CS 381 Week 13 PSO: Reservoir Sampling

Mukul Agarwal

April 26, 2026

Problem. You receive a stream of elements $e_1, e_2, \ldots$, one at a time, without knowing when it will end. Each element can be read only when it arrives; to access it later, you must store it. After every arrival, maintain a sample of the elements seen so far, ready to return if the stream ends. If m elements have arrived, the sample must be drawn from $e_1, \ldots, e_m$. Treat different occurrences as distinct elements, even if their values are equal.

(a) Design and analyze an algorithm to sample one element uniformly at random from the stream; you may use $O(1)$ space.

Solution. The algorithm is as follows:

1. Store the first item e_1 as the sample.
2. For each subsequent item e_j, replace the sample with e_j with probability $1/j$.

Fix an item e_i among the first m items, and let X_j be the event that it is the sample after round j. The algorithm selects e_i at round i with probability $1/i$. Once replaced, it cannot return, so $X_i \supseteq X_{i+1} \supseteq \ldots \supseteq X_m$. If it is still the sample before round $j > i$, it survives unless the new item is selected. Thus $\Pr\left[X_j \mid X_{j-1}\right] = 1 - 1/j = (j-1)/j$, and

$$\Pr[X_m] = \frac{1}{i} \cdot \frac{i}{i+1} \cdot \frac{i+1}{i+2} \cdots \frac{m-1}{m} = \frac{1}{m}.$$

Since i was arbitrary, the sample is uniform among the first m items. The algorithm uses $O(1)$ space and $O(1)$ time per arrival.

(b) Design and analyze an algorithm to sample k elements uniformly at random without replacement from the stream; you may use $O(k)$ space. If fewer than k items arrive, keep all of them.

Solution. The algorithm is as follows:

1. Store the first k items in an array called the reservoir.
2. For each subsequent item e_j, accept it with probability k/j. If accepted, replace a uniformly chosen item in the reservoir with e_j.

Fix an item e_i, and let X_j be the event that it is in the reservoir after round j. At a later round $j > \max(i, k)$, it is replaced only if the new item is accepted and its slot is chosen. Conditional on X_{j-1}, this happens with probability $(k/j) \cdot (1/k) = 1/j$, so it survives with probability $(j-1)/j$. As in part (a), an item cannot return once replaced, so the events are nested.

If $i \leq k$, the item is in the initial reservoir, so $\Pr[X_k] = 1$. Multiplying the subsequent survival probabilities gives

$$\Pr[X_m] = \frac{k}{k+1} \cdot \frac{k+1}{k+2} \cdots \frac{m-1}{m} = \frac{k}{m}.$$

If $i > k$, the item enters at round i with probability k/i, giving

$$\Pr[X_m] = \frac{k}{i} \cdot \frac{i}{i+1} \cdot \frac{i+1}{i+2} \cdots \frac{m-1}{m} = \frac{k}{m}.$$

Thus, in either case, every item is included with probability k/m (if $m < k$, every item is included deterministically). The array uses $O(k)$ space, and each arrival takes $O(1)$ time.