Homework #2—EAS 520 & DSC 520 & MTH 499

Assigned: Thursday, October 12, 2017
Due: Wednesday, October 25, 2017 (midnight)

Reading & viewing: If you would like additional experience with C programing, please consider viewing all or part of https://www.youtube.com/watch?v=-CpG3oATGIs and consulting http://www.cprogramming.com/tutorial/c-tutorial.html

On every homework you turn in, be sure to upload the computer code you used to generate your solutions to Bitbucket (that is, commit your code to the local git repository and push these changes to Bitbucket). Put all the figures and answers asked for in all the questions into a single well organized PDF document (prepared using LaTeX) and upload it onto the myCourses website.

Monte Carlo integration: Please consult homework 1 for general background on Monte Carlo integration.

Problem 1: (*50 points*)

In this problem you will use Monte Carlo integration to compute π. I strongly recommend you read problem 2 before attempting this problem – if you write your code correctly, you should be able to extend it to solve problem 3 with little effort!

In order to calculate the value of π, consider the following setup. Let $D \subset \mathbb{R}^2$ be the two-dimensional unit disc, i.e., $D = \left\{x \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leq 1\right\}$, and let R be the circumscribing square: $R = \left\{x \in \mathbb{R}^2 \mid |x_1| \leq 1 \text{ and } |x_2| \leq 1\right\}$. Let $f(x) \equiv \mathbb{1}_D(x)$ be the indicator function on D:

$$\mathbb{1}_D(x) = \begin{cases} 1, & x \in D, \\ 0, & x \notin D. \end{cases}$$

Then

$$\int_R f(x)\,\mathrm{d}x = \int_R \mathbb{1}_D(x)\,\mathrm{d}x = \int_D 1\,\mathrm{d}x = \mathrm{area}(D) = \pi. \tag{1}$$

Thus, if we take X_i as iid random variables uniformly distributed on R, the Monte Carlo integration formula (from homework 1) can be used to compute π:

$$\pi = 4 \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_D(X_i)\,, \tag{2}$$

where we have used the fact that

$$V = \int_R \mathrm{d}x = 4\,, \tag{3}$$

since the integration region R is a square whose edge length is 2. When taking $N < \infty$ the approximation to π is

$$\pi_N = 4 \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_D(X_i) \tag{4}$$

$$. \tag{5}$$

(a) Write a C, C++ or Fortran program that computes this Monte Carlo approximation to π for $N = 10^k$ for $k = 2, 3, 4, 5, 6, 7, 8, 9, 10$. Now that you know two different ways to get data out of your code (redirecting the program's output or writing to a file) you should use the method you find best suited for the job. Also, note that if X_i is a uniformly distributed on $R = [-1, 1]^2$, then its x and y coordinates are uniform scalar random variables on the domain $[-1, 1]$.

(b) Plot N vs relative error $|\pi - \pi_N|$ on a log-log plot. Provide convincing evidence that the error is decreasing at the correct rate.

(c) Time your program's runtime using the Unix program **time** (or another reliable method. If you do not use the Unix program **time**, say what you did and why it can be trusted). Plot the program's runtime vs relative error on a log-log plot. Estimate how long you would need to run the program to compute π to 10^{-16} accuracy (about the level of precision one can achieve with floating-point numbers using double precision). Estimate how long you would need to run the program to compute π to an accuracy of 10^{-70030}. This would match the world record for "known digits of π" (see http://www.pi-world-ranking-list.com/index.php?page=lists&category=pi).

Problem 2: (*50 points*)

In Bayesian model selection (and parameter estimation) one is often faced with difficult, high-dimensional integrals. For example, suppose we have two models for our data, $M_1(x_1, x_2)$ and $M_2(x_1, x_2)$, both of which are two-dimensional with model parameters x_1 and x_2. To find out which model does a better job at describing our dataset, within a Bayesian framework (and assuming no prior preference for either model) we would compute the Bayesian evidence factors, $\mathcal{Z}_1$ and $\mathcal{Z}_2$, where

$$\mathcal{Z}_i = \int \mathcal{L}(x_1, x_2; M_i)\pi(x_1, x_2; M_i)\, \mathrm{d}x_1\, \mathrm{d}x_2\,. \tag{6}$$

Here $\mathcal{L}(x_1, x_2; M_i)$ is the likelihood function and $\pi(x_1, x_2; M_i)$ is the prior distribution of parameters. If $\mathcal{Z}_1 > \mathcal{Z}_2$ we will prefer model 1 over model 2, for example. Here, the semi-colon notation in $\mathcal{L}(x_1, x_2; M_i)$ means that the model $M_i(x_1, x_2)$ has been assumed in the computation of the likelihood function. How this is done is beyond the scope of this class. Instead, we will assume the likelihood function is given. For simplicity we shall assume $\pi(x_1, x_2; M_i) = 1$. (The prior must satisfy $\int \pi = 1$, so strictly speaking we cannot set $\pi = 1$ in general.)

In this problem you will tackle a challenging case when the parameters are degenerate: that is, there is a special combination if x_1 and x_2 where the model makes the same prediction while simultaneously varying the values of both x_1 and x_2. In such situations, neural networks trying to learn the likelihood surface may also experience significant difficulties. Obviously, the problem gets even worst when the dimensionality of the parameter space grows beyond two.

In the presence of a model degeneracy, one may encounter a likelihood function such as

$$\mathcal{L}_1(x_1, x_2) = \exp\left(-(1-x_1)^2 - 100\left(x_2 - x_1^2\right)^2\right)\,. \tag{7}$$

(a) Plot $\ln\left(\mathcal{L}_1(x_1, x_2)\right)$ over the region $x_1, x_2 \in [-5, 5]$. The model's degeneracy will show up as lines in the x_1-x_2 plane where the value of $\ln \mathcal{L}_1$ does not change.

(b) Modify the code you wrote to solve problem 2 to compute the value of

$$\mathcal{Z}_1 = \int_{-5}^{5}\int_{-5}^{5} \mathcal{L}_1(x_1, x_2)\, \mathrm{d}x_1\, \mathrm{d}x_2 \tag{8}$$

(c) Let $N = 10^k$ for $k = 2, 3, 4, ...$ and compute $\ln\left(\mathcal{Z}_1\right)$ by continually increasing N. Continue increasing N until you can convincingly get at least 2 digits of precision. Quote some values of N and $\ln\left(\mathcal{Z}_1\right)$.

(d) By any means possible (using your code or some other way entirely), compute $\mathcal{Z}_2$ for likelihood function

$$\mathcal{L}_2(x_1, x_2) = 1\,.$$

What fits the data better, model 1 or model 2?