Simple Random Sampling from Countable Streams

Stephane Moore

April 6, 2011

Problem Statement

Given a countable stream of unknown length, how can we select a single element out of the stream with uniform probability distribution?

1] Naive Approach

The simplest way to tackle this problem is to exhaustively accumulate the contents of the stream and then randomly select an element from your data structure with uniform distribution. For the stream $A = \langle a_1, ..., a_n \rangle$,where $n \in \mathbb{N}^*$, this can be represented as follows:

$$\begin{aligned} v &= \langle a_1, ..., a_n \rangle \\ P(v_i) &= \begin{cases} \frac{1}{n} & \text{for } i \in \mathbb{N}^*, i \in [1, n] \\ 0 & \text{otherwise} \end{cases} \\ \sum_{i=1}^{n} P(v_i) &= n\left(\frac{1}{n}\right) = 1 \end{aligned}$$

For a stream of n objects, storage of v requires $O(n)$ memory. This can be acceptable in a number of scenarios; however, for extremely large values of n, this approach can be costly.

2] Iterative Approach

One way of avoiding the need to store the entire stream is to update selection state after each iteration through the stream. For the stream $A = \langle a_1, ..., a_n \rangle$, where $n \in \mathbb{N}^*$, consider the function $F : \{i \in \mathbb{N}^* : i \in [1, n]\} \rightarrow \{a | a \in A\}$:

$$F(i) = \begin{cases} F(i-1) & \text{if } X \sim U(0, i) < i - 1 \\ a_i & \text{otherwise} \end{cases}$$

Now consider this informal proof that $F(k)$ selects an element from $\langle a_1, ..., a_k \rangle$ with a uniform probability distribution:
Proof. (by Induction)
Base Case: When $k = 1$, $F(1)$ selects a_1 with probability 1 because $X \sim U(0, 1) \geq 0$. This means that $F(1)$ selects an element from $\langle a_1 \rangle$ with uniform probability distribution.

Inductive Step: If $F(k)$ selects an element from $\langle a_1, ..., a_k \rangle$ with uniform probability distribution, then the following is true:

$$\begin{aligned}
P(F(k) = a_i) &= \begin{cases} \frac{1}{k} & \text{for } i \in \mathbb{N}^*, i \in [1, k] \\ 0 & \text{otherwise} \end{cases} \\
P(F(k+1) = a_i) &= \begin{cases} \int_k^{k+1} U(0, k+1)\ dx & \text{for } i \in \mathbb{N}^*, i = k+1 \\ (\int_0^k U(0, k+1)\ dx)(P(F(k) = a_i)) & \text{for } i \in \mathbb{N}^*, i \in [1, k] \\ 0 & \text{otherwise} \end{cases} \\
&= \begin{cases} \frac{1}{k+1} & \text{for } i \in \mathbb{N}^*, i = k+1 \\ \frac{k}{k+1}(P(F(k) = a_i)) & \text{for } i \in \mathbb{N}^*, i \in [1, k] \\ 0 & \text{otherwise} \end{cases} \\
&= \begin{cases} \frac{1}{k+1} & \text{for } i \in \mathbb{N}^*, i = k+1 \\ \frac{k}{k+1}(\frac{1}{k}) & \text{for } i \in \mathbb{N}^*, i \in [1, k] \\ 0 & \text{otherwise} \end{cases} \\
&= \begin{cases} \frac{1}{k+1} & \text{for } i \in \mathbb{N}^*, i = k+1 \\ \frac{1}{k+1} & \text{for } i \in \mathbb{N}^*, i \in [1, k] \\ 0 & \text{otherwise} \end{cases} \\
&= \begin{cases} \frac{1}{k+1} & \text{for } i \in \mathbb{N}^*, i \in [1, k+1] \\ 0 & \text{otherwise} \end{cases} \\
\sum_{i=1}^{k} P(F(k+1) = a_i) &= (k+1)\left(\frac{1}{k+1}\right) = 1
\end{aligned}$$

Therefore if $F(k)$ selects an element from $\langle a_1, ..., a_k \rangle$ with uniform probability distribution, then $F(k+1)$ selects an element from $\langle a_1, ..., a_{k+1} \rangle$ with uniform probability distribution.
From the base case and the inductive step, it follows that for any $n \in \mathbb{N}^*$, $F(n)$ selects an element from $\langle a_1, ..., a_n \rangle$ with uniform probability distribution.

3] Comparison

The naive approach requires $O(1)$ operations per selection and $O(n)$ memory to store the contents of the stream. Meanwhile, the iterative approach requires $O(n)$ operations to make a single selection but also requires $O(1)$ memory per selection. The two approaches have distinct advantages and are appropriate for different scenarios. The described iterative approach is useful in situations where the number of required selections is much smaller than n. As the number of needed selections approaches n, the benefits of using the iterative approach diminish as the memory requirements of the selection states approach the memory required to store the entire stream.