Lab2

October 31, 2019

[2]:
```
%matplotlib inline
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import scipy
from scipy import stats
```

1 Lab #2: Probability, Distributions, and Statistical Questions

Lab Done By: Lexie Peterson
Lab Partner: Kun Lee

1.1 Problem 1

In lecture and homework we explored how the convolution can be used to calculate the probability of a sum or average. For this problem we are going to imagine that we are looking for gamma-ray sources (e.g. with the Fermi telescope). In this kind of telescope there is a background of cosmic-rays (electrons and protons, mostly) that provides a discrete noise term across the sky that precisely follows a Poisson distribution. To detect a gamma-ray source, you need to ask what is the probability that the cosmic-ray background would have given you a measurement as signal-like or more than the signal that you received.

To set up the problem, assume in 1 day the average cosmic-ray background is some number X (pick someting between 0.5 and 10, with different values for you and your lab partner); and the average number of gamma-rays emitted by your hypothetical source is Y (pick something larger than X).

A) Show how the probability distribution of the background changes as you integrate (sum) for more days.

[92]:
```
background = 4 # average cosmics-ray background
gamma = 9      # average number of gamma-rays

x  = np.linspace(0, 20 , 21)
x2 = np.linspace(0,40,41)

base  = stats.poisson.pmf(x, background, loc=0) #creating background␣
 ↪probability distribution
base2 = np.convolve(base,base)
```

```
fig,ax = plt.subplots(1,2)
fig.set_size_inches(11,8.5)

ax[0].step(x,base) #plotting initial distribtuion
ax[0].set_title("Initial Poisson Probability Distribution")
ax[0].set_xlabel('Number of Event Occurances')
ax[0].set_ylabel('Probability')

ax[1].step(x2,base2) #plotting distribution summed over two days
ax[1].set_title('Poisson Probability Distribution Summed Over Two Days')
ax[1].set_xlabel('Number of Event Occurances')
ax[1].set_ylabel('Probability');
```

After plotting the initial distribution and the distribution summed over two days, we see that the mean doubled (shifted right), and the probability decreased as the distribution spread out more.

B) Show that after 5 days, the summed probability distribution is still a Poisson distribution. Explain why this makes sense from a mathematical and conceptual point of view.

[93]:
```
x5 = np.linspace(0,100,101)

base  = stats.poisson.pmf(x, background, loc=0)
base2 = np.convolve(base,base)
base3 = np.convolve(base2,base)
base4 = np.convolve(base3,base)
base5 = np.convolve(base4,base)

fig,ax = plt.subplots(1,1)
fig.set_size_inches(11,8.5)

fig.tight_layout()
ax.step(x5,base5)
ax.set_title('Probability Distribution Summed Over 5 Days')
ax.set_xlabel('Number of Event Occurances')
ax.set_ylabel('Probability');
```

A poisson distribution shows the probability of a given number of events occurring in a fixed interval of time if these events occur with a known constant rate. In this case we are measuring the number of gamma-ray sources hitting our detector given a rate of nine events per day. If we measured over

multiple days, we would expect the number of total events occuring to increase (as more days allows for more events to happen), and the probability of the mean value of events should decrease as there are more possiblities for events to occur. Dispite the change from day to day, the distribution is still poisson because it still describes a the probability of events to occur given a rate. Mathematically, we just multipled a poisson by five times, so the shape shouldn't change just the amount of events.

C) Show how the probability distribution evolves as you average days. Calculate for many different ranges of days, and explore the shape of the distribution as the number of days becomes larger. Discuss this in relation to both B) and the central limit theorem.

[94]:
```
fig,ax = plt.subplots(1,1)
fig.set_size_inches(11,8.5)

fig.tight_layout()

ave = base
for day in range(1,10):
    x = np.linspace(0,10,ave.size)
    ax.step(x, ave, label=day)
    ave = np.convolve(ave,base)

ax.set_title('Average Probability Distribution of the Background')
plt.legend(title="Days Averaged Over:");
```

The central limit theorem says that convolving distributions with itself over and over again results in a gaussian distribution. From a graph above, we can see that this is true. The blue is the result of averaging the first day, and thus still looks a like a poisson distribution. However, after nine days (gold plot above), the distribution looks much more gaussian in shape (looks like a bell). In part B, we saw a similar tend as we added more days. The distribution started to look more gaussian. However, unlike the result of averaging the days, the central humped moved right. This is expected because it is a sum, adding more events instead of finding the average of all of the days.

D) Pick some number of days N, and assume you saw Y*N gamma rays from your source. Calculate the 'sigma' of your observation. [In reality the number of gamma-rays seen from a source will also fluctuate, but we're going to ignore that complication for a couple of labs.]

[95]:
```
days = 3 #number of days
rays = days * gamma #amount of gamma rays seen from source

x = np.linspace(0, 2 * rays , 2 * rays + 1)
distribution = stats.poisson.pmf(x, background, loc=0)
average = distribution
```

```
for day in range(1,days):
    average = np.convolve(distribution, average)

probability = sum(average[0:rays])
sigma = stats.norm.ppf(probability)

sigma
```

[95]: 3.6456664375361654

It turns out after three days, the significance of my observation is only a sigma of 3.645, which does not reach the 5-sigma value physicist strive for.

1.2 Problem 2

Pick a skewed continuous distribtuion, such as a Rayleigh, that describes your background over some observing interval.

A) Show how the distribution changes as you average over more observing intervals.

[45]:
```
plt.rcParams["figure.figsize"] = (10,100)
fig,ax = plt.subplots(10,1)

x = np.linspace(0, 5)
ray_dis = stats.rayleigh.pdf(x, loc=0) #creating an initial rayleigh␣
 ↪distribution
ave = ray_dis

#for loop to convolve distribtuion with itself and plot the results
for day in range(0,10):
    x = np.linspace(0,15,ave.size)
    ax[day].plot(x, ave, label=day)
    ax[day].set_xlim(0,10)
    ax[day].set_xlabel('Events')
    ave = np.convolve(ave,ray_dis)

ax[0].set_title("Initial Rayleigh Distribution")
ax[1].set_title("Rayleigh Distribution Average Over Two Observing Intervals")
ax[2].set_title("Rayleigh Distribution Average Over Three Observing Intervals")
ax[3].set_title("Rayleigh Distribution Average Over Four Observing Intervals")
ax[4].set_title("Rayleigh Distribution Average Over Five Observing Intervals")
ax[5].set_title("Rayleigh Distribution Average Over Six Observing Intervals")
ax[6].set_title("Rayleigh Distribution Average Over Seven Observing Intervals")
ax[7].set_title("Rayleigh Distribution Average Over Eight Observing Intervals")
ax[8].set_title("Rayleigh Distribution Average Over Nine Observing Intervals")
ax[9].set_title("Rayleigh Distribution Average Over Ten Observing Intervals");
```

B) Discuss how the shape changes. Does it approach a Gaussian distribution? If yes, after how many intervals? Initially, the shape of the Rayleigh distribution has a large hump followed by a tail on its right side that falls slower than the tail on the left side, hence it being a skewed distribution. After averaging over two intervals, the hump becomes more prounounced while the right-hand tail assimilates and beings to fall off faster. Averaging over more and more intervals produces a mountain of a hump that has tails that look nearly identical, thus approaching the appearance of a gaussian distribution. At about the 8th averaged interval, the asymetry of the rayleigh distribution is no longer noticable.

1.3 Problem 3

The discovery of optical/infra-red counterparts of Neutron star mergers initially detected with gravity-waves is one of the great discoverys in the last few years (wikipedia, scientific paper), and has ushered in the age of "multi-messenger astrophysics." The science that can be done by matching a gravity-wave signal (directly measures the mass of the neutron stars and their distance) with the optical emisson (redshift, nuclear astrophysics of the resulting explosion) is staggering. Lots of science from how the heaviest elements are formed to constraints on Dark Energy.

We're going to explore one of the analysis questions that comes up when looking for the optical counterparts (a new optical source) of a gravity wave signal. For this problem let's assume that we are using an optical telescope with a thermal noise background from the CCD (dark current, particularly an issue with older CCDs and/or infra-red CCDs). After flat-fielding, the background appears as a zero-mean Gaussian with constant width over the image.

1.3.1 Part 1

You have an alert from LIGO that is also seen in with the X-ray/ultra-violet satellite SWIFT. SWIFT gives you a very precise location, so you take an image of that part of the sky. But because of SWIFT's accuracy, you know which pixel in your image to look for a counterpart in.

A) From looking at all the other pixels in your image, you can measure the width of the background Gaussian distribution X (pick something). Assuming you see a signal of strength Y (pick a floating-point number; optical CCDs are not sensitive enough to count photons so the readings are floating point brightnesses, not integer photons). Calculate the significance of your detection. Can you claim a discovery (traditionally 5-sigma or more)? The width of a Gaussian distribution is equal approximately equal to 2.355σ. So, with a width of 4, our background distribution has a standard deviation of 1.69851.

[4]:
```
signal = 10.34 #signal strength
background = 1.69851 #background std

probability = 1 - stats.norm.cdf(signal, loc=0, scale=background)
sigma = stats.norm.ppf(1 - probability)

sigma
```

[4]: 6.087688611046708

Fortunately, a signal with a strength of 10.34 is enough to claim a discovery. The signal turned out to have a 6.0876 sigma, surpassing the tradiational 5-sigma barrier.

1.3.2 Part 2

You have an alert from LIGO, but no associated detection from SWIFT. This could be because it is a black hole-black hole merger, a black hole-neutron star merger (neither seem to emit X-rays or UV light), or it could because SWIFT was indisposed at the time (wrong side of the earth in its orbit). Whatever the cause, you know what region of the sky to look in, but not which pixel.

B) If you have to look for a signal in 10k pixels, what is the probability distribution of your background? (Clearly state the statistical question, then turn that into math using your background distribution from part 1 of this problem.) **Question:** Given that we know the background of a single pixel is gaussian with a width of 4, what is the probability distribution of 10k pixels? We can find this distribtion by convolving the background distribution of a single pixel ten-thousand times.

C) Taking your brightest candidate signal from the region (assume it has the same signal as in part 1), calculate the significance of your detection. We can approximate the probability of the signal's significance by multiplying the p-value from version one by 10000.

[219]:
```
p_value = probability * 10000
sigma = stats.norm.ppf(1 - p_value)
sigma
```

[219]: 4.387706167517519

The significance of this detection is only 4.3877 sigma and thus doesn't quite reach the 5-sigma threshold we need to claim discovery.

1.4 Problem 4

The statistical issue we were exploring in the previous problem is called a trials factor (sometimes known as a look-elsewhere effect). This is an important effect, as if you search through a million locations, you would expect to see ~1 one in a million event. However, it is also often over estimated how big an impact this makes on the sensitivity of a search. So in this part of the lab we are going to invert the problem.

Let us again assume we have a Gaussian background (same parameters as Problem 3).

A) Calculate the signal required for a 5-sigma detection in Version 1

[230]:
```
probability = stats.norm.cdf(5)
signal_strength = stats.norm.ppf(probability, loc=0, scale=background)
signal_strength
```

[230]: 8.492549999949343

The signal strength required for a 5-sigma detection is 8.49.

B) Calculate the signal required for a 5-sigma detection in Version 2

[233]:
```
probability = (1 - (1 - stats.norm.cdf(5))/10000)
signal_strength_2 = stats.norm.ppf(probability, loc=0, scale=background)

signal_strength_2
```

[233]:
```
11.126153426860345
```

The signal strength for a 5-sigma detection for version 2 is 11.126.

C) Discuss how much brighter the signal must be for discoverey if you have a trials factor of 10k. Looking at your probability distributions, explain why the sensitivity penalty due to a trials factor is so low. The signal strength for discovery of trails factor of 10k is about 1.31 times greater than that of the single pixel (an increase of about 25%). The probability of achieving 5-sigma is already quite low (5.727e-10 away from 100% for a single pixel). Since we are so close to 100%, getting 10,000 times closer effects the value only a little, so the penalty is low.

D) If you changed the trials factor significantly (orders of magnitude), how large is the effect on your 5-sigma sensitivity threshold?

[6]:
```
probability = (1 - (1 - stats.norm.cdf(5))/1000000000)
signal_strength_2 = stats.norm.ppf(probability, loc=0, scale=background)

signal_strength_2
```

[6]:
```
13.718136616225056
```

After increasing the trials factor by 5 orders of magnitude greater than version 2 (one billion trials, holy moley), there is hardly any increase (only a about 20% greater than version 2). Thus, increasing the trial factor doesn't have much of an effect on the required signal.