APTS Computer Intensive Statistics 2016/17

Assessment Questions

The intention of these exercises is to provide an opportunity for you to demonstrate what you can do. They are intended to include many of the disparate aspects of *Computer Intensive Statistics* and it is not envisaged that anyone should be asked to answer all of them (although, of course, you're welcome to do so if you find it interesting). Discuss which questions you should attempt with your supervisor. It's anticipated that most students will attempt about two of these questions and won't expend much more than one day of effort on the task.

1. Consider the application of bootstrap methods to a simple random sample of size $n = 100$ obtained from a standard normal population:

$$X_1, \ldots, X_n \overset{\text{iid}}{\sim} \mathsf{N}(0, 1).$$

   (a) Simulate such a sample, let us call it $\boldsymbol{x}^\star = x_1^\star, \ldots, x_n^\star$.

   (b) We know that the most immediately intuitive estimator of the population variance,

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2 \qquad \text{where } \bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

   is biased.

      i. What is the bias in this case.

      ii. Use a bootstrap method to estimate the bias of this estimator using $\boldsymbol{x}^\star$ as the real sample.

      iii. Repeat *ii.* a number of times to assess the bootstrap method in this case.

   (c) Population kurtosis is another quantity which we might wish to estimate from sample data. One simple estimator of this quantity might be:

$$\hat{k} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^4}{n(\hat{\sigma}^2)^2}$$

      i. Estimate $\hat{k}$ for $\boldsymbol{x}^\star$.

      ii. Estimate the bias of $\hat{k}$ using a bootstrap method.

      iii. Obtain a 95% confidence interval (try a bootstrap percentile interval, or a more sophisticated method if you prefer) for the population kurtosis.

2. Choose a multimodal univariate probability density and:

   (a) Implement an exact slice sampler for this density. You'll need to identify the level sets either analytically or numerically and to sample from these level sets.

   (b) Implement a random walk Metropolis algorithm for which your chosen density is the invariant distribution.

   (c) Implement a Metropolised slice sampler for this density.

   (d) Compare the three algorithms which you have implemented, taking into account the quality of the approximation and the computational cost of the various approaches.

3. Simulated Annealing.

   (a) Implement a simulated annealing algorithm to minimize the function:

   $$f(x_1, x_2) = (4 - 2.1x_1^2 + x_1^4/3) \cdot x_1^2 + x_1 \cdot x_2 + 4(x_2^2 - 1) \cdot x_2^2$$

   within the bounded region defined by $-3 \leq x_1 \leq 3$ and $-2 \leq x_2 \leq 2$.

   (b) Investigate the performance of your algorithm with various annealing schedules and proposal scales. What's the lowest value you can locate for f and at what value of x_1, x_2 is it found?

   (c) Consider the more challenging function

   $$\begin{aligned} f(x_1, x_2) = &\exp(\sin(50x_1)) + \sin(60\exp(x_2)) + \sin(70\sin(x_1)) + \sin(\sin(80x_2)) \\ &- \sin(10(x_1 + x_2)) + \frac{1}{4}(x_1^2 + x_2^2) \end{aligned}$$

   for $|x_1| \leq \pi/2, |x_2| \leq \pi/2$. What's the smallest value of f you can identify, and at what coordinates (x_1, x_2) is this achieved?

4. Consider the Ising model on a graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$ which was briefly mentioned in lectures. Let $m = |\mathcal{V}|$. Recall that the distribution over the ± 1-valued binary variables attached to each vertex has probability mass function:

   $$p(x_1, \ldots, x_m) = \frac{1}{Z} \exp\left(J \sum_{(i,j) \in \mathcal{E}} x_i x_j \right)$$

   where Z is a normalising constant and the sum is over all adjacent vertices with the graph. Consider the *ferromagnetic* case in which the coupling strength $J > 0$.

   This question concerns the relationship between the Ising model and a related bond percolation model and a data augmentation strategy which allows a very efficient algorithm to be implemented for this model.

   Consider adding a Bernoulli variable to every edge in the graph such that there is a $U_{i,j}$ associated with every $(i, j) \in \mathcal{E}$. Conditional upon the value of $\boldsymbol{X} = X_1, \ldots, X_m$, these variables are mutually independent. If $X_i \neq X_j$ then $U_{i,j} = 0$, otherwise $U_{i,j} \sim \mathsf{Ber}\,(1 - \exp(-2J))$. This may seem rather a peculiar thing to do at first, but the interpretation is that bonds are introduced at random between those adjacent vertices which take common values; the stronger the coupling strength the greater the probability that adjacent like vertices are bonded.

   (a) Write down the joint distribution of $\boldsymbol{X}$ and $\boldsymbol{U}$ where $\boldsymbol{U} = \{u_{i,j} : (i, j) \in \mathcal{E}\}$.

   (b) Simplify your expression to express the probability distribution (up to a constant of proportionality) in terms of: the number of unlike adjacent vertices, the number of bonded like adjacent vertices and the number of unbonded like adjacent vertices implied by $\boldsymbol{X}$ and $\boldsymbol{U}$.

   (c) What is the conditional distribution of $\boldsymbol{X}$ given $\boldsymbol{U}$?

   (d) Identify a Gibbs Sampling algorithm in which one iteratively updates first the entirety $\boldsymbol{X}$ and then the entirety of $\boldsymbol{U}$.

   (e) Why might the Gibbs Sampler described here be expected to outperform the Naïve Gibbs sampling strategy for the Ising model?

5. If your PhD involves a model[1] for which you've been conducting inference by other means, try implementing one or two simple Markov chain Monte Carlo algorithms in order to obtain Bayesian estimates of the unknown parameters (you'll need to choose some prior distributions if you don't already have these). Contrast these estimates with those obtained with whatever other methods you've been using.

[1] If it's a complicated model you might want to consider a simplification of that model, this exercise isn't intended to take a very long time and implementing good MCMC algorithms for complex models can be very time consuming.