Algorithmic Analysis

In this section we are going to use probability to analyze code. Specifically we are going to be calculating expectations on code: expected run time, expected resulting values etc. The reason that we are going to focus on expectation is that it has several nice properties. One of the most useful properties that we have seen so far is that the expectation of a sum, is the sum of expectations, *regardless* of whether the random variables are independent of one another. In this section we will see a few more helpful properties, including the Law of Total Expectation, which is also helpful in analyzing code:

Law of Total Expectation
The law of total expectation gives you a way to calculate $\mathrm{E}[X]$ in the scenareo where it is easier to compute $\mathrm{E}[X|Y=y]$ where Y is some other random variable:

$$\begin{aligned}\mathrm{E}[X] &= \mathrm{E}\left[E[X|Y]\right] \\ &= \sum_y \mathrm{E}[X|Y=y]\,\mathrm{P}(Y=y)\end{aligned}$$

Distributed File System

Imagine the task of loading a large file from your computer at Stanford, over the interet. Your file is stored in a distributed file system. In a distributed file system, the closest instance of your file might be on one of several computers at different locations in the world. Imagine you know the probability that the file is in one of a few locations l: $\mathrm{P}(L=l)$, and for each location the expected time, T, to get the file, $\mathrm{E}[T|L=l]$, given it is in that location:

Location	$P(L=l)$	$\mathrm{E}[T\|L=l]$
SoCal	0.5	0.3 seconds
New York	0.2	20.7 seconds
Japan	0.3	96.3 seconds

The Law of Total Expectation gives a straightforward way to compute $\mathrm{E}[T]$:

$$\begin{aligned}\mathrm{E}[T] &= \sum_l \mathrm{E}[T|L=l]\,\mathrm{P}(L=l) \\ &= 0.5\cdot 0.3 + 0.2\cdot 20.7 + 0.3\cdot 96.7 \\ &= 33.3 \text{ seconds}\end{aligned}$$

Toy Example of Recursive Code

In theoretical computer science, there are many times where you want to analyze the expected runtime of an algorithm. To practice this technique let's try to solve a simple recursive function. Let Y be the value returned by `recurse()`. What is $E[Y]$?

```
def recurse():
    x = random.choice([1, 2, 3])  # Equally likely values
    if x == 1:
        return 3;
    else if (x == 2):
        return 5 + recurse()
    else:
        return 7 + recurse()
```

In order to solve this problem we are going to need to use the law of total expectation considering X as your background variable.

$$\begin{aligned}\mathrm{E}[Y] = &\sum_{i \in \{1,2,3\}} \mathrm{E}[Y|X=i]\,\mathrm{P}(X=i) \\ =&\mathrm{E}[Y|X=1]\,\mathrm{P}(X=1) \\ &+ \mathrm{E}[Y|X=2]\,\mathrm{P}(X=2) \\ &+ \mathrm{E}[Y|X=3]\,\mathrm{P}(X=3)\end{aligned}$$

We know the probability $\mathrm{P}(X=x) = \frac{1}{3}$ for $x \in \{1,2,3\}$. How can we compute a value such as $\mathrm{E}[Y|X=2]$? Well that is the expectation of your return, in the world where $X=2$. In that case you will return 5 + recurse(). The expectation of that is $5 + \mathrm{E}[Y]$. Plugging a similar result for each case we can continue our solution:

$$\begin{aligned}E[Y|X=1] &= 3 \\ E[Y|X=2] &= 5 + \mathrm{E}[Y] \\ E[Y|X=3] &= 7 + \mathrm{E}[Y]\end{aligned}$$

Now we can just plug values into the law of total expectation:

$$\begin{aligned}\mathrm{E}[Y] = &\sum_{i \in \{1,2,3\}} \mathrm{E}[Y|X=i]\,\mathrm{P}(X=i) \\ =&\mathrm{E}[Y|X=1]\,\mathrm{P}(X=1) \\ &+ \mathrm{E}[Y|X=2]\,\mathrm{P}(X=2) \\ &+ \mathrm{E}[Y|X=3]\,\mathrm{P}(X=3) \\ =&3 \cdot 1/3 \\ &+ (5 + \mathrm{E}[Y]) \cdot 1/3 \\ &+ (7 + \mathrm{E}[Y]) \cdot 1/3 \\ =&(15 + 2\,\mathrm{E}[Y]) \cdot 1/3 \\ 1/3 \cdot \mathrm{E}[Y] =&5 \\ \mathrm{E}[Y] =&15\end{aligned}$$