}{M}\sum_m E_{p(X_d|x_p^m,e)}[f(x_p^m, X_d)]$$