IC252 Lab 8

May 10, 2019

1. **Monte Carlo integration:** Determine the area under the normal density $f()$ between the range a and b, using Monte Carlo sampling. Accept a and b from the user. Print the area for $N = 10$, 100, $\ldots$ 10^8 data points. Verify correctness of your answer using the normal distribution table.

   In brief, Monte Carlo integration is illustrated in figure 1. The area under the curve between $a = 1$ and $b = 2$ is given by the proportion of points inside the curve (in other words, if a point is (x, y), the points having $y < f(x)$), when compared to the total area (a rectangle, in this case.) The random points are to be taken between the appropriate range for both the axes.

2. **Generation of continuous random variables:** Given a random number generator which can produce uniformly distributed samples from $(0, 1)$, the *inverse transform method* can be used to generate samples from any given distribution (exponential, normal etc.)

   In general, the method works as follows:

   - Let X be a random variable whose distribution can be described by the distribution function (cdf) $F_X(x)$.
   - We wish to generate values of X which are distributed according to $F_X(x)$. Note that $U = F_X(X)$ is uniform distributed (question 7 assignment 6).

     We do the following.
   - Generate a random number u from $U(0, 1)$.
   - Find the inverse of the desired cdf i.e. $F_X^{-1}(x)$.
   - Compute $X = F_X^{-1}(u)$. Then X has the desired distribution $F_X(x)$.

   For distributions such as the exponential distribution, $F_X^{-1}(x)$ has a closed-form expression. But for the normal distribution, there is no closed-form expression for the inverse. But you can work around this by generating the table of $F_X(x)$ for a particular quantization (for example, a precision of 3 digits.) Then perform the last step of the above algorithm by performing a table lookup (i.e. search the table for the value of u and return the corresponding x.)

   The table can be generated using the Monte Carlo simulation of the previous question.

   Generate a set of n normally distributed data points as described above. Accept n from the user. Plot the histogram of the data to verify that it follows the desired distribution. Of course, you must not use the library function for generating normally distributed data.

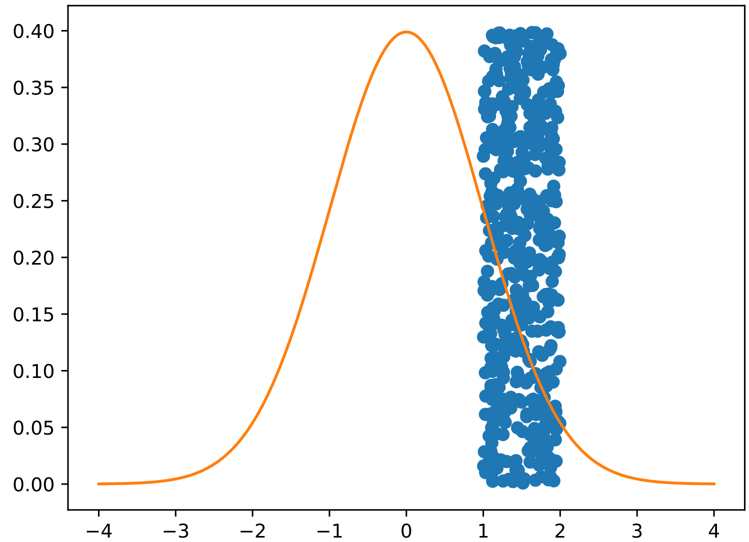

Figure 1: Generating random points in the x-axis range a, b, with $a = 1$ and $b = 2$.

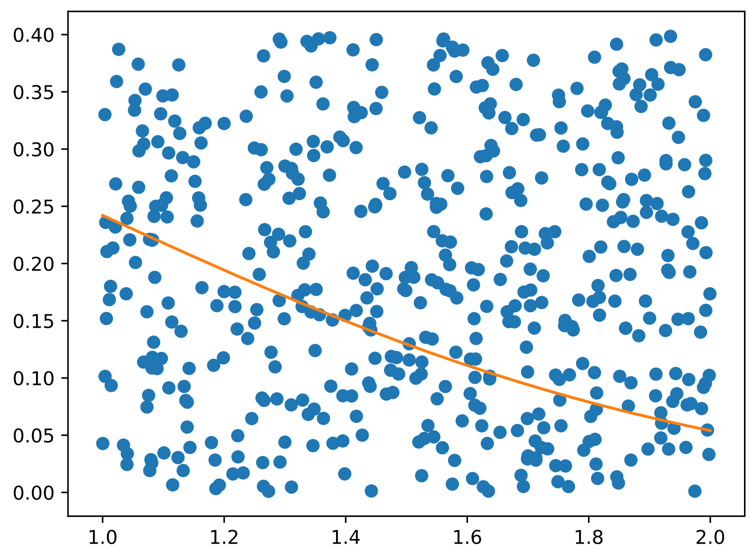

Figure 2: In more detail.