CSCI 6220/4030: Homework 6

Assigned Friday November 22 2018. Due at beginning of class Monday December 9 2018.

Remember to typeset your submission, and label it with your name. Please start early so you have ample time to see me during office hours. Provide mathematically convincing arguments for the following problems. Ask me if you are unclear whether your arguments are acceptable.

1. Gibbs sampling requires the ability to sample from the conditional distribution of one coordinate of the state given the other coordinates of the state. In class we saw how to do this for the Ising model, where each coordinate has two possible values, ± 1. To do Gibbs sampling over models where the coordinates can take on a finite sets of values, we can generalize the idea from class.

   Consider the following algorithm for sampling from a given discrete distribution over $[n]$:

Input: p, a probability vector on $1, \ldots, n$
Output: a sample from the distribution given by p

$z \leftarrow$ sample from Unif$[0, 1]$
$c \leftarrow 0$
for $i \leftarrow 1 \ldots n$ **do**
  $c \leftarrow c + p[i]$
  if $z \leq c$ **then**
    return i

   Prove the correctness of this algorithm: show that $\mathbb{P}[\text{i is returned}] = p_i$ for $i \in \{1, \ldots, n\}$.

2. Remember a drawback of Gibbs sampling is that it requires the ability to *sample* from the conditional distributions $f(x_i | x_1, \ldots, x_{i-1}, x_{i+1}, x_d)$, which can be intractable. Consider the following more general scheme which only requires the ability to *compute* $f(\mathbf{x})$.

Input: $\tilde{\mathbf{P}}$, a Markov chain on S such that $\tilde{P}_{\mathbf{x},\mathbf{y}} = \tilde{P}_{\mathbf{y},\mathbf{x}}$ and $\tilde{P}_{\mathbf{x},\mathbf{x}} = 0$; $\boldsymbol{\pi}_0$, an initial distribution on S; and f, a probability distribution on S.
Output: $\mathbf{x}_T$, an approximate sample from the distribution given by f

$\mathbf{x}_0 \leftarrow$ a sample from $\boldsymbol{\pi}_0$
for $i \leftarrow 1 \ldots T$ **do**
  $\mathbf{y} \leftarrow$ sample from $\tilde{\mathbf{P}}$ with starting state $\mathbf{x}_{i-1}$
  $z \leftarrow$ sample from Unif[0,1]
  if $z \leq \min\{1, \frac{f(\mathbf{y})}{f(\mathbf{x}_{i-1})}\}$ **then**
    $\mathbf{x}_i \leftarrow \mathbf{y}$
  else
    $\mathbf{x}_i \leftarrow \mathbf{x}_{i-1}$

   This algorithm (called a Metropolis sampler) proposes state updates using the Markov chain $\tilde{\mathbf{P}}$ and accepts them with probability $\min\{1, \frac{f(\mathbf{y})}{f(\mathbf{x}_{i-1})}\}$; thus the probability of moving from state $\mathbf{x}$ to state $\mathbf{y}$ is

$$P_{\mathbf{x},\mathbf{y}} = \tilde{P}_{\mathbf{x},\mathbf{y}} \min\left\{1, \frac{f(\mathbf{y})}{f(\mathbf{x})}\right\} \quad \text{when } \mathbf{x} \neq \mathbf{y}.$$

   (a) What is $P_{\mathbf{x},\mathbf{x}}$?

   (b) Argue that $\mathbf{P}$ is a Markov chain, and the above algorithm samples from $\mathbf{P}$.

   (c) Show that f satisfies detailed balance with respect to $\mathbf{P}$ and therefore is a stationary distribution of $\mathbf{P}$.

(d) Argue that if $\tilde{\mathbf{P}}$ is ergodic and $f(\mathbf{x}) > 0$ for all states $\mathbf{x}$, then $\mathbf{P}$ is also ergodic and therefore this algorithm will approximately sample from f if T is large enough.

3. Juhwan works at an engineering company that produces industrial piping[1]. His latest project involves structural optimization to balance the price versus reliability of the piping designed for a particular application.

   The application is characterized by the probability that the flow will have given x and y velocity components, and the problem of choosing the optimal structure boils down to choosing the function f— from a large library of possible choices—with minimum expectation under the probability distribution of the given flow. In this application, the velocity is distributed on $\{-200, \ldots, 700\} \times \{-200, \ldots, 700\}$ according to the distribution

$$p(x, y) = c \exp\left(-\frac{(x/100)^2(y/100)^2 + (x/100)^2 + (y/100)^2 - 8x/100 - 8y/100}{2}\right),$$

   where c is the constant required to make this a distribution.

   The structural library holds S structures, and Juhwan wants to design a scheme for finding the optimal structure that is more efficient than brute force $O(810{,}000S)$ computation of the expectations. Also, he wants to future-proof his approach: he knows that the company will eventually start using more detailed six-dimensional performance models that take into account the z velocity components as well as the accelerations in each direction, and the company is always adding more structures to their library.

   Juhwan took a randomized algorithms course, so he has the vague idea that MCMC methods may be relevant here. Indeed, he refers back to his notes and notices that if $\{(x_i, y_i)\}_{i=1}^N$ are i.i.d. samples from an ergodic Markov chain $\mathbf{P}$ that has p as its stationary distribution, each taken after $\tau(\epsilon)$ steps of the Markov chain starting from distribution $\boldsymbol{\pi}_0$, then for any function f that satisfies $|f(x, y)| \leq M$ for all velocities,

$$\begin{aligned}\left|\mathbb{E}_p f - \frac{1}{N}\sum_{i=1}^N f(x_i, y_i)\right| &\leq 2M d_{\mathrm{TV}}(p, \boldsymbol{\pi}_0\mathbf{P}^t) + \sqrt{\frac{\mathrm{Var}_{\boldsymbol{\pi}_0\mathbf{P}^t}(f)}{\delta N}} \\ &\leq 2M\left(\epsilon + \frac{1}{\sqrt{\delta N}}\right)\end{aligned}$$

   with probability at least $1 - \delta$. The first term in this approximation error comes from approximating the expectation of f under the distribution p with its expectation under the distribution $\boldsymbol{\pi}_0\mathbf{P}^t$, and the second term comes from estimating the latter expectation with an empirical mean.

[1] Or circuits, or ballerina shoes.

This suggests the following scheme: given a precision $\varepsilon \in (0,1)$ take

$$N \geq \frac{16SM^2}{\delta\varepsilon^2}, \quad \epsilon \leq \frac{\varepsilon}{4M}$$

and generate N samples from a Markov chain of the type described above. Then, with probability at least $1-\delta$, the Monte Carlo estimates of the expected values of all S functions in the structural library will be accurate to within $\pm\varepsilon$. Thus if ε is small enough, the Monte Carlo estimates can be used to find the optimal structure[2].

This scheme has attractive properties: it can readily be extended to handle three-dimensional velocities, the convergence rate with respect to N does not depend on the number of dimensions in the model[3], and Juhwan can overestimate S to get a set of Monte Carlo samples that will remain useful with high probability even if the library is expanded.

As long as $16M^2(\delta\varepsilon^2)^{-1} < 810,000$, this scheme will be more efficient than brute force computation. And when the company gets around to building six-dimensional models, this scheme will be more efficient than brute force computation as long as $16M^2(\delta\varepsilon^2)^{-1} < (900)^6$.

Help Juhwan implement his scheme.

(a) Explain why it would be very expensive to use a Gibbs sampler to draw samples from this distribution.

(b) Explain why it would be very expensive to use the approach from Problem 1 to draw samples from this distribution in the six-dimensional case.

(c) Implement a Metropolis sampler to draw samples from this distribution. Use this sampler to draw 10^6 samples, and plot the last 10000 samples in a scatter plot.

Take $\boldsymbol{\pi}_0$ to assign probability 1 to $(-200,-200)$, and for the proposals, to ensure symmetry ($\tilde{P}_{\mathbf{v}_1,\mathbf{v}_2} = \tilde{P}_{\mathbf{v}_2,\mathbf{v}_1}$), use the following wrap-around Markov chain:

$$\tilde{P}_{\mathbf{v}_1,\mathbf{v}_2} = \begin{cases} 0 & \mathbf{v}_1 = \mathbf{v}_2 \text{ or } \max\{|x_1 - x_2|, |y_1 - y_2|\} \bmod 901 > \Delta \\ \frac{1}{(2\Delta+1)^2 - 1} & \text{otherwise} \end{cases}$$

with $\Delta = 20$.

(d) Attach your code[4] for the Metropolis sampler above to your homework submission.

[2] We are ignoring the issue of determining the mixing time.

[3] But note that the mixing time will probably depend on the number of dimensions in the model.

[4] Make your code coherent and easy to follow. I don't care what language you use