Notes on the Alias Method for Sampling from Discrete Distribution

Xuecheng Liu

March 16, 2021

1 The Alias Method

See Algorithm 1.

2 Efficiency

The initialization time is $\Theta(n)$, the generation time is $\Theta(1)$, the memory usage is $\Theta(n)$. See for the implementation.

Algorithm 1: Vose's Alias Method

Input: A discrete distribution $\mathbf{p} = (p_0, p_1, \ldots, p_{n-1})$ on the support $\mathcal{I} = \{0, 1, \ldots, n-1\}$

Output: Generate a random number $i \in \mathcal{I}$ with probability p_i.

`/* Initialization */`

Create two arrays Alias and Prob, each of size n;
Create two worklists, Small and Large;
Multiply each probability by n;
foreach *scaled probability* p_i **do**
    if $p_i < 1$ **then**
        Add i to Small;
    else
        Add i to Large;
    end
end
while Small *and* Large *are not empty* **do**
    $l \leftarrow$ Remove the first element from Small;
    $g \leftarrow$ Remove the first element from Large;
    Prob$[l] \leftarrow p_l$;
    Alias$[l] \leftarrow g$;
    $p_g \leftarrow (p_g + p_l) - 1$;
    if $p_g < 1$ **then**
        Add g to Small;
    else
        Add g to Large;
    end
end
while Large *is not empty* **do**
    $g \leftarrow$ Remove the first element from Large;
    Prob$[g] \leftarrow 1$;
end
while Small *is not empty* **do** This is only possible due to numerical instability
    $l \leftarrow$ Remove the first element from Small;
    Prob$[l] \leftarrow 1$;
end

`/* Generation */`

Side $i \leftarrow$ Generate a fair die roll from an n-sided die;
Flip a biased coin that comes up heads with probability Prob$[i]$;
if *the coin comes up "heads"* **then**
    return i;
else
    return Alias$[i]$.
end