Radiation Detection and Measurement Lab

Micheal Jones

September 24, 2018

Binomial Probability

$$P(k,n,p) = \frac{n!}{(n-k)!k!}\, p^k(1-p)^{n-k}$$

Normalization: For n total trials, there exist k independent possible number of successes. The sum of the probability of obtaining each individual possible result, or number of successes k up to and including n, must include all possible results, and therefore must add up to 1 or 100%. Hence,

$$\sum_{k=0}^{n} P(k,n,p) = 1.$$

$$\sum_{k=0}^{n} P(k,n,p) = (1-p)^n + \frac{n!}{(n-1)!}\, p(1-p)^{n-1} + \frac{n!}{(n-2)!\,2}\, p^2(1-p)^{n-2} + \ ... \ + p^n = 1.$$

Mean: For n total trials, the average result or expected value must be equal to the sum of all possible number of successes, k, weighted by the probability of their occurrence. The most likely events have the highest probability and therefore contribute most to the expectation value. Hence,

$$\mu = \sum_{k=0}^{n} kP(k,n,p) = np.$$

To show this explicitly we factor out n and p, cancel one factor of k, and sum over the remaining equation as follows,

$$\mu = \sum_{k=1}^{n} k\frac{n!}{(n-k)!k!}\, p^k(1-p)^{n-k} = np\sum_{k=1}^{n} \frac{(n-1)!}{(n-k)!(k-1)!}\, p^{k-1}(1-p)^{n-k}$$

Now, seeing (n-1) - (k-1) = (n-k) and applying the Binomial Theorem for (n-1) and (k-1),

$$np\sum_{k=1}^{n} \frac{(n-1)!}{(n-k)!(k-1)!}\, p^{k-1}(1-p)^{n-k} = np\sum_{k=1}^{n} \frac{(n-1)!}{((n-1)-(k-1))!(k-1)!}\, p^{k-1}(1-p)^{(n-1)-(k-1)} = np$$

Variance and Standard Deviation: The variance can be derived from the second moment about the mean for the binomial distribution,

$$\sigma^2 = \sum_{k=0}^{n} (k - \bar{k})^2 \, P(k, n, p) = E(k^2) - E(k)^2 = np(1-p)$$

To show this result explicitly we begin with a similar approach to the proof for the mean value, beginning with $E(k^2)$,

$$\sigma^2 = \sum_{k=0}^{n} k^2 \frac{n!}{(n-k)!k!} \, p^k (1-p)^{n-k} = np \sum_{k=1}^{n} ((k-1)+1) \frac{(n-1)!}{((n-1)-(k-1))!(k-1)!} \, p^{k-1} (1-p)^{(n-1)-(k-1)}$$

Now let $m = n - 1$ and $j = k - 1$, then split the summation,

$$np \sum_{j=0}^{m} (j+1) \frac{m!}{(m-j)!\, j!} \, p^j (1-p)^{m-j} = np \left(\sum_{j=0}^{m} j \frac{m!}{(m-j)! j!} \, p^j (1-p)^{m-j} + \sum_{j=0}^{m} \frac{m!}{(m-j)!\, j!} \, p^j (1-p)^{m-j} \right)$$

Applying a similar treatment to the first sum we obtain,

$$np \left(\sum_{j=0}^{m} m \frac{(m-1)!}{((m-1)-(j-1))!(j-1)!} \, p^j (1-p)^{m-j} + \sum_{j=0}^{m} \frac{m!}{(m-j)!\, j!} \, p^j (1-p)^{m-j} \right) \; =$$

$$np \left(\sum_{j=1}^{m} p(n-1) \frac{(m-1)!}{((m-1)-(j-1))!(j-1)!} \, p^{(j-1)} (1-p)^{(m-1)-(j-1)} + \sum_{j=0}^{m} \frac{m!}{(m-j)!\, j!} \, p^j (1-p)^{m-j} \right)$$

Applying binomial theorem and noticing that (p + q) = 1,

$$np \; (p \; (n-1) \; (p+q)^{(m-1)} \; + \; (p+q)^m) = np \; (p \; (n-1) + 1) = n^2 p^2 + np \; (1-p)$$

Now addressing the square of the expectation value and using the previous result we obtain,

$$n^2 p^2 + np \; (1-p) + (np)^2 \; = \; np \; (1-p).$$

The standard deviation is simply the square root of this value,

$$\sqrt{\sigma^2} = \sigma.$$

Poisson Distribution Probability

$$P(k, \lambda) = e^{-\lambda} \frac{\lambda^k}{k!}$$

Mean For a Poisson ditribution, the expectation value is equal to the rate over a specified time interval. If the average is k events over some t time interval, this is not only the expected reult for any period of time equal to your time interval it is also a rate which can be applied to larger or shorter spans of time and is associated with a Poisson probability distribution. Therefore the mean or expectation value is defined as follows,

$$\mu = \sum_{k=0}^{\infty} kP(k,\lambda) = \lambda$$

Showing this result explicitly will be similar to the process for the binomial distribution but utilize a Taylor series approximation as follows,

$$\mu = \sum_{k=0}^{\infty} ke^{-\lambda}\frac{\lambda^k}{k!} = \lambda e^{-\lambda}\sum_{k=0}^{\infty}\frac{\lambda^{k-1}}{(k-1)!} = \lambda e^{-\lambda}e^{\lambda} = \lambda$$

```
In [1]: # This code block imports packages for use in answering the questions

        import numpy as np
        import matplotlib.pyplot as plt
        % matplotlib inline
```

```
In [2]: # Here I define the functions I'll need

        def poissonDist(x, rate):
             return np.exp(-rate)*rate**x/np.math.factorial(x)

        def binomialDist(x, n, p):
            return (np.math.factorial(n)/(np.math.factorial(n-x)*np.math.factorial(x)))*(p**x)*(

        def binomialDist2(x, n):
            return (np.math.factorial(n)/(np.math.factorial(n-x)*np.math.factorial(x)))

        def stirlingApprox(x, n):
            return (n**x)/np.math.factorial(x)
```

1.) Calculate the following binomial probabilities: P(2,5,0.3), P(4,5,0.3), P(4,10,0.3), and P(4,10,0.9)

```
In [77]: a, b, c, d = binomialDist(2, 5, 0.3) , binomialDist(4, 5, 0.3) , binomialDist(4, 10, 0.
         print(' P(2,5,0.3) =', a, '\n P(4,5,0.3) =', b, '\n P(4,10,0.3) =', c, '\n P(4,10,0.9)
```

```
P(2,5,0.3) = 0.30870
P(4,5,0.3) = 0.02835
P(4,10,0.3) = 0.20012
P(4,10,0.9) = 0.00014
```

$$P(2,5,0.3) = 30.87\% P(4,5,0.3) = 2.83\% P(4,10,0.3) = 20.01\% P(4,10,0.9) = 0.01\%$$

2.) For large n and (n-k) evaluate the error introduce by Stirling's Approximation relative to the exact result. Stirling's method for approximating factorials is as follows:

$$n! \approx \sqrt{2\pi n}\,(\frac{n}{e})^n.$$

This yields the following interpretation of the binomial probability function:

$$P(k,n,p) \approx \frac{n^k}{k!}p^k(1-p)^{n-k}.$$

I will plot,

$$f(k,n) \approx \frac{n^k}{k!}$$

versus

$$g(k,n) = \frac{n!}{(n-k)!k!}$$

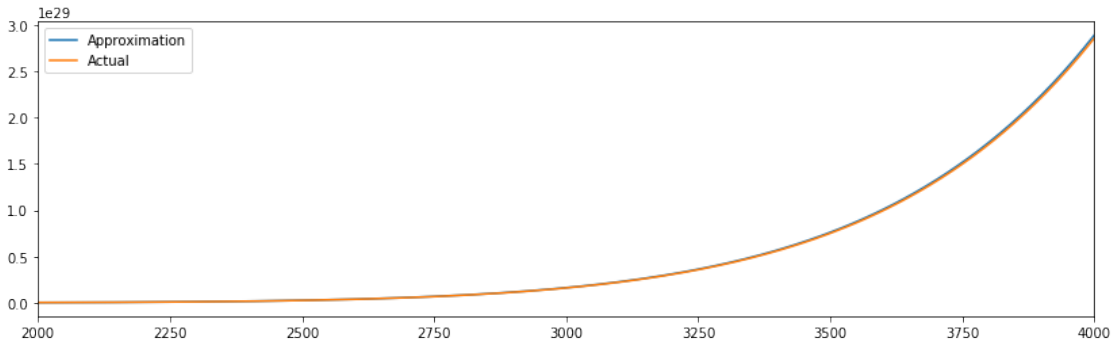
in order to focus on the portion of the equation affected by the approximation.

```
In [58]: aprx = np.zeros(400)
         actl = np.zeros(400)
         lrg = np.linspace(10,4000,400)

         i=0
         for i in range (0,len(aprx)):
             aprx[i] = stirlingApprox(10,lrg[i])
             actl[i] = binomialDist2(10,lrg[i])

         plt.figure(figsize=(14,4))
         plt.plot(lrg, aprx, label = 'Approximation')
         plt.plot(lrg, actl, label = 'Actual')
         plt.xlim(2000,4000)
         plt.legend(loc='upper left')
         plt.show()

         f, g = np.math.factorial(10), (np.sqrt(2*3.1415*10)*(10*np.exp(-1))**10)
         h = f - g
         perc = h/f*100
         print('Exact=',f, ', Estimated=', g, ', Difference=', h, ', Percent Error=', perc, '%')
```



```
Exact= 3628800 , Estimated= 3598642.551 , Difference= 30157.449 , Percent Error= 0.83106 %
```

As shown above a plot of the evaluation of the approximation at high values is quite close to the expected value. An evaluation of the approximation when n is 10 yields the following,

$$10! = 3628800$$

$$\sqrt{2\pi(10)}\,(\frac{10}{e})^{10} \approx 3598642.551$$

$$\frac{3628800 - 3598642.551}{3628800} \approx 0.0083 = 0.83\%$$

So, at lower values the error is less than one percent, and evaluating this at higher values, such as when n is 100, only increases the relative accuracy of this estimate despite the increasing difference between the actual and estimated values.

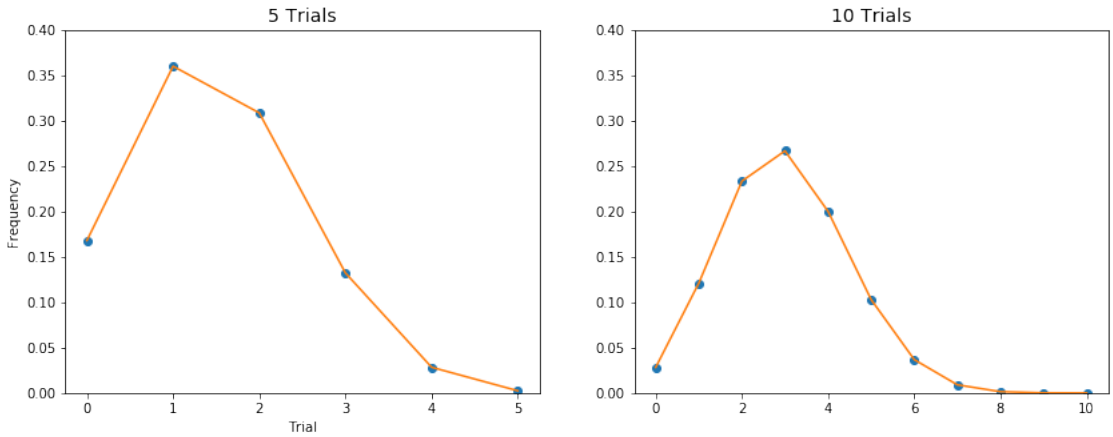
3.) Plot P(k, 5, 0.3) and P(k, 10, 0.3).

```
In [50]: trials5 = np.arange(0,6,1)
         trials10 = np.arange(0,11,1)
         prob5 = np.zeros(len(trials5))
         prob10 = np.zeros(len(trials10))

         i=0
         for i in range (0,len(trials5)):
             prob5[i] = binomialDist(trials5[i], 5, 0.3)

         j=0
         for j in range (0,len(trials10)):
             prob10[j] = binomialDist(trials10[j], 10, 0.3)

         plt.figure(figsize=(14,5))
         plt.subplot(121)
         plt.plot(trials5, prob5, 'o')
         plt.plot(trials5, prob5)
         plt.ylim(0,0.4)
         plt.title('5 Trials', size='14')
         plt.ylabel('Frequency')
         plt.xlabel('Trial')
         plt.subplot(122)
         plt.plot(trials10, prob10, 'o')
         plt.plot(trials10, prob10)
         plt.title('10 Trials', size='14')
         plt.ylim(0,0.4)
         plt.show()
```



4.) Obtain the formula for the 2nd and 3rd moment of P(k, n, p) about the mean. The n-th moment about some centralized value, c, for some function, f(x), is as follows,

$$\mu_i = \int_{-\infty}^{\infty} (x - c)^i \, f(x) \, dx$$

For our purposes c is the mean and the function is P(k, n, p).

$$\mu_i = \int_{-\infty}^{\infty} (k - \bar{k})^n \, P(k, n, p) \, dk = \sum_{k=0}^{n} (k - \bar{k})^i \, P(k, n, p)$$

Calculating the second moment we obtain,

$$\mu_2 = \int_{-\infty}^{\infty} (k - \bar{k})^2 \, P(k, n, p) \, dk = \sum_{k=0}^{n} (k - \bar{k})^2 \, P(k, n, p)$$

Which is equivalent to the variance. Calculating the formula for the third moment we obtain,

$$\mu_3 = \int_{-\infty}^{\infty} (k - \bar{k})^3 \, P(k, n, p) \, dk = \sum_{k=0}^{n} (k - \bar{k})^3 \, P(k, n, p)$$

Which is equivalent to the skewness of the distribution.

5.) Plot both the Poisson and binomial probability for n = 10, 20, 100 with an expectation value of 5.

```
In [34]: t10 = np.arange(0,11,1)
         t20 = np.arange(1,21,1)
         t100 = np.arange(1,101,1)
         p10 = np.zeros(len(t10))
         p20 = np.zeros(len(t20))
         p100 = np.zeros(len(t100))
         bp10 = np.zeros(len(t10))
         bp20 = np.zeros(len(t20))
```

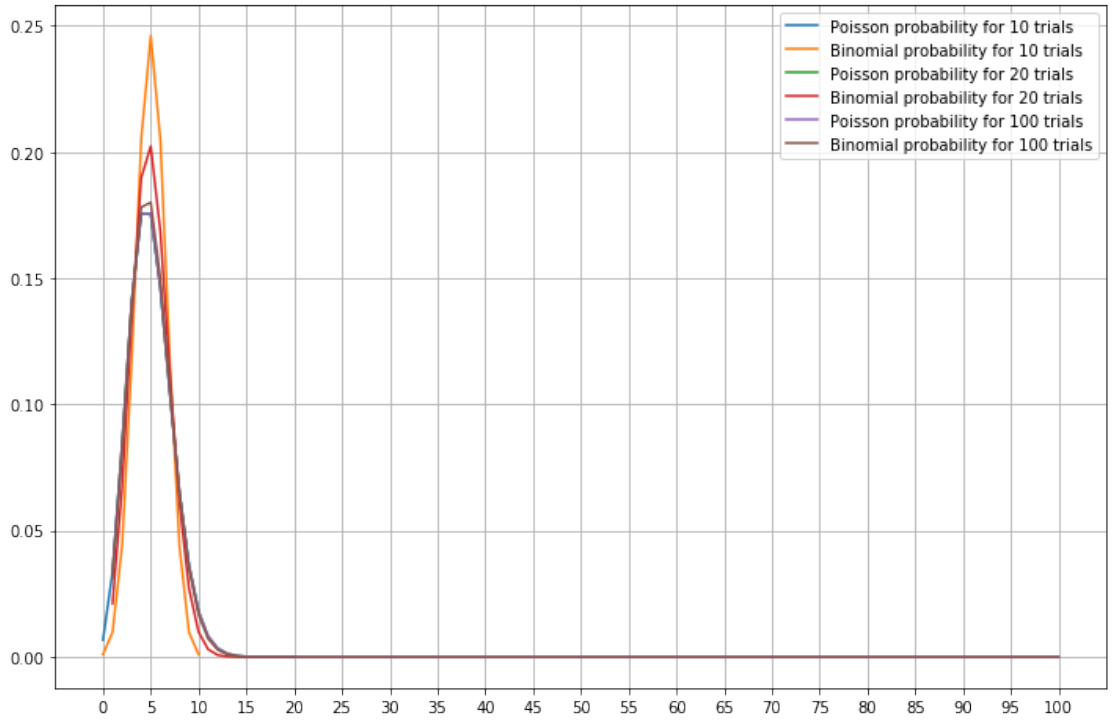
```
bp100 = np.zeros(len(t100))

i=0
for i in range (0,len(t10)):
    p10[i] = poissonDist(t10[i], 5)
    bp10[i] = binomialDist(t10[i], 10, 0.5)

j=0
for j in range (0,len(t20)):
    p20[j] = poissonDist(t20[j], 5)
    bp20[j] = binomialDist(t20[j], 20, 0.25)

k=0
for k in range (0,len(t100)):
    p100[k] = poissonDist(t100[k], 5)
    bp100[k] = binomialDist(t100[k], 100, 0.05)

plt.figure(figsize=(12,8))
plt.plot(t10,p10, label='Poisson probability for 10 trials')
plt.plot(t10,bp10, label='Binomial probability for 10 trials')
plt.plot(t20,p20, label='Poisson probability for 20 trials')
plt.plot(t20,bp20, label='Binomial probability for 20 trials')
plt.plot(t100,p100, label='Poisson probability for 100 trials')
plt.plot(t100,bp100, label='Binomial probability for 100 trials')
plt.title('Various Trials', size='14')
plt.ylabel('Frequency')
plt.xlabel('Trial')
plt.xticks(np.arange(0,105,5))
plt.grid(True)
plt.legend()
plt.show()
```



It is notable that all of these methods gave extremely similar results given the same expectation value.