10-708: Probabilistic Graphical Models 10-708, Spring 2017

Lecture 15: Approximate Inference: Monte Carlo methods

Lecturer: Eric P. Xing *Name: Yuan Liu(yuanl4), Yikang Li(yikangl)*

1 Introduction

We have already learned some exact inference methods in the previous lectures, such as the elimination algorithm, message-passing algorithm and the junction tree algorithms. However, there are many probabilistic models of practical interest for which exact inference is intractable. One important class of inference algorithms is based on deterministic approximation schemes and includes methods such as variational methods. This lecture will include an alternative very general and widely used framework for approximate inference based on stochastic simulation through sampling, also known as the Monte Carlo methods.

The purpose of the Monte Carlo method is to find the expectation of some function $f(x)$ with respect to a probability distribution $p(x)$. Here, the components of x might comprise of discrete or continuous variables, or some combination of the two.

$$\langle f \rangle = \int f(x)p(x)dx$$

The general idea behind sampling methods is to obtain a set of examples $x^{(t)}$(where $t = 1, ..., N$) drawn from the distribution $p(x)$. This allows the expectation (1) to be approximated by a finite sum

$$\hat{f} = \frac{1}{N}\sum_{t=1}^{N} f(x^{(t)})$$

If the samples $x^{(t)}$ is i.i.d, it is clear that $\langle \hat{f} \rangle = \langle f \rangle$, and the variance of the estimator is $\langle (f - \langle f \rangle)^2 \rangle / N$. Monte Carlo methods work by drawing samples from the desired distribution, which can be accomplished even when it it not possible to write out the pdf.

Despite its simplicity, there are still a number of challenges:

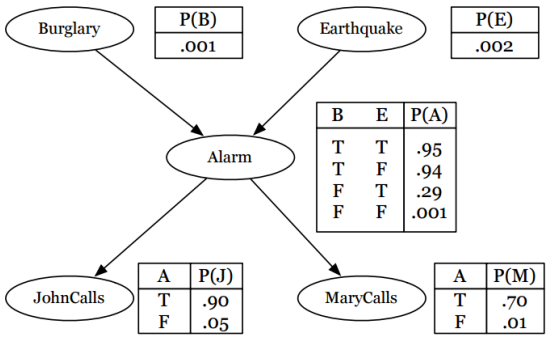
- How to draw samples from a given distribution?
- How to make better use of the samples?
- How to know we have sampled enough?

2 Naive sampling

Here, we are interested in the case in which the distribution $p(x)$ is specified in terms of a directed graph. The joint distribution of a directed graph can be written as:

$$p(x) = \prod_{i=1}^{d} p(x_i | x_{\pi(x_i)})$$

We can assume $\{x_1, ..., x_n\}$ is arranged in the order that $\pi(x_i) \subset \{1, ..., i-1\}$. The meaning of this order is the index of x_i's ancestors is less than i. Then to obtain a sample from the joint distribution we make one pass through the set of variables in the order $x_1, ..., x_d$ sampling from the conditional distribution $p(x_i|x_{\pi(x_i)})$ This is always possible since at each step all of the parent values will have been instantiated. After one pass through the graph we will have obtained a sample from the joint distribution.



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

For example, consider the binary, 5-dimensional probability distribution defined by the Bayesian Network in Figure **??**. First we well arrange the variables in the following order $\{x_B, x_E, x_A, x_J, x_M\}$. Then we need to make one pass through this variable list. First, B and E will be sampled independently, at this time the distribution of A will be determined by the values of these samples. Second, we can sample B from this distribution. So on and so on.

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 its 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

The rejection sampling framework allows us to sample from relatively complex distributions. Suppose we wish to sample from a distribution $p(x)$ which is not one of the simple, standard distributions considered so far, and that sampling directly from $p(x)$ is difficult. Furthermore suppose, as is often the case, that we can only express $p(x)$ in the following form:

$$p(x) = \frac{1}{Z}\tilde{p}(x)$$

In order to apply rejection sampling we need some simpler distribution $q(x)$, from which we can readily draw samples. We next need to find a constant k whose values is chosen such that $kq(x) \geq \tilde{p}(x)$ for all values of x. Then, draw sample x_0 from $q(x)$ and accept it with probability $\tilde{p}(x_0)/kq(x_0)$. The correctness of rejection

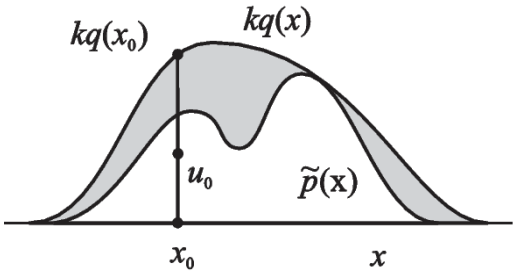


Figure 2: Rejection Sampling: samples are drawn from $q(x)$ and rejected if fall in grey area

sampling is shown in the following equation:

$$\begin{aligned} p(x_0) &= \frac{[\tilde{p}(x_0)/kq(x_0)]q(x_0)}{\int[\tilde{p}(x)/kq(x)]q(x)dx} \\ &= \frac{\tilde{p}(x_0)}{\int \tilde{p}(x)dx} \\ &= p(x_0) \end{aligned}$$

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 $\tilde{p}(x)$ 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 $p(x) = N(x; \mu, \sigma_p^{2/d})$ and the proposal distribution $q(x) = N(x; \mu, \sigma_q^{2/d})$. 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.

4 Importance Sampling

4.1 Unnormalized Importance Sampling

We also assume that sampling directly from $p(x)$ is difficult, and drawing samples from $q(x)$ is readily. Then the following equations can be shown:

$$\begin{aligned} \langle f \rangle &= \int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx \\ &\approx \frac{1}{N}\sum_{t=1}^{N} f(x^{(t)})\frac{p(x^{(t)})}{q(x^{(t)})} \quad \text{where } x^{(t)} \sim q(x) \\ &= \frac{1}{M}\sum_{t=1}^{N} f(x^{(t)})w^{(t)} \end{aligned}$$

The quantities $w^{(t)} = \frac{p(x^{(t)})}{q(x^{(t)})}$ are known as importance weights, and they correct the bias introduced by sampling from the wrong distribution. Note that, unlike rejection sampling, all of the samples generated are retained.

As with rejection sampling, the success of the importance sampling approach depends crucially on how well the sampling distribution $q(x)$ matches the desired distribution $p(x)$. If, as is often the case, $p(x)f(x)$ is

strongly varying and has a significant proportion of its mass concentrated over relatively small regions of x space, then the set of importance weights $\{w^{(t)}\}$ may be dominated by a few weights have large values, which the remaining weights being relatively insignificant. Thus the effective sample size can be much smaller than the apparent sample size N.

4.2 Normalized Importance Sampling

4.2.1 Procedures of normalized importance sampling

In unnormalized importance sampling, we have to calculate the value of $P(x^m)$. However, sometimes it is difficult to compute $P(x)$ directly. (For example, in Markov Random Fields.) Suppose we can only evaluate $P'(x) = \alpha P(x)$.
Let's define the ratio as $r(x) = \frac{P'(x)}{Q(x)}$, then we have:

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

Thus, we can compute the expectation of $f(x)$ as:

$$\begin{aligned}
E_P(f(x)) &= \int f(x)P(x)dx \\
&= \frac{1}{\alpha}\int f(x)\frac{P'(x)}{Q(x)}Q(x)dx \\
&= \frac{\int f(x)r(x)Q(x)dx}{\int r(x)Q(x)dx} \\
&\approx \frac{\sum_m f(x^m)r^m}{\sum_m r^m} \qquad \text{where } x^m \sim Q(x) \\
&= \sum_m f(x^m)w^m \qquad \text{where } w^m = \frac{r^m}{\sum_m r^m}
\end{aligned}$$

To summarize, we draw M samples from Q, compute the normalized ratio as weights of samples to compute $E_P(f(x))$.

4.2.2 Comparison of normalized/unnormalized importance sampling

First, we should know that unnormalized importance sampling is unbiased and normalized importance is biased.
For unnormalized importance sampling:

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

For normalized importance sampling, we consider a simple case when $M = 1$:

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

Although the normalized one is biased in estimation, its variance is often lower than unnormalized one. In other words, we can get a more robust estimator if we choose normalized importance sampling. What's more, in practice it is common that we can only evaluate $P'(x)$ but not $P(x)$. (In bayesian network, we can easily evaluate $P'(x, e) = P(x|e)P(e)$. In MRF, we can easily evaluate $P'(x) = P(x) * Z$.)

4.2.3 Application in graphical models: likelihood weighted sampling

Consider a Bayesian network and we would like to estimate the conditional probability of a variable given some evidence: $P(X_i = x_i|e)$. We can set $f(X_i) = \delta(X_i = x_i)$ and rewrite the probability $P(X_i = x_i|e)$ as $E_P[f(X_i)]$. The proposal Q is gotten from the mutilated BN where we clamp evidence nodes and cut their incoming arcs. We call this $Q = P_M$ and set the unnormalized posterior as $P'(x) = P(x, e)$.
Thus, we can get:

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

The procedure of this algorithm is simple: Suppose we have a topological ordering for the variables $X_1, \ldots, X_n$ of graph G. For each variable X_i in turn, we check whether the variable is in the evidence set. If it's true, then we set them to its instantiated value $x_i = e(X_i)$, else we sampled x_i from the conditional distribution $P(X_i|u_i)$ in which the conditional variables are set to their currently sampled values.
The efficiency of likelihood weighting depends on how close the proposal is to the target P. If the evidence is at the roots, then we have exactly $Q = P(X|e)$ and all the weights are equal to 1. If the evidence is at the leaves, then Q is the prior distribution and many samples may get small weights.

4.3 Weighted Resampling

4.3.1 Pitfall of importance sampling

Just the same of rejection sampling, the success of the importance sampling depends crucially on how well the sampling distribution Q matches the true distribution P. However, we cannot design a Q that is close to P, because we have no idea what P will look like. If $P(x)f(x)$ is strongly varying and has significant proportion of its mass concentrated in a small regions of x space, then importance weights r_m may be dominated by a few samples that has large weights and the remaining large amounts of samples being insignificant because of small weights, just as Figure **??** shows.
This situation will lead to few or even none effective samples, and the apparent variance of $r_m f(x_m)$ may be small even though the expectation is totally wrong. What's make it worse is that we cannot detect it and goes to error without even notice it.

4.3.2 Weighted resampling

The possible solution is:

1. Use heavy tail Q so that there can be enough samples in all of the regions, especially where mass of P concentrate.

2. Apply weighted resampling in a two stage schema:

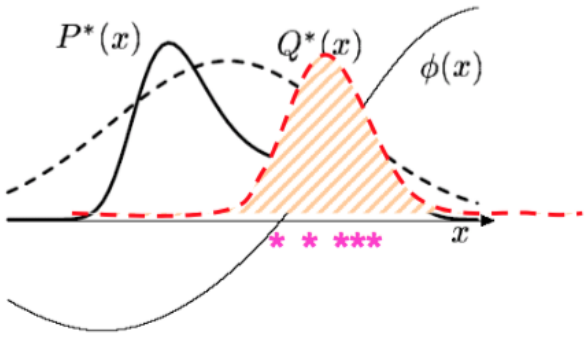


Figure 3: Possible problem of importance sampling

In the first stage, we draw N samples from Q: $X_1, \ldots, X_N$ and compute the weights $w_1, \ldots, w_N$ as:

$$w^n = \frac{P(x^n)/Q(x^n)}{\sum_l P(x^l)/Q(x^l)} = \frac{r^n}{\sum_l r^l}$$

In the second stage, we draw M samples from the discrete distribution $\{X_1, \ldots, X_N\}$ with probabilities given by the weights $w_1, \ldots, w_N$. It can be proven that the resulting subsamples are approximately distributed according to P and converges to P as sampling number $N \to \infty$.

5 Particle Filter

Particle filters is a sequential Monte Carlo algorithm which can be seen as a special case of re-sampling. Suppose we have a model consist of observations y_t arriving sequentially and hidden variables x_t and we want to estimate the posterior distribution of the hidden variables after the coming of new observation. If we choose more complex conditional distributions for $p(y_t|x_t)$ other than discrete distributions or Gaussian distribution, then the posterior distribution is computationally intractable and we therefore use sampling methods to solve it.

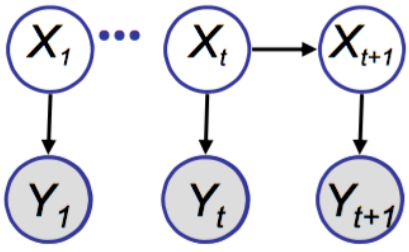


Figure 4: Sequence of observations explained in a hidden Markov chain of latent variables

Suppose we are given the observed values $Y_t = (y_1, \ldots, y_t)$ and we want to draw M samples from the

posterior distribution $p(x_t|Y_t)$ in order to evaluate the expectation of $f(x)$. Thus, we have:

$$\begin{aligned} E_p[f(x)] &= \int f(x_t)p(x_t|Y_t)dx_t \\ &= \int f(x_t)p(x_t|y_t, Y_{t-1}) \\ &= \frac{\int f(x_t)p(y_t|x_t)p(x_t|Y_{t-1})dx_t}{\int p(y_t|x_t)p(x_t|Y_{t-1})dx_t} \\ &\simeq \sum_{m=1}^{M} w_t^m f(x_t^m) \end{aligned}$$

Which means we can represent $p(x_t|Y_t)$ by M samples $\{x_t^m\}$ we draw from $p(x_t|Y_{t-1})$ and corresponding weights $\{w_t^m\}$ of these samples. Now we will use the samples and weights we get at time step t to find the samples and updates the weights at time step $t+1$. It includes two stages: time update and measurement update.
In time update, we draw samples from the distribution $p(x_{t+1}|Y_t)$:

$$\begin{aligned} p(x_{t+1}|Y_t) &= \int p(x_{t+1}|x_t)p(x_t|Y_t)dx_t \\ &= \frac{\int p(x_{t+1}|x_t)p(y_t|x_t)p(x_t|Y_{t-1})dx_t}{\int p(y_t|x_t)p(x_t|Y_{t-1})dx_t} \\ &= \sum_m w_t^m p(x_{t+1}|x_t^m) \end{aligned}$$

Actually, it can be seen as a mixture model and samples can be drawn by first choose a component m with probability w_t^m then draw a sample from the corresponding component. At measurement update stage, we update the weight w_{t+1}^m similarly and the posterior probability at time step $t+1$ can be represented in the same form:

$$\left\{ x_{t+1}^m \sim p(x_{t+1}|Y_t), \;\; w_{t+1}^m = \frac{p(Y_{t+1}|x_{t+1}^m)}{\sum_{l=1}^{M} p(Y_{t+1}|X_{t+1}^l)} \right\}$$

The step of particle filter algorithm can be illustrated as Figure **??**. We can also apply particle filter to switching SSM. Please see the slides for details.

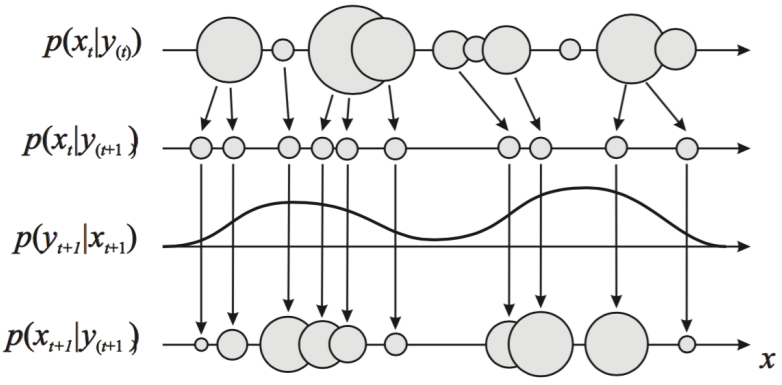


Figure 5: Schematic illustration of the particle filter

6 Rao-Blackwellised Sampling

As shown in previous slides, sampling in high dimensional distribution space can lead to high variance of the estimation. Thus, we can avoid this drawback by doing integral for variables that are easy to compute and leave others for sampling. The theory behind this is the property of total variance:

$$var[\tau(X_p, X_d)] = var[E[\tau(X_p, X_d)|X_p]] + E[var[\tau(X_p, X_d)|X_p]]$$

We can perform the Rap-Blackwellised sampling by first sample variables X_p and then compute the expected value of X_d conditioned on X_p:

$$\begin{aligned} E_{p(X|e}[f(X_p, X_d)] &= \int p(X_p, X_d) f(X_p, X_d) dX_p dX_d \\ &= \int p(X_p|e) \Big(\int p(X_d|X_p, e) f(X_p, X_d) dX_d \Big) dX_p \\ &= \int p(X_p|e) E_{p(X_d|X_p,e)}[f(X_p, X_d)] dX_p \\ &= \frac{1}{M} \sum_{m=1}^{M} E_{p(X_d|X_p,e)}[f(X_p, X_d)] \qquad X_p \sim p(X_p|e) \end{aligned}$$

By subsampling in a relative lower dimension space, we can get a lower variance estimator.