Statistical Methods for Computer Engineering

Homework 4 Report

Yaşar Cahit Yıldırım
2310647

Report

Before starting the Monte Carlo study, we should estimate a threshold value which will denote the minimum number of simulations in order to have the confident answers we want. We want to be 98% confident in our estimates with the maximum error of 0.03. So deciding that $\alpha = 0.02$ and $\epsilon = 0.03$, with the lack of an intelligent preliminary estimate, we can calculate this threshold N as:

$$N \geq 0.25\left(\frac{z_{\alpha/2}}{\epsilon}\right)^2 = 0.25\left(\frac{2.236}{0.03}\right)^2 = 1502.8544$$

We can choose N as any integer that is higher than the threshold found. We can choose, for example, the first integer that is bigger than the value found, i.e. 1503 or the first prime number that is bigger than the value found, i.e. 1511. Let us simply choose 1503 to be able to show the calculation steps in the running code.

Part A

After creating a random symmetric graph with the probability of elements being compatible is 0.012, we can calculate the number of triangles with the formula that has been given to us. For the first shipment to have at most one choice, triangle count of at most one, i.e. less than two, is needed. So, *simulations that has less than two triangles* divided by *the number of simulations* is the answer. The answer, estimated by my simulation in a test run, is as: 0.28277.

Part B

Number of triangles with the graph created by probability 0.79 can be calculated as **Part A**. For the ratio of triangles to triplets, where triplet count is determined by choosing three items out of number of goods available in that simulation, we divide the triangle count of every simulation to the triplet count. So, *simulations that has a ratio more than half* divided by *the number of simulations* is the answer. The answer, estimated by my simulation in a test run, is as: 0.74983.

Part C

Estimated X is the mean of distinct triangles on all simulations.
Estimated Y is the mean of ratios of distinct triangles to triplets on all simulations.

First Scenario
X is, estimated in a test run, as: 3.3560.
Y is, estimated in a test run, as: 4.9794×10^{-6}.

Second Scenario
X is, estimated in a test run, as: 344951.32590.
Y is, estimated in a test run, as: 0.50491.

Part D

Estimated $Std(X)$ is the standart deviation of distinct triangles on all simulations.
Estimated $Std(Y)$ is the standart deviation of ratios of distinct triangles to triplets on all simulations.

First Scenario
$Std(X)$ is, estimated in a test run, as: 2.2552.
$Std(Y)$ is, estimated in a test run, as: 3.3280×10^{-6}.

Since estimated X is bigger by more than one estimated $Std(X)$ from 0 and 1, low percentage in **Part A** is not a surprise.
Because related graph is sparse and Poisson Variable is high i.e. possible triplets is high, estimated Y -also $Std(Y)$- to be really small is expected as well.

Second Scenario
$Std(X)$ is, estimated in a test run, as: 81599.13908.
$Std(Y)$ is, estimated in a test run, as: 0.0070984.

Estimated Y is higher than 50% and estimated $Std(Y)$ is small relative to it, so most of the first standart deviation is higher than 50% too. Thus, high percentage in **Part B** is meaningful.
In a dense graph, a good can participate in a lot of triangles with combinations of other goods. Estimated X to be very big on a dense graph is, then, logical. In the same way, small number of changes in compatibility of goods can affect a lot of triangles. So, $Std(X)$ to be big is also accurate.

Code

```
pA = 0.012;                                  % probability of compatibility for A
pB = 0.790;                                  % probability of compatibility for B

z_a_2 = 2.326; eps = 0.03;                   % P=0.98, Eps=0.03
N = ceil((z_a_2/eps)**2 / 4);                % >= 1502.8544, so 1503

A_num_tri = zeros(N, 1); A_ratio = zeros(N, 1);% containers for sparse case, A
B_num_tri = zeros(N, 1); B_ratio = zeros(N, 1);% containers for dense  case, B

for i = 1 : N;                               % SIMULATIONS
  num_good = poissrnd(160);                  % sample number of goods
  G = rand(num_good, num_good);              % generate random NxN matrix

  G = G - triu(G) + tril(G)';                % copy lower half to upper half
                                             % to make it symmetric

  A = G <= pA;                               % sparse graph for A, G(i, j) <= pA
  B = G <= pB;                               % dense  graph for B, G(i, j) <= pB

  A_num_tri(i) = trace(A^3) / 6;             % triangle count of A
  B_num_tri(i) = trace(B^3) / 6;             % triangle count of B

  gc3 = nchoosek(num_good, 3);               % number of triplets

  A_ratio(i) = A_num_tri(i) / gc3;           % ratio of triangles/triplets for A
  B_ratio(i) = B_num_tri(i) / gc3;           % ratio of triangles/triplets for B
end;

                            % Part A %
partA = mean(A_num_tri < 2)                  % P(shipment <= 1) for A
                            % Part B %
partB = mean(B_ratio > 0.5)                  % P(ratio > 0.5) for B
                            % Part C %
partC_X_A = mean(A_num_tri)                  % estimation of expected X for A
partC_Y_A = mean(A_ratio)                    % estimation of expected Y for A
partC_X_B = mean(B_num_tri)                  % estimation of expected X for B
partC_Y_B = mean(B_ratio)                    % estimation of expected Y for B
                            % Part D %
partD_stdX_A = std(A_num_tri)                % estimation of Std(X) for A
partD_stdY_A = std(A_ratio)                  % estimation of Std(Y) for A
partD_stdX_B = std(B_num_tri)                % estimation of Std(X) for B
partD_stdY_B = std(B_ratio)                  % estimation of Std(Y) for B
```

Example Output

```
%                     partC_X_A = 3.3560            partD_stdX_A = 2.2552
% partA = 0.28277     partC_Y_A = 0.0000049794      partD_stdY_A = 0.0000033280
% partB = 0.74983     partC_X_B = 344951.32590      partD_stdX_B = 81599.13908
%                     partC_Y_B = 0.50491           partD_stdY_B = 0.0070984
```