cs201c: Practice Lab 1

Instructor: Apurva Mudgal

Friday, 9th August 2019, 11 am - 12:50 pm

1 Pareto distribution

Consider a (random) quantity Y, which always takes real values. We say that Y follows the Pareto distribution $\mathcal{P}(\alpha, \beta)$ with parameters $\alpha > 0, \beta > 0$ if it satisfies the following conditions:

1. For any real number $x < \beta$, $Pr(Y \leq x) = 0$.
2. For any real number $x \geq \beta$, $Pr(Y \leq x) = 1 - \left(\frac{\beta}{x}\right)^{\alpha}$

(Here, $Pr(Y \leq x)$ denotes the probability that quantity Y takes a real value less than or equal to x.)

Lab Question 1. Write a C++ function which when given α, β as input, generates a real number by sampling from the Pareto distribution $P(\alpha, \beta)$.

Hint. The above can be implemented using the following steps:

1. Generate a real number r uniformly at random from the interval $[0, 1)$.
2. The output of your C++ function is the unique real number x^* such that: (i) $x^* \geq \beta$ and (ii) $1 - \left(\frac{\beta}{x^*}\right)^{\alpha} = r$.

Note. Use *rand* function in header file *cstdlib.h* in C++ for generating r.

2 Sending files on the internet

Consider two computers A and B connected to the internet. A wants to send B a large file F. To do so, A breaks file F into N small "IP packets", with each packets containing at most 1024 bytes consecutive bytes of file F. Call these packets $P_1, P_2, \ldots, P_N$.

Now, A sends packet P_1 at time $t = 1$, packet P_2 at time $t = 2$, and so on till it finishes by sending packet P_N at time $t = N$. In general, for $1 \leq i \leq n$, packet P_i is sent by computer A on the internet at time $t = i$. Further, each packet is stamped with the time t at which it departs computer A.

The time taken by a packet to go from computer A to computer B on the internet is called "end-to-end delay". The end-to-end delay is not a fixed number, but instead can vary based on factors such as internet traffic, link bandwidth, routers, path taken by the packet (different packets can take different paths through the internet), etc.

The end-to-end delay can be modeled as following a "heavy-tailed distribution", such as the Pareto distribution above. In this section, *we assume that the end-to-end delay follows the Pareto distribution* $\mathcal{P}(2,1)$.

Suppose the end-to-end delay for packet P_i is equal to d_i. To be clear, each d_i is a real number obtained by sampling independently from the distribution $\mathcal{P}(2,1)$. Thus, packet P_i reaches computer B at time $i + d_i$. Ordering the packets by their order of arrival at computer B gives us a permutation $P_{\sigma(1)}, P_{\sigma(2)}, \ldots, P_{\sigma(n)}$ of the packets sent by A. (Permutation σ is a one-to-one function with both domain and range equal to $\{1, 2, \ldots, N\}$.)

Lab Question 2. Write C++ code which when given a positive integer N as input, simulates the above file delivery process from A to B, and outputs the final permutation σ of packets in order of their arrival at computer B.

Notes. (i) In real-life networks, some packets may also get lost and never reach B. We exclude this possibility in the above model. (ii) As the end-to-end delays d_i $(1 \leq i \leq N)$ are random quantities, different runs of your C++ code may produce different permutations (based on the output of random number generator).

3 Worst-case and average-case analysis of insertion sort

Once the N packets are received by computer B in the sequence $P_{\sigma(1)}, P_{\sigma(2)}, \ldots, P_{\sigma(n)}$, B reconstructs file F from these pieces by *sorting them in increasing order of their departure times from computer* A. The purpose of this section is to show that insertion sort will perform very well on this task *on average*, though its worst-case running time is $\Theta(N^2)$.

Lab Question 3. Do experimental evaluation of time taken by insertion sort (code discussed in class) on a permutation τ of $1, 2, \ldots, N$, when:

- (worst-case analysis) τ is the decreasing permutation $(N\ N{-}1\ N{-}2\ \ldots\ 1)$.
- (average-case analysis) τ is a random permutation $(\sigma(1), \sigma(2), \ldots, \sigma(n))$ generated by the C++ code in Lab Question 2 above.
- (average-case analysis) τ is a random permutation generated by the C++ code in Lab Question 2 above, the only difference being that end-to-end delay now follows the distribution $\mathcal{P}(1,1)$.

Plot graph of running time (y-axis) vs. N (x-axis) for $N = 128$, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536.

Notes. (i) You can use "std::chrono" library of C++ for computing time taken by insertion sort code. (ii) For experimental evaluation of worst-case analysis, you can run insertion sort 3 times and take the average of the three running times. (iii) For experimental evaluation of time taken for average-case analysis, you can generate $M = 100$ (random) permutations using C++ code in Lab Question 2, and then take the average of the time taken by insertion sort on each of these M permutations. To be specific, if the times taken by insertion sort on the M permutations are $T_1, T_2, \ldots, T_M$, then you output $\frac{T_1+T_2+\ldots+T_M}{M}$ as the average time.

Lab Question 4. Suppose insertion sort is run on a (random) permutation τ generated by running the C++ code in Lab Question 2 with a general Pareto distribution $\mathcal{P}(\alpha, 1)$ instead of $\mathcal{P}(2, 1)$.

Keep $\beta = 1$ and do experimental evaluation of average-case running time for different values of real parameter α in the range $[1, 20]$. Can you plot the dependence of average-case running time on the parameter α?