IMPORTANCE SAMPLING

Monday, November 17, 2008

Rice University

STAT 631 / ELEC 639: **Graphical Models**

Instructor:
Dr. Volkan CEVHER

Scribe:
Ryan E. Guerra (*war@rice.edu*)
Tahira N. Saleem (*ts4@rice.edu*)
Terrance D. Savitsky (*tds1@rice.edu*)

1 Motivation

In machine learning and statistics, we're often tasked with computing the expected value of a function $f(\boldsymbol{x})$ with respect to a probability distribution $p(\boldsymbol{x})$, where $\boldsymbol{x} \in \mathbb{R}^n$. In many cases, the canonical technique of evaluating the integral

$$\int_{\boldsymbol{x}} f(\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x},$$

is intractable due to the nature or complexity of $p(\boldsymbol{x})$.

On one hand, if a cumulative distribution function is non-decreasing and easily invertible then we can draw samples from its distribution by using *inverse transform sampling* [1] where we map $i.i.d.$ samples from $U[0,1]$ through the inverse CDF $P^{-1}(\mathbf{y})$. If one wishes to draw samples from a multivariate Gaussian distribution, then the well-known *Box-Muller* method [1] will suffice. On the other hand, many distributions are difficult or impossible to invert, and in some cases a closed-form representation might not exist or be computationally intractable to obtain. This is a problem since finding expected values of functions is often a step in larger engineering problems or algorithms.

Importance sampling is a discrete method for approximating $I[f] = \mathbb{E}_p[f(\boldsymbol{x})]$ by replacing $p(\boldsymbol{x})$ with a similar, but easily sampled, distribution $q(\boldsymbol{x})$ and then correcting for the error introduced by making this switch. It is generally cited as a Monte Carlo variance reduction technique in that it provides a framework for reducing the computational complexity of computing the expectation $I[f]$ while directly relating the complexity to decreased simulation variance. In this paper, we will try to develop the theory of importance sampling and highlight certain important properties to consider when utilizing the technique. In the final section we present a MATLAB simulation with discussion.

2 Importance Sampling

Consider a set of samples $\{\boldsymbol{x}^{(i)}\}$ generated from $p(\boldsymbol{x})$, a given probability distribution. Then the form of the expectation of $f(\boldsymbol{x})$ under p in (1) can be approximated by the average of $f(\boldsymbol{x})$ evaluated at those samples (2). For the sake of notation, we will be using $I[f]$ to represent the expectation throughout this discussion.

$$I[f] = \mathbb{E}_p[f(\boldsymbol{x})] = \int f(\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{1}$$

$$\simeq \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}^{(i)}) \tag{2}$$

As in the previous section, we are operating under the assumption that while we can evaluate the value of $p(\boldsymbol{x})$ at a given $\boldsymbol{x}$, we cannot easily draw the samples from the distribution needed for our estimate. To deal

with this, we introduce another distribution $q(\boldsymbol{x})$ called the *sampling distribution* that we will draw samples $\{\boldsymbol{x}^{(i)}\}$ from instead.

$$\begin{aligned} I[f] &= \int f(\boldsymbol{x}) \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} q(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &\simeq \frac{1}{N} \sum_{i=1}^{N} \frac{p(\boldsymbol{x}^{(i)})}{q(\boldsymbol{x}^{(i)})} f(\boldsymbol{x}^{(i)}) \\ \hat{I}[f] &= \frac{1}{N} \sum_{i=1}^{N} \tilde{w} f(\boldsymbol{x}^{(i)}), \;\; \tilde{w} = \frac{p(\boldsymbol{x}^{(i)})}{q(\boldsymbol{x}^{(i)})} \end{aligned} \tag{3}$$

The basic idea of importance sampling is to draw from a similar distribution other than $p(\boldsymbol{x})$, say $q(\boldsymbol{x})$, and then modify the resulting equation to correct the bias introduced by sampling from the wrong distribution. In (3) we can see that the bias correction, or the *importance weight* $\tilde{w}$ can be determined exactly for a given $\boldsymbol{x}^{(i)}$ since we assumed that we could evaluate $p(\boldsymbol{x})$ at a point.

In practice, our actual $p(\boldsymbol{x})$ or $q(\boldsymbol{x})$ will often be unnormalized [1], in other words $p(\boldsymbol{x}) = \frac{1}{Z_p}\tilde{p}(\boldsymbol{x})$ and $q(\boldsymbol{x}) = \frac{1}{Z_q}\tilde{q}(\boldsymbol{x})$ so that

$$\tilde{w} = \frac{p(\boldsymbol{x}^{(i)})}{q(\boldsymbol{x}^{(i)})} = \frac{Z_q}{Z_p} \frac{\tilde{p}(\boldsymbol{x}^{(i)})}{\tilde{q}(\boldsymbol{x}^{(i)})}$$

where

$$\begin{aligned} \frac{Z_p}{Z_q} = \frac{\int \tilde{p}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}}{Z_q} &= \int \frac{\tilde{p}(\boldsymbol{x})}{\tilde{q}(\boldsymbol{x})} q(\boldsymbol{x}) \\ &\simeq \frac{1}{N} \sum_{i=1}^{N} \frac{\tilde{p}(\boldsymbol{x})}{\tilde{q}(\boldsymbol{x})}, \end{aligned} \tag{4}$$

and we have used the set of our samples from $q(\boldsymbol{x})$ in order to estimate the normalization factor in (4). If we redefine our importance weight as w_i we emerge with two forms of our approximation, where (6) is simply the approximation derived in (3)

$$\hat{I}_1[f] = \sum_{i=1}^{N} w_i f(\boldsymbol{x}^{(i)}), \;\; w_i = \frac{\tilde{p}(x^{(i)})/\tilde{q}(x^{(i)})}{\sum_{k=1}^{N} \tilde{p}(x^{(k)})/\tilde{q}(x^{(k)})} \tag{5}$$

$$\hat{I}_2[f] = \frac{1}{N} \sum_{i=1}^{N} \frac{p(\boldsymbol{x}^{(i)})}{q(\boldsymbol{x}^{(i)})} f(\boldsymbol{x}^{(i)}) \tag{6}$$

It is important to note that (5) is an approximation where the probability distribution of $q(\boldsymbol{x})$ has been approximated *twice*: once for the main term and once to normalize the importance weights in (4). This approximation is necessary when there is no guarantee that either $p(\boldsymbol{x})$ or $q(\boldsymbol{x})$ are normalized. The second form of the approximation (6) assumes that both distributions are normalized and generally will not be equal to the original except for in the expectation: $\mathbb{E}\left\{ I[f] - \hat{I}_i[f] \right\} = 0$.

3 Properties

There is one particular property to keep in mind when using the importance sampling method. Clearly, as the number of samples is increased the variance of our estimation $\hat{I}[f]$ will decrease. Since $\mathbb{E}\{\hat{I}[f]\} = I[f]$, this is quantified in [2] as the *asymptotic variance* σ_f^2,

$$\mathrm{var}(\hat{I}[f]) = \frac{\sigma_f^2}{N} = \frac{1}{N} \left\{ \int \frac{f^2(\boldsymbol{x}) p^2(\boldsymbol{x})}{q(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} - I^2[f] \right\}, \tag{7}$$

and the density $q^*(\boldsymbol{x})$ that minimizes (7) is proportional to $|f(\boldsymbol{x})|p(\boldsymbol{x})$. To try and give some intuition into what this means, lets consider the common case where $f(\boldsymbol{x}) > 0, \forall \boldsymbol{x}$. Then, if the numerator in (7) goes to zero slower than $q(\boldsymbol{x})$ does, the $\text{var}(\hat{I}[f]) \to \infty$. This means that the selection of $q(\boldsymbol{x})$ will have a huge impact on the accuracy of our estimation. In fact, one of the biggest problems with using the importance sampling method is that a poor selection of our sampling distribution will lead to a high-variance estimate $\hat{I}[f]$ that yields the wrong answer without any indication [1].

One way of making sure that this doesn't happen is to first choose a sampling distribution $q(\boldsymbol{x})$ with non-zero support where $p(\boldsymbol{x})$ has non-zero support. This can be accomplished by choosing a q by minimizing the α-divergence[1]

$$D_\alpha(p||q) = \frac{4}{1-\alpha^2}\left(1 - \int p(\boldsymbol{x})^{\frac{1+\alpha}{2}} q(\boldsymbol{x})^{\frac{1-\alpha}{2}} \mathrm{d}x\right), \tag{8}$$

for $\alpha \geq 1$. Recall that $D_\alpha(p||q)$ is *zero-avoiding* in this range and will typically select a q that covers p. Furthermore, if we choose a sampling distribution as in *rejection sampling* where $p(\boldsymbol{x}) \leq kq(\boldsymbol{x})$ with k a constant, then we are guaranteed to avoid a situation where our variance will "blow up." Our approximation (5) guarantees that our q does not have to be a normalized probability distribution for this method to work.

When actually running this simulation, one might wish to know how many samples are enough. In [3], Kong *et al.* recommend that one use the *effective sample size* defined as:

$$ESS \triangleq \frac{N}{1 + \text{var}(w_i)}$$

to evaluate the impact on the simulation variance of increasing sample size.

4 MATLAB Example

In this section, we will implement importance sampling in order to calculate the expectation of (11) where x is distributed according to a distribution similar to a Chi distribution, but with a non-integer DOF parameter. We will use a scaled normal distribution $\mathcal{N}[x|0.8, 1.5]$ as our sampling distribution where the parameters are chosen so that $p(x) < kq(x), \forall x \geq 0$.

$$p(x) = x^{(1.65)-1} e^{-\frac{x^2}{2}},\ x \geq 0 \tag{9}$$

$$q(x) = \frac{2}{\sqrt{2\pi(1.5)}} e^{-\frac{((0.8)-x)^2}{2(1.5)}} \tag{10}$$

$$f(x) = 2\sin(\frac{\pi}{1.5}x), x \geq 0 \tag{11}$$

It is important to note that neither p nor q are proper distributions here without normalization. The variance of q has been chosen so that as $x \to \infty$, $q(x)$ will decay slower than $p(x)$ in order to analytically keep the variance of our approximation bounded. Furthermore, $p(x)$ is supported only for $x \geq 0$ so we must be careful when sampling from $q(x)$ supported on $x \in \mathbb{R}$ to discard any samples $x < 0$.

[1] Bishop pg. 470

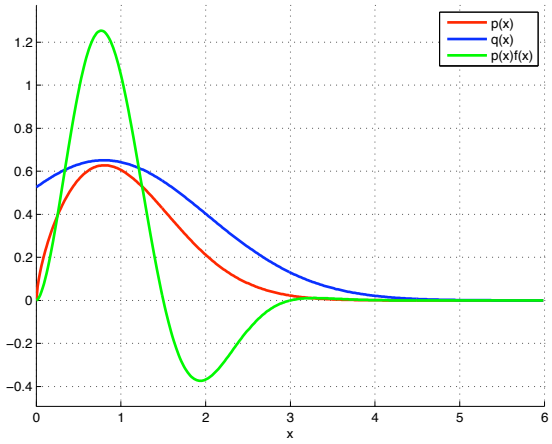

Figure 1: A plot of $p(x)$ and $q(x)$ shows how well the sampling distribution covers $p(x)$. We show $p(x)f(x)$ in order to justify using numerical evaluation from [0,5] to approximate the expectation.

```
% MATLAB code for implementing a simple importance sampling scheme
% Ryan Guerra
% Nov 17, 2008

clear all;

mu = 0.8;                                   % gaussian mean parameter
sigma = sqrt(1.5);                          % gaussian standard deviation parameter
k=1.65;                                     % chi DOF parameter
c=2;                                        % multiplicative constant to make p < cq
num_iterations = 100;                       % number of Monte Carlo runs to execute

p = @(x) x^(k-1)*exp(-x^2/2);               % target distribution
q = @(x) c/sqrt(2*pi*sigma^2)...            % sampling distribution
    *exp(-(mu-x)^2/(2*sigma^2));
f = @(x) 2*sin(pi/1.5*x);                   % function we want to take expectation of

for iter=1:num_iterations
    samp_size = 1000;                       % number of initial samples to draw from q
    X = normrnd(mu,sigma,[samp_size 1]);    % generate vector of samples from q

    for i=1:length(X)
        if X(i)>=0                          % we will discard all sampled X that are not in the support of p
            W(i) = p(X(i))/q(X(i));         % calculate importance weight for sample i
            I(i) = W(i)*f(X(i));            % weigh our function
        else
            W(i)=0;                         % these values will be ignored
            I(i)=0;
            samp_size = samp_size - 1;
        end
    end
    I_hat = sum(I)/sum(W);                  % perform summation in (5)
    [tmp tmp2 non_zero_weights] = find(W);  % remove all discarded importance weights
    variance = var(non_zero_weights);       % calculate the variance of the importance weights
    eff_samp_size = samp_size/(1 + variance);   % calculate the effective sample size
    format short g                          % store the results of this run
    results(iter,:) = [I_hat variance samp_size eff_samp_size];
end
                                            % print compiled results
totals = [mean(results(:,1)) mean(results(:,2)) mean(results(:,3)) mean(results(:,4));...
          var(results(:,1))  var(results(:,2))  var(results(:,3))  var(results(:,4))]
```

In order to explore the effects of selecting a poor distribution to cover p, we repeat the experiment and shift the mean of q so that the centers of mass for each distribution are far apart. Again, our $N_0 = \{10, 100, 10^3, 10^4\}$

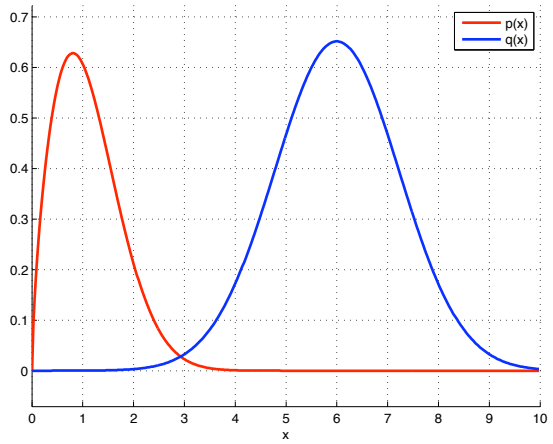

Figure 2: By shifting the mean of the sampling distribution to $\mu = 6$, we can see the results of choosing a $q(x)$ with a center of mass far away from $p(x)$.

Sample Size Per Trial	Good Sampling Distribution Match				Poor Sampling Distribution Match			
	Estimate	Weight Variance	Sample Size	ESS	Estimate	Weight Variance	Sample Size	ESS
10								
Mean of 100 Trials	0.795140	0.06476	7.3	6.88	0.3675	14.00	10	9.175000
Trial Variance	*0.221170*	*0.0011*	*1.94*	*1.71*	*1.78*	*12,144*	*0*	*6*
100								
Mean of 100 Trials	0.730630	0.06824	74.1	69.41	0.3304	6.81	100	88.455000
Trial Variance	*0.025659*	*0.00010*	*17.82*	*16.14*	*1.38*	*2,455*	*0*	*691*
1000								
Mean of 100 Trials	0.764220	0.07020	744.9	696.07	-0.8650	1,110.80	1,000	391.820000
Trial Variance	*0.001929*	*0.000011*	*201*	*190*	*0.81*	*31,354,000*	*0*	*144,260*
10000								
Mean of 100 Trials	0.768300	0.06970	7,434	6,949.70	-0.1625	6,199.90	10,000	343.170000
Trial Variance	*0.000219*	*0.000001*	*2,415*	*2,136*	*1.07*	*437,300,000*	*0*	*407,050*

When this code is run with several different sample sizes $N_0 = \{10, 100, 10^3, 10^4\}$ we can see that around 25% of our samples from $q(x)$ are discarded as being unsupported, thus making our real sample sizes on average $N = \{7, 74, 745, 7434\}$. Naturally, when the mean of our sampling distribution is shifted to $\mu = 6$, the probability of sampling an unsupported x value becomes negligibly small.

Numerical evaluation of $\mathbb{E}_p\{f(x)\}$ on the interval $[0, 5]$ with MATLAB's adaptive Simpson quadrature integrator yields 0.7753 for comparison with our results.

As expected, when we have a sampling distribution that closely matches our target distribution then an increased sample size has an inverse relationship to the Monte Carlo simulation variance. This can be interpreted as a direct result of the *effective* sample size scaling linearly with increased sample size due to the constant variance of the importance weights.

When our sampling distribution is a poor match, we can see that an increased sample size has little or no effect on the Monte Carlo simulation variance. As we can see, this is a direct result of the effective sample size failing to scale with sample size. Thus naively increasing the sample size without improving our sampling distribution will lead to dramatically decreased computational efficiency without any tangible benefit.

While we chose a sampling distribution in this experiment by empirical observation, an even better method would be to use the result of a Laplace approximation or optimizing the α-divergence as recommended in the previous section to discover an optimal sampling distribution. This would be required in a more complicated problem where observation is not possible.

References

[1] C. Bishop, *Pattern Recognition and Machine Learning*, Cambridge, U.K., Springer Science 2006.

[2] A. Owen, Y. Zhou, "Safe and Effective Importance Sampling", Journal of the American Statistical Association, Vol. 95, No. 449, Theory and Methods, March 2000.

[3] A. Kong, J. Liu, W. Wong, "Sequential Imputations and Bayesian Missing Data Problems" Journal of the American Statistical Association, Vol. 89, No. 425, Theory and Methods, March 1994.