CS 5135/6035 Learning Probabilistic Models

Exercise Questions for Lecture 20: Monte Carlo Integration

Gowtham Atluri

4/7/2020

Questions

1. You want to find the mean (point-estimate) of a posterior that is not in a standard form. How do you compute this mean using Monte Carlo integration? **[3 points]**
   [*Hint:* Using $\mathbb{E}_{p(\theta|y)}[\theta] = \int \theta p(\theta|y)$, determine your choice of $g(x)$ and $p(x)$]

2. Using *ordinary* Monte Carlo method, compute the value of the integral **[10 points]**

$$\int_1^7 x^{2.7} e^{-x} dx$$

   a. Specify the values for $g(x)p(x)$ in the factorization $f(x) = g(x)p(x)$.
   b. What tests do you have to do to ensure that your choices of $g(x)$ and $p(x)$ are valid?
   c. Write the algorithm for *ordinary* Monte Carlo method
   d. Write Julia code and approximate the value of the integral
   e. Empirically estimate the mean and variance of the approximation (use 100 runs)

3. Using *Importance Sampling* method, compute the value of the above integral **[10 points]**
   a. Visualize the integrand and the function $q(x) = (7 - x)/18$ and determine if this $q(x)$ is a better choice for $p(x)$ than a Uniform distribution.
   b. Write the algorithm for *Importance Sampling* method (use the above $q(x)$)
   c. Write Julia code and approximate the value of the integral (use the above $q(x)$)
   d. Empirically estimate the mean and variance of the approximation (use 100 runs)

4. Compare the mean and variance of the approximations from *ordinary* Monte Carlo method and *Importance Sampling* method. Which of these methods resulted in lower variance and why? **[2 points]**

Bonus question

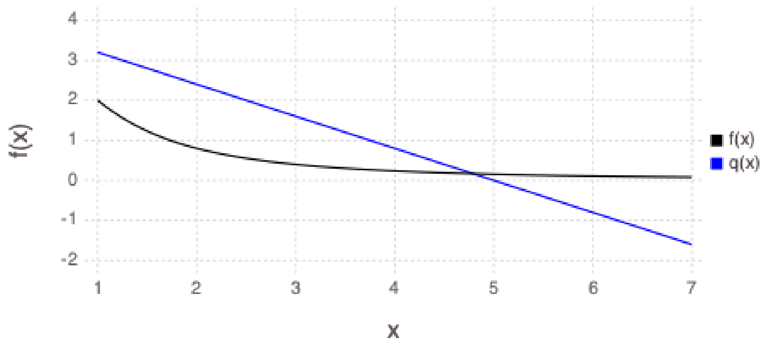
1. For question (3) above, determine a better choice for $q(x)$ and show that the empirical variance is much smaller than what you observed in (3).

Sample code

1. Julia code for visualizing an function

```
x = collect(1:0.01:7);
f(x) = 4/(1+(x.*x));
g(x) = (100-20*x)/25;
```

```
myplot = plot(
        layer(x=x,y=f.(x),Geom.line,Theme(default_color=colorant"black")),
        layer(x=x,y=g.(x),Geom.line,Theme(default_color=colorant"blue")),
        Guide.ylabel("f(x)"),Guide.xlabel("x"),Coord.Cartesian(xmin=1, xmax=7),
        Guide.manual_color_key("", ["f(x)", "q(x)"], ["black","blue"]));
draw(PNG("./figs/viz_f.png", 5inch, 2.5inch), myplot);
```



2. Julia code for *ordinary* Monte Carlo method

```
n=10000;
delta = 5;
f(x) = 4/(1+x^2);
x = rand(Uniform(0,5),n);
S = sum(delta.*f.(x))/n
```

3. Julia code to estimate mean and variance of the approximation for *ordinary* Monte Carlo method

```
n = 10000;
delta = 5;
f(x) = 4/(1+x^2);
S = zeros(100);
for i= 1:100
    x = rand(Uniform(0,5),n);
    S[i] = sum(delta.*f.(x))/n;
end
mean(S)
var(S)
```

4. Julia code for accept-reject method

```
function accept_reject_method(n)
    x = 0:0.01:5;
    f(x) = (10-2x)/25;
    g(x) = pdf(Uniform(0,5),x);
    M = maximum(f.(x)./g.(x));
    count = 0;
    samples = [];
    while(count<n)
        y = rand(Uniform(0,5));
        u = rand(Uniform(0,1));
        if(u<f(y)/(M*g(y)))
            samples = [samples; y];
            count +=1;
```

```
        end
    end
    return samples;
end
```

5. Julia code for Importance sampling

```
f(x) = 4/(1+x^2);
q(x) = (10-2x)/25;

n = 10000;
x = accept_reject_method(n);
S = sum(f.(x)./(q.(x)))/length(x)
```

6. Julia code to estimate mean and variance of the approximation for Importance Sampling method

```
n = 10000;
f(x) = 4/(1+x^2);
p(x) = (10-2x)/25;
S = zeros(100);
for i= 1:100
    x = accept_reject_method(n);
    S[i] = sum(f.(x)./(p.(x)))/n;
end
mean(S)
var(S)
```