MTH 511a - 2020: Lecture 3

Instructor: Dootika Vats

1 Generating Discrete Random Variables

Suppose X is a discrete random variable having probability mass function

$$\Pr(X = x_j) = p_j \quad j = 0, 1, \dots, \quad \sum p_j = 1\,.$$

Examples if such random variables are: Bernoulli, Poisson, Geometric, Negative Binomial, Binomial, etc. We will learn four methods to draw samples realizations of this discrete random variable

1. Inverse transform method
2. The acceptance-rejection technique
3. The composition approach
4. Bernoulli factories

1.1 Inverse transform method

We will demonstrate this method with an example first.

Example 1 (Bernoulli distribution). If $X \sim \text{Bern}(p)$, then

$$\Pr(X = 0) = q = 1 - p \text{ and } \Pr(X = 1) = p.$$

Let $U \sim U[0, 1]$. Define

$$X = \begin{cases} 0 & \text{if } U < q \\ 1 & \text{if } q \leq U \leq 1 \end{cases}.$$

Then $X \sim \text{Bern}(p)$.

Proof. To show the result we only need to show that $\Pr(X = 1) = p$ and $\Pr(X = 0) = 1-p$. Recall that by the cumulative distribution function of $U[0, 1]$, for any $0 < t < 1$m $\Pr(U < t) = t$. Using this,

$$\Pr(X = 0) = \Pr(U < q) = q\,,$$

and also

$$\Pr(X = 1) = \Pr(q \leq U < 1) = 1 - q = p\,.$$

□

Algorithm 1 Inverse transform for $\text{Bern}(p)$

```
Draw U ~ U[0,1]
if U < q then X = 0 else X = 1
```

Inverse transform method: The principles used in the above example can be extended to any generic discrete distribution. For a distribution with mass function

$$\Pr(X = x_j) = p_j \qquad \text{for } j = 0, 1, \dots \qquad \text{with} \qquad \sum p_j = 1\,.$$

Let $U \sim U[0, 1]$. Set X to be

$$X = \begin{cases} x_0 & \text{if } U < p_0 \\ x_1 & \text{if } p_0 \leq U < p_0 + p_1 \\ x_2 & \text{if } p_0 + p_1 \leq U < p_0 + p_1 + p_2 \\ \vdots & \\ x_j & \text{if } \sum_{i=0}^{j-1} p_i \leq U < \sum_{i=0}^{j} p_i \end{cases}.$$

This works because

$$\Pr(X = x_j) = \Pr\left(\sum_{i=0}^{j-1} p_i \leq U < \sum_{i=0}^{j} p_i\right) = \sum_{i=0}^{j} p_i - \sum_{i=0}^{j-1} p_i = p_j\,.$$

This method is called the *Inverse transform method* since the algorithm is essentially looking at the inverse cumulative distribution function of the random variable.

Example 2 (Poisson random variables). The probability mass function for the Poisson random variable is

$$\Pr(X = i) = p_i = \frac{e^{-\lambda}\lambda^i}{i!} \quad i = 0, 1, 2, \ldots,$$

Algorithm 2 Inverse transform for Poisson(λ)

Draw $U \sim U[0, 1]$
if $U < p_0$ **then**
    $X = 0$
else if $U < p_0 + p_1$ **then**
    $X = 1$
    $\ldots$
else if $U < \sum_{i=1}^{j} p_i$ **then**
    $X = j$
    $\ldots$

Algorithm 2 implements the inverse transform method for Poisson(λ). The above algorithm can be written more neatly using an iterative structure. Note that

$$\Pr(X = i + 1) = \frac{e^{-\lambda}\lambda^{i+1}}{(i + 1)!} = \frac{\lambda}{i + 1}\frac{e^{-\lambda}\lambda^i}{i!} = \frac{\lambda}{i + 1}\Pr(X = i)\,.$$

Algorithm 3 Iterative version of inverse transformation for Poisson(λ)

1: Draw $U \sim U[0, 1]$
2: Set $i = 0, p = e^{-\lambda}, A = p$
3: **if** $U < A$ **then**
4:     $X = i$ and stop
5: **else**
6:     Set $p = \frac{\lambda}{i + 1} p, A = A + p$, and $i = i + 1$ goto Step 3

However, Algorithm 2 outlines a challenge in implementing this algorithm.

Q. What happens when λ is large?

A Poisson(λ) distribution with a large λ will yield p_j to be small when j is small. This implies Algorithm 2 can be quite slow here. We will therefore discuss a more computationally efficient algorithm.

We know that most likely, a realization from Poisson will be closer to λ, so it will be beneficial to start from around λ. Set $I = \lfloor \lambda \rfloor$, and check whether

$$\sum_{i=0}^{I-1} p_i < U < \sum_{i=0}^{I} p_i \,.$$

If it is, then return $X = I$. Else, if $U > \sum_{i=1}^{I} p_i$, then increase I, otherwise, decrease I and check again.

1.1.1 Questions to think about

- What other example can you think of where the inverse transform method could take a lot of time?
- Can you try and implement this for a Binomial random variable?

References