10-708: Probabilistic Graphical Models 10-708, Spring 2016

15 : Approximate Inference: Monte Carlo Methods

Lecturer: Eric P. Xing *Scribes: Binxuan Huang, Yotam Hechtlinger, Fuchen Liu*

1 Introduction to Sampling Methods

1.1 General Overview

We have so far studied Variational methods to address the problem of inference. Variational methods turn the problem of inference to an optimization problem in which everything is deterministic. The drawback of Variational methods is in the fact that they only offer an approximation and not the correct answer to the problem.

In this class we study Monte Carlo sampling methods, which offers two advantages over Variational methods. The first is that the solution they provide is consistent, in the sense that it is guranteed to converge to the right solution given sufficient amount of data. The second is that sampling methods are usually easier to derive compared to Variational methods.

There are two classes of Monte Carlo methods - Stochastic sampling methods, which we have discussed in this lecture, and Markov Chain Monte Carlo (MCMC), which is a special class of Monte Carlo enabling more flexable sampling, and it will be discussed in future lectures.

1.2 Monte Carlo Sampling Methods

Suppose $x \sim p$ is high dimensional random vector from a distribution p. Often during the process of inference there is a need to compute the quantity

$$\mathbb{E}_p\left(f\left(x\right)\right) = \int f\left(x\right)p\left(x\right)dx,$$

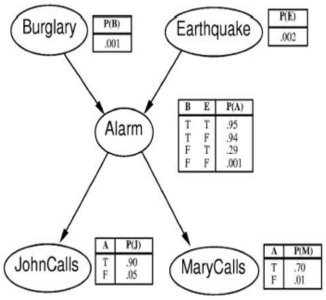
for some function f (If f is the identity this correspond to the mean of the distribution). The expected value might be hard to calculate directly, either because it is high dimensional or there is no closed form solution.

Sampling methods approximate the expected value by drawing a random sample $x^{(1)}, \ldots, x^{(n)}$ from the distribution p and use the asymptotic guaranties provided by the the Law of Large Numbers to estimate:

$$\mathbb{E}_p\left(f\left(x\right)\right) \cong \frac{1}{N}\sum_{n=1}^{N} f\left(x^{(n)}\right).$$

The challenges with sampling methods are:

- Sampling from the distribution p might not be trivial.
- How to make better use of the samples? Not all samples are equally useful.
- How many samples are required for the asymptotic to be sufficiently close?



Alarm example: (Choose the right sampling sequence)
1) Sampling:P(B)=<0.001, 0.999> suppose it is false, B0. Same for E0. P(A|B0, E0)=<0.001, 0.999> suppose it is false...
2) Frequency counting: In the samples right, **P(J|A0)=P(J,A0)/P(A0)=<1/9, 8/9>.**

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: Example of a 5 variables Bayesian network. When naively sampled, $P\left(J=1 \mid B=1\right) = P\left(J=1, B=1\right)/P\left(B=1\right)$, for example, can not be defined with the current sample, and would require significant amount of samples to be accurately estimated.

1.3 Example: Naive Sampling

It is sometimes possible to naively sample from the graphical model by drawing Bernoulli draws according to the graph distribution. Although tempting, it will not be useful under many scenarios. To demonstrate the complications that might be encountered when directly sampling the distribution, suppose we sample from the Bayesian Graph presented at Figure 1 in a naive way according to the values given in the figure. In many cases it will be interesting to calculate the conditional distribution of rare events, which is estimated by the sample counts. Accurate estimation of rare events requires large number of samples even in simple network such as the one in the figure. As the networks become more complicated, naive sampling methods becomes less and less efficient.

2 Rejection Sampling

2.1 The Rejection Sampling Process

Rejection sampling is useful in a situation where we want to sample from a distribution $\Pi\left(X\right) = \Pi'\left(X\right)/Z$, and the normalizing constant Z is unknown, thus it is hard to sample directly from Π, but it is easy to evaluate Π'.

In order to apply rejection sampling, we use a proposal distribution $Q\left(x\right)$, which we can easily sample directly from, and also assume there exist a constant K such that for all x in the support:

$$KQ\left(x\right) \geq \Pi'\left(x\right).$$

The rejection sampling algorithm will then be:

$$\begin{aligned} \text{Sample} \quad & x' \sim Q(x) \\ \text{Accept with proportion to} \quad & \frac{\Pi'(x')}{KQ(x')}. \end{aligned}$$

Figure 2 present an intuitive explanation to the process.

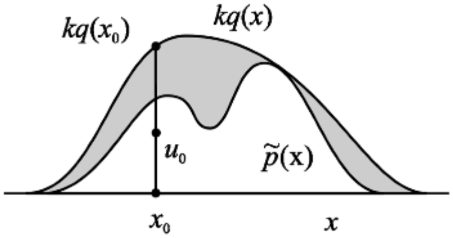


Figure 2: The Rejection sampling algorithm can be thought of a uniform sample from the area under the distribution graph. The first step is to use Q to draw x_0, and in the second step the observation is accepted with proportion equivalent to $\frac{\Pi'(x')}{KQ(x')}$. This is equivalent to draw a point u_0 uniformaly from the interval $[0, KQ(x_0)]$ and accept the observation if $u_0 \le \Pi'(x_0)$, that is in Π' graph.

The correctness of the procedure can be shown using Bayesian analysis:

$$\begin{aligned} p(x) &= \frac{[\Pi'(x)/KQ(x)] \cdot Q(x)}{\int [\Pi'(x)/KQ(x)] \cdot Q(x)\, dx} \\ &= \frac{\Pi'(x)}{\int \Pi'(x)\, dx} \\ &= \frac{\Pi'(x)}{Z} = \Pi(x). \end{aligned}$$

2.2 High Dimensional Drawback

A crucial step in the process is the selection of Q, and K. The number of samples accepted is equal to the ratio between the areas of the distributions. It is therefore important to control the rejection area to be as small as possible. In high dimensions this becomes a major drawback due to the curse of dimension, effectively limiting the method to low dimensions. Figure 3 further explain the problem using an example.

3 Important sampling

3.1 Unnormalized important sampling

The finite sum approximation to the expectation depends on being able to draw samples from the distribution $P(x)$. However, it could be impractical to sample directly from $P(x)$. Suppose we have a proposal distribution

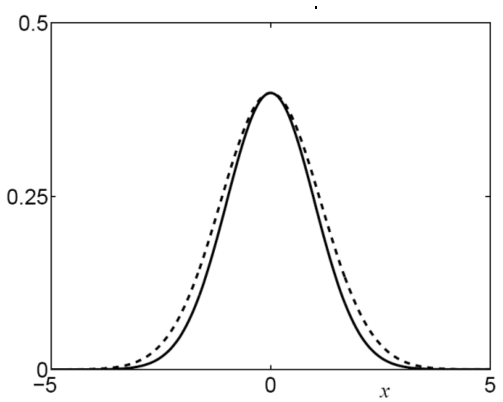


Figure 3: As a thought experiment, suppose we sample from $P = \mathcal{N}\left(\mu, \sigma_p^{2/d}\right)$ with the proposal distribution $Q = \mathcal{N}\left(\mu, \sigma_q^{2/d}\right)$, where σ_q exceed σ_p by 1%. The figure demonstrate the densities when $d = 1$. This example is just for instructional purposes - since we known how to sample one Guassian, we could sample the other also. When the dimension $d = 1000$, the optimal acceptance rate is $K = \left(\frac{\sigma_q}{\sigma_p}\right)^d \approx 1/20,000$. It follows that only 1 sample out of $20,000$ will be accepted.

$Q(x)$ which can be simpler sampled from and Q dominate P (i.e. $Q(x) > 0$ whenever $P(x) > 0$), then we can sample from Q and reweight each sample by importance $w(x) = \frac{P(x)}{Q(x)}$.

The procedure of unnormalized important sampling is as follows:

1 Sample x^m from Q for m = 1,2, . . . , M

2 Compute $\hat{E}(f) = \frac{1}{M}\sum_{m=1}^{M} f(x^m)\frac{P(x^m)}{Q(x^m)}$

It is because

$$\begin{aligned}
E_P(f) &= \int f(x)P(x)dx \\
&= \int f(x)\frac{P(x)}{Q(x)}Q(x)dx \\
&\simeq \frac{1}{M}\sum_{m=1}^{M} f(x^m)\frac{P(x^m)}{Q(x^m)} \\
&= \frac{1}{M}\sum_{m=1}^{M} f(x^m)w(x^m)
\end{aligned}$$

One advantage of unnormalized important sampling beyond rejection sampling is that it uses all samples and avoids the waste. However, we need to be able to compute the exact value of P(x) (i.e. P need to be close form) in unnormalized important sampling.

3.2 Normalized important sampling

But sometimes, we can only evaluate $P'(x) = \alpha P(x)$ (e.g. for an MRF) with an unknown scaling factor $\alpha > 0$. In this case, we can get around the nasty normalization constant α as follows: let ratio $r(x) = \frac{P'(x)}{Q(x)}$, then

$$E_Q[r(x)] = \int \frac{P'(x)}{Q(x)} Q(x)dx = \int P'(x)dx = \alpha$$

Now

$$\begin{aligned} E_P[f(x)] &= \int f(x)P(x)dx \\ &= \frac{1}{\alpha} \int f(x) \frac{P'(x)}{Q(x)} Q(x)dx \\ &= \frac{\int f(x)r(x)Q(x)}{\int r(x)Q(x)} \\ &\simeq \frac{\sum_{m=1}^{M} f(x^m)r(x^m)}{\sum_{m=1}^{M} r(x^m)} \qquad x^m \sim Q(X) \\ &= \sum_{m=1}^{M} f(x^m)w^m \qquad w^m = \frac{r(x^m)}{\sum_{l=1}^{M} r(x^l)} \end{aligned}$$

Then the procedure of normalized importance sampling is:

1 Sample $x^m \sim Q(x)$ for m = 1, 2,. . . , M

2 Compute scaling factor $\hat{\alpha} = \frac{1}{M} \sum_{m=1}^{M} r(x^m)$

3 Compute $\hat{E}_P(f) = \frac{\sum_{m=1}^{M} f(x^m)r(x^m)}{\sum_{m=1}^{M} r(x^m)}$

Normalized importance sampling allows us to use a scaled approximate of P(x) but it is biased. Notice that for unnormalized importance sampling:

$$\begin{aligned} E_Q[f(X)w(X)] &= \int f(x)w(x)Q(x)dx = \int f(x) \frac{P(x)}{Q(x)} Q(x)dx \\ &= \int f(x)P(x)dx = E_p(f) \end{aligned}$$

So unnormalized importance sampling is unbiased. But for normalized importance sampling, e.g. M=1:

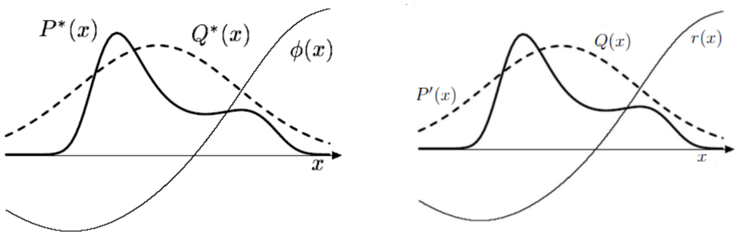


Figure 4: Examples of weight functions in unnormalized and normalized importance sampling

$$E_Q[\frac{f(x^1)r(x^1)}{r(x^1)}] = \int f(x)Q(x)dx \neq E_P(f) \text{in general}$$

However, in practice, the variance of the estimator in the normalized case is usually lower than in the unnormalized case. Also, it is common that we can evaluate $P'(X)$ but not $P(x)$. For example in Bayes nets, it is more reasonable to assume that $P'(x|e) = P(x|e)P(e)$ is computable, where $P(e)$ is the scaling factor. And In MRF, $P(x) = \frac{P'(x)}{Z}$ and Z is generally hard to compute.

3.3 Normalized sampling method to BN

We now apply normalized importance sampling to a Bayes net. The objective is to estimate the conditional probability of a variable given some evidence :$P(X_i = x_i|e)$. We rewrite the probability $P(X_i = x_i|e)$ as the expectation $E_{P(X_I|e)}[f(X_i)]$ where $f(X) := \delta(X_i = x_i)$. Then we get the proposal distribution from the multilated BN where we clamp evidence nodes and cut off the incoming arcs. Figure 2 gives an illustration of this procedure. Define $Q = P_M$, $P'(x) = P(x,e)$, then we get

$$\hat{P}(X_i = x_i|e) = \frac{\sum_{m=1}^{M} w(x^m)\delta(x^m = x_i)}{\sum_{m=1}^{M} w(x^m)}$$
$$\text{where} w(x^m) = \frac{P'(x^m, e)}{P_M(x^m)}$$

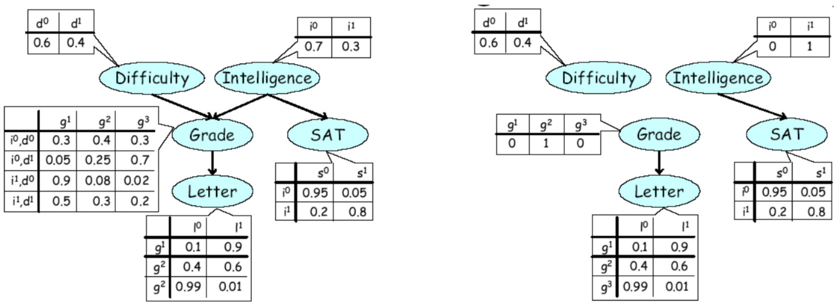


Figure 5: Illustration of how the proposal density is constructed in likelihood weighting. The evidence consists of $e = (G = g2, I = i1)$

Likelihood weighting is a special case of normalized importance sampling used to sample from a BN. This part is skipped by Eric. The pseudo code and efficiency of likelihood weighting method could be found in lecture slides.

3.4 Weighted resampling

The performance of Importance sampling depends on how well Q matches P. Like figure 3 shows, if $P(x)f(x)$ is strongly varying and has a significant proportion of its mass concentrated in a small region, $r(x)$ will be dominated by a few samples. And if the high-prob mass region of Q falls into the low-prob mass region of P, the there will be a lot of samples have less weight, like the star points showed in figure 3. And in the high-prob region of P, there may be few or no samples. The problem is that there is no way to diagnose it in a importance sampling procedure. We need to draw more samples to see whether it changes the estimator, but the variance of $r^m = \frac{P(x^m)}{Q(x^m)}$ can be small even if the samples come from low-prob region of P and potentially erroneous.

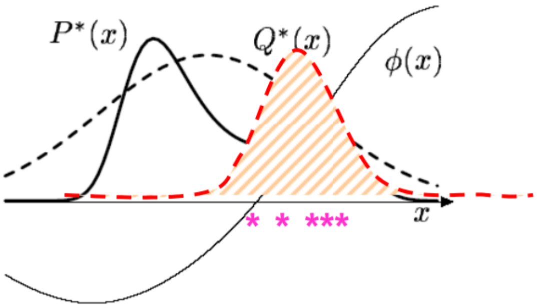


Figure 6: An examples of the problem of importance sampling: the high-prob mass region of Q falls into the low-prob mass region of P

There are 2 possible solutions for this problem. Firstly we can use a heavy tail Q in order to make sure there are enough samples in all of the region. The second solution is to apply a weighted resampling method. Sampling important resampling (SIR) if one of the resampling method based on weight of the samples:

1 Draw N samples from Q: $X_1, \cdots, X_N$

2 Constructing weights: $w(x_1), \cdots, w(x_N)$, where $w(x^m) = \frac{P(x^m)/Q(x^m)}{\sum_{l=1}^{M} P(x^l)/Q(x^l)} = \frac{r(x^m)}{\sum_{l=1}^{M} r(x^l)}$

3 Sub-sampling x from $X_1, \cdots, X_N$ w.p $w(x_1), \cdots, w(x_N)$

Another way to do it particular filtering, which will be showed in the next section.

4 Particle Filter

Particle Filter is a sampling method used to estimate the posterior distribution $P(X_t|Y_{1:t})$ in a state space model (SSM) with known transition probability distribution $P(X_{t+1}|X_t)$ and emission probability $P(Y_t|X_t)$. In the previous lectures, we have studied some algorithms like Kalman Filtering to solve SSM. However, Kalman Filtering assumes that the transition probabilities are Gaussian distributions, which is a big constraint. That's why we need Particle Filter.

Particle Filter can be viewed as an online algorithm. At time $t+1$, a new observation Y_{t+1} is recieved as input, and the algorithm output is $P(X_{t+1}|Y_{1:t+1})$ based on previous estimation $P(X_t|Y_{1:t})$.

Notice we assume have already have $P(X_t|Y_{1:t})$ which can be represented by

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

where $\{X_t^m\}$ are M samples we drew from the prediction at time $t-1$, $P(X_t|Y_{1:t-1})$, and $\{w_t^m\}$ are the

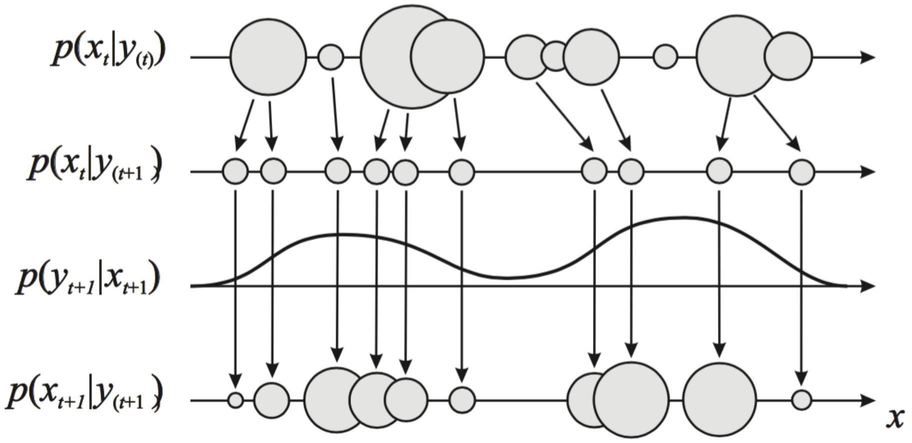


Figure 7: Schematic illustration of the operation of the particle filter. At time step t the posterior $p(x_t|y_t^m)$ is represented as a mixture distribution, shown schematically as circles whose sizes are proportional to the weights $w_t^{\cdot}$. A set of M samples is then drawn from this distribution, and the new weights w_{t+1}^m evaluated using $p(y_{t+1}|x_{t+1}^m)$.

corresponding weights from samples. This representation suffice because:

$$\begin{aligned} P(X_t|Y_{1:t}) = P(X_t|Y_t, Y_{1:t-1}) &= \frac{P(X_t|Y_{1:t-1})P(Y_t|X_t)}{\int P(X_t|Y_{1:t-1})P(Y_t|X_t)dX_t} \\ &= P(X_t|Y_{1:t-1})\frac{P(Y_t|X_t)}{\int P(X_t|Y_{1:t-1})P(Y_t|X_t)dX_t}, \end{aligned}$$

where the right part of right equation above is just the weight approximated by M samples.

Next, at next time step $t+1$, we will calculate $P(X_{t+1}|Y_{t+1})$ using two updates: Time Update and Measurement Update.

In Time Update, we will draw M new samples $\{X_{t+1}^m\}$ from $P(X_{t+1}|Y_{1:t})$, which is given by

$$P(X_{t+1}|Y_{1:t}) = \int P(X_{t+1}|X_t)P(X_t|Y_{1:t})dX_t = \sum_{m=1}^{M} w_t^m P(X_{t+1}|Y_{1:t}).$$

Here we can see that $P(X_{t+1}|Y_{1:t})$ is a mixture model with M weights and M known component models given by the transition probability $P(X_{t+1}|X_{1:t})$.

At Measurement Update step, we will update the weight $\{w_{t+1}^m\}$ again by

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

The desired posterior probability at time $t+1$, $P(X_{t+1}|Y_{1:t+1})$, follows from the two step updates because it can be represented in the same manner by:

$$\left\{ X_{t+1}^m \sim P(X_{t+1}|Y_{1:t}),\ w_{t+1}^m = \frac{P(Y_{t+1}|X_{t+1}^m)}{\sum_{m=1}^M P(Y_{t+1}^m|X_{t+1}^m)} \right\}.$$

5 Rao-Blackwellised sampling

Sampling in a high dimensional probability spaces can sometimes result with high variance in the estimate. In the class, the lecturer gave an example of multivariate Gaussian distribution. In that case, with high dimensition n, making small changes to the standard deviation σ in every dimension will cause the estimation to change a lot.

To avoid this drawback, we can utilize the property of total variance:

$$var[\tau(x_p, x_d)] = var[E[\tau(x_p, x_d)|x_p]] + E[var[\tau(x_p, x_d)|x_p]].$$

From the equation above, we can see $var[E[\tau(x_p, x_d)|x_p]] \leq var[\tau(x_p, x_d)]$. There is a simple proof at: `https://en.wikipedia.org/wiki/Law_of_total_variance`

Hence when computing $E_{p(X|e)}[f(X_p, X_d)]$, instead of sampling x_p, x_d directly from probability $p(x_p, x_d|e)$ just like $E_{p(X|e}[f(X_p, X_d)] = \frac{1}{M}\sum_m f(x_p^m, x_d^m)$, we can first sample variables X_p and then compute the expected value of X_d conditioned on X_p.

$$\begin{aligned} E_{p(X|e)}[f(X_p, X_d)] &= \int p(x_p, x_d|e) f(x_p, x_d) dx_p dx_d \\ &= \int_{x_p} p(x_p|e)[\int_{x_d} p(x_d|x_p, e) f(x_p, x_d) dx_d] dx_p \\ &= \int_{x_p} p(x_p|e) E_{p(X_d|x_p, e)}[f(x_p, X_d)] dx_p \\ &= \frac{1}{M} \sum_m E_{p(X_d|x_p^m, e)}[f(x_p^m, X_d)], \qquad x_p^m \sim p(x_p|e). \end{aligned}$$

Basically, this sampling process transforms sampling in spaces with high dimension $p + d$ into spaces with low dimension p.