The Alias Method

Isaac De Vlugt

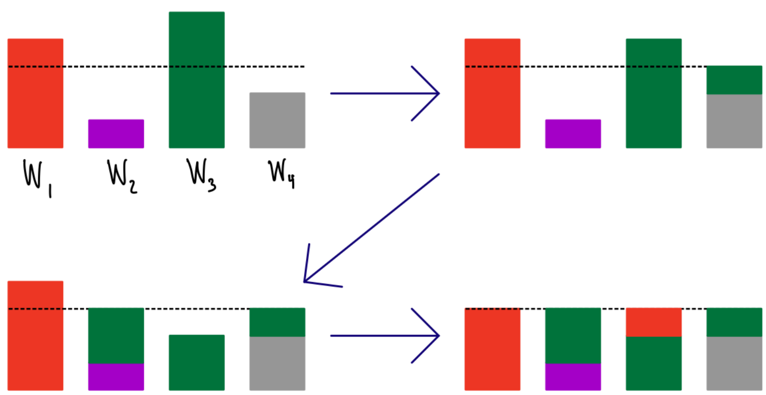
October 27, 2021

Without question, the fastest distributions to sample are uniform distributions $p = [1/N, 1/N, \cdots, 1/N]$, where N is the size of the probability distribution. Clearly, sampling from p would take $\mathcal{O}(1)$ time,

```
result = int(numpy.random.uniform() * N),
```

since each outcome has probability $1/N$. A general discrete distribution $p = [p_1, p_2, \cdots, p_N]$ clearly won't be this efficient to sample, as a random number drawn from $[0, 1]$ has to be compared to almost N numbers to obtain a result. However, what if we can manipulate a general discrete distribution into being uniform? If each probability density that is greater than the uniform value, we can shuffle around this "excess" probability density to the lower-valued probabilities. Then, to ensure that we are still sampling from the original distribution, if we randomly choose an integer $[1, N]$, we then draw a random number $x \in [0, 1]$ and draw from a biased binary distribution. Let's give an example.

Consider this length-four distribution $p = [W_1 = 1/3, W_2 = 1/12, W_3 = 5/12, W_4 = 1/6]$ pictured below, where the black dashed line represents the uniform distribution case ($W_{1-4} = 1/4$).



Clearly, W_1 and W_3 are *over* the uniform-distribution line, while W_2 and W_4 are *under*. Let's start by telling W_3 to give some of its probability density $\tilde{W}_3 = 1/12$ to W_4 such that $W_4 + 1/12 = 1/4$ (follow the first arrow). $W_3 - \tilde{W}_3$ is stil over $1/4$, but we "fixed" W_4.

So, let's demand again that $W_3 - \tilde{W}_3 = 1/3$ share some of its probability density $\tilde{W}_3' = 1/6$ to the other under-valued probability density, W_2, such that $W_2 + \tilde{W}_3' = 1/4$. Unfortunately, in doing so, now $W_3 - \tilde{W}_3 - \tilde{W}_3' = 1/6$ is *under* the uniform value. Luckily, W_1 is still over, and it can replenish what is missing (follow the last arrow).

In the bottom right pane, we now have a uniform distribution wherein each event is itself a binary distribution. So, to sample from this "distribution of distributions" while ensure that we're sampling from the original distribution in the top left pane, we do the following.

1. Select a random integer in $[1, N]$. The chosen integer tells us which of the N binary distributions we sample from.

2. Now choose a random number in $[0, 1]$. This number is compared to the probability that defines the binary distribution, which results in our sample.

Although the operations required to perform what is pictured in the above example still scale unfavourably with N, this procedure only needs to be done *once*. Then, in uniformly sampling an integer and then a binary distribution, generating samples from the same distribution scales as $\mathcal{O}(1)$. In essence, we've traded unfavourable scaling to draw a single sample in favour of an initial unfavourable overhead with constant-time sampling.