CS 5135/6035 Learning Probabilistic Models

Exercise Questions for Lecture 21: Markon Chain Monte Carlo Methods I

Gowtham Atluri

4/9/2020

Questions

1. State-space of a Markov Chain is $[sunny, cloudy, rainy]$, Initial distribution is $\pi_0 = [0.8, 0.05, 0.15]$

$$\text{Transition probability matrix } K = \begin{pmatrix} & Sunny & Cloudy & Rainy \\ Sunny & 0.6 & 0.3 & 0.1 \\ Cloudy & 0.2 & 0.3 & 0.5 \\ Rainy & 0.4 & 0.1 & 0.5 \end{pmatrix}$$

   Compute the probability of observing a sequence $sunny, sunny, rainy$. **[3 points]**

2. A posterior distribution $p(\theta|y)$ has the functional form $f(x) \propto (1-x)^2 sin^2(6\pi x) \quad (0 \leq x \leq 1)$ using Metropolis-Hastings Algorithm. Use the proposal distribution $q(x_{cand}|x) = \mathcal{N}(x, 0.1)$ **[15 points]**

   a. Write the equation for acceptance probability.
   b. Write the steps involved in Metropolis-Hastings Algorithm
   c. Visualize the function and the proposal distribution (at points $x = 0.2$, $x = 0.5$, and $x = 0.7$). Based on your visualization, is it possible for the proposal distribution to generate a sample $x_{cand} = 0.3$ when $x_{i-1} = 0.5$?
   d. Implement Metropolis-Hastings Algorithm in Julia and generate 10,000 samples.
   e. Generate a histogram of the samples and comment if the shape of this histogram is similar to the shape of the target function.
   f. Generate a visualization for the trace of the MH algorithm.

3. Changing the variance in the proposal distribution. **[7 points]**

   a. Using the proposal distribution $q(x_{cand}|x) = \mathcal{N}(x, 0.03)$, visualize the function and the proposal distribution (at points $x = 0.2$, $x = 0.5$, and $x = 0.7$). Based on your visualization, is it possible for the proposal distribution to generate a sample $x_{cand} = 0.3$ when $x_{i-1} = 0.5$?
   b. Make changes to your MH implementation according to use this new proposal distribution. Generate 10,000 samples.
   c. Generate a histogram of the samples and comment if the shape of this histogram is similar to the shape of the target function.
   d. Generate a visualization for the trace of the MH algorithm. What changes do you notice in this trace in comparison to the trace when $q(x_{cand}|x) = \mathcal{N}(x, 0.1)$ was used as the proposal distribution?
   e. Explain the reason for the differences that you may have noticed in the traces.

Bonus questions

1. *Irreducibility of a Markov Chain*
   a. How do you determine if a given Markov Kernel represents an irreducible Markov Chain?
   b. Is the Markov Chain represented by the following Markov Kernel irreducible?
      [Hint: You may use write Julia code to test for irreducibility]

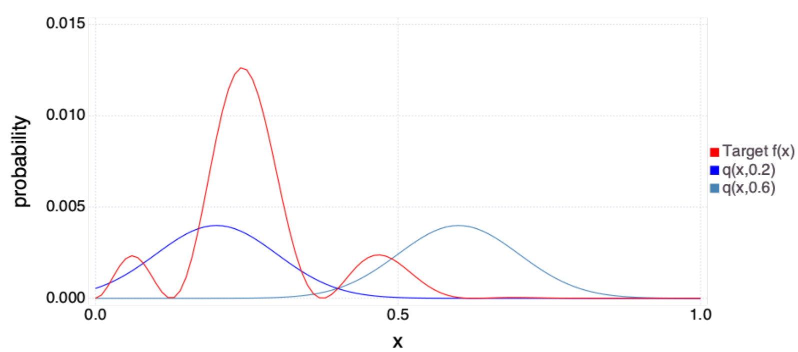
$$K = \begin{array}{cc} & \begin{array}{ccc} A & B & C \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{pmatrix} 0.6 & 0.4 & 0 \\ 0 & 0.5 & 0.5 \\ 0.4 & 0 & 0.6 \end{pmatrix} \end{array}$$

   c. Using $\pi_0 = [0.1, 0.4, 0.5]$, determine the stationary distribution.
      [Hint: Write Julia code to compute the stationary distribution]
2. Changing the variance in the proposal distribution.
   a. Using the proposal distribution $q(x_{cand}|x) = \mathcal{N}(x, 0.01)$, visualize the function and the proposal distribution (at points $x = 0.2$, $x = 0.5$, and $x = 0.7$). Based on your visualization, is it possible for the proposal distribution to generate a sample $x_{cand} = 0.3$ when $x_{i-1} = 0.5$?
   b. Make changes to your MH implementation according to use this new proposal distribution. Generate 10,000 samples.
   c. Generate a histogram of the samples and comment if the shape of this histogram is similar to the shape of the target function. Print your samples and make sure they are in the desired range.
   d. Generate a visualization for the trace of the MH algorithm. What changes do you notice in this trace in comparison to the trace when $q(x_{cand}|x) = \mathcal{N}(x, 0.1)$ was used as the proposal distribution?
   e. Explain the reason for the differences that you may have noticed in the traces.

Sample code

1. Julia code for visualizing target function and proposal distribution

```
x=collect(0:0.01:1);
f(x) = 2.*x.^2.*(1.-x).^8.*cos.(4.*pi.*x).^2;
q(x,y) = pdf.(Normal.(y,0.1),x);
myplot=  plot(layer(x=x,y=f(x),Geom.line,Theme(default_color=colorant"red")),
        layer(x=x,y=q(x,0.2)/1000,Geom.line,Theme(default_color=colorant"blue")),
        layer(x=x,y=q(x,0.6)/1000,Geom.line,Theme(default_color=colorant"steelblue")),
        Guide.ylabel("probability"),Guide.xlabel("x"), Guide.manual_color_key("",
        ["Target f(x)", "q(x,0.2)", "q(x,0.6)"], ["red","blue","steelblue"]), white_panel);
draw(PNG("./figs/mh_viz_fun.png", 10inch, 4.5inch), myplot);
```



2. Julia code for Metropolis Hastings Algorithm

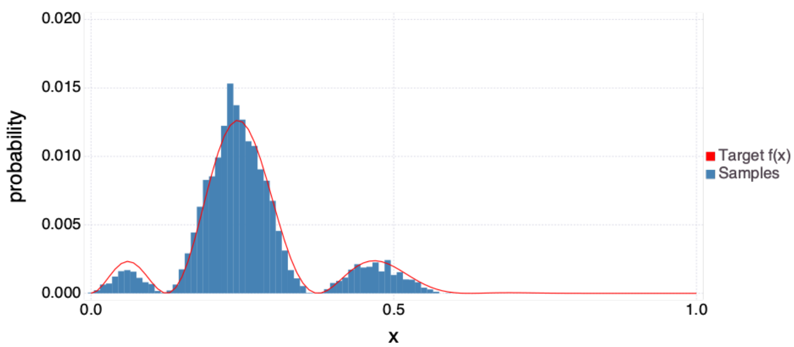
```
function metropolis_hastings(n)
    x = zeros(n); count = 1;
    x[1] = abs(rand(Normal(0,0.1)));

    while(count < n)
        x_cand = rand(Normal(x[count],0.1));
        if((x_cand<0) | (x_cand > 1)) continue; end
        rho = (q(x[count],x_cand)/
                q(x_cand,x[count]))*(f(x_cand)/f(x[count]));
        alpha = minimum([1,rho]);
        u = rand();
        count = count + 1
        if (u < alpha)
            x[count] = x_cand;
        else
            x[count] = x[count-1];
        end
    end
    return x;
end
```

```
## metropolis_hastings (generic function with 1 method)
```

```
samples = metropolis_hastings(10000);
```

3. Julia code for visualizing sample distribution

```
hist = [fit(Histogram,samples,collect(0:0.01:1)).weights; 0]./53150;
myplot = plot(layer(x=x,y=f(x),Geom.line,Theme(default_color=colorant"red")),
         layer(x=collect(0:0.01:1),y=hist, Geom.bar,
               Theme(default_color=colorant"SteelBlue")),
         Guide.manual_color_key("", ["Target f(x)", "Samples"],
                                ["red","SteelBlue"]),
         Guide.ylabel("probability"),Guide.xlabel("x"),
         Coord.Cartesian(xmin=0,xmax=1),white_panel);
draw(PNG("./figs/mh_viz_sample.png", 10inch, 4.5inch), myplot);
```



4. Julia code to trace Metropolis-Hastings Algorithm

```
myplot1 = plot(y=900:1100,x=samples[900:1100], Geom.path, Guide.ylabel("time"),
               Guide.xlabel("Samples"),Coord.Cartesian(xmin=0,xmax=1),white_panel);
myplot2 = plot(y=1:10000,x=samples[1:10000], Geom.path, Guide.ylabel("time"),
               Guide.xlabel("Samples"),Coord.Cartesian(xmin=0,xmax=1),white_panel);
myplot= hstack(myplot1,myplot2);
```

```
draw(PNG("./figs/mh_trace.png", 10inch, 6inch), myplot);
```