# EP219: Data Analysis and Interpretation

Report: Assignment 4
Team Darth Analysis

October 29 2018 to November 4, 2018

# Contents

# 1 Problem Statement

Consider a dark matter direct detection exper- iment that is designed to measure the recoil energy of nuclei being scattered by dark matter particles. The measured recoil energies (ER) range from 0  40 KeV and the total number of events are reported in 1 KeV bins. The published data is attached in the text file "recoilenergydata EP219.csv" showing the number of events as a function of recoil energy (For the first bin 0.5 KeV is the central value of the bin, so the first bin corresponds to recoil energies between 0-1 KeV). We want to analyze this data to look for a dark matter signal! Unfortunately, there are a large number of background processes that could also contribute to dark matter scattering.

The dark matter signal spectrum has the following triangular form as a function of the recoil energy of the nucleus (ER).

$$\frac{dN}{dEr} = \sigma * 20 * (E_R - 5KeV) \; for \; 5KeV < E_R \; < \; 15KeV$$

$$= \sigma * 20 * (-E_R + 25KeV) \; for \; 15KeV < E_R \; < \; 25KeV$$

$$= 0 \; otherwise$$

Here the signal strength depends on a single parameter $\sigma$ which is the dark matter- nucleus scattering cross-section measured in femto-barns (fb) (1 fb = $10^{-39}$ $cm^2$).

The background rate is exponentially falling with energy and has the form,

$$\frac{dN}{dEr} = 1000 * exp(-\frac{E_R}{10KeV})$$

1. Make a clearly labelled histogram of the data.
2. Assuming background only processes, calculate the mean number of events that you would expect to see in each bin. Make a histogram of this expected background.
3. Assuming cross-sections of 0.01 fb, 0.1 fb, 1 fb, 10 fb, 100 fb, calculate the mean number of events that you would expect to see in each bin assuming background and signal. Make histograms for each of these cases. In which cases do you expect to tell by eye whether or not you have a dark matter signal?

4. Find the log likelihood function of the cross-section log L() and plot it. De- scribe in detail the process used to arrive at this log likelihood function.

5. Use this log likelihood function to find the maximum likelihood estimate (MLE) of the cross-section. Also report a 1-$\sigma$ interval of cross-sections that are consistent with the data.

## 2 Python Code

Here's our python code for to extract data into an numpy array and then changing the values of the columns of the array to plot the required histograms.

```python
import numpy as np
import pandas as pd
from scipy.integrate import quad
import math
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.interpolate import spline

def darkmatterfunction(x):
    if  x<5 or x>25:
        return 0

    elif x>15:
        return sigma*20*(25-x)
    else:
        return sigma*20*(x-5)

def backgroundrate(x):
    return 1000*(math.exp(-x/10))

def loglikelihood(x,d):
    return x*math.log(d)-math.log(math.factorial(math.floor(x)))

```

```python
data=1
backgroundsignalcalc=[0]*40
darkmattersignalcalc=[[0 for j in range(40)] for i in range(5)]



data = np.genfromtxt("recoilenergydata_EP219.csv", delimiter=',', skip_header = 1, usecols = (0,1))
dt=np.transpose(data)


err=0
for i in range(40):
    backgroundsignalcalc[i],err=quad(backgroundrate, data[i][0]-.5,data[i][0]+.5)

    for j in range(5):
        sigma=10**(j-2)
        darkmattersignalcalc[j][i],err=quad(darkmatterfunction,data[i][0]-.5,data[i][0]+.5)
        darkmattersignalcalc[j][i]+=backgroundsignalcalc[i]
        j+=1

plt.bar(dt[0],dt[1])
plt.title('Measured signal')
plt.xlabel('Energy in kEV')
plt.ylabel('Number of events')
plt.show()


plt.bar(dt[0],backgroundsignalcalc)
plt.title('Calculated Background signal')
plt.xlabel('Recoil energy in kEV')
plt.ylabel('Number of events')
plt.savefig('BackgroundSignal.png')
plt.show()
```

```


for j in range(5):
    sigma=10**(j-2)
    print("Total signal for sigma=", sigma)
    plt.bar(dt[0],darkmattersignalcalc[j])
    plt.title('Calculated signal')
    plt.xlabel('Energy in kEV')
    plt.ylabel('Number of events')
    plt.savefig('Signal for Sigma.png')
    plt.show()







#
#We call the array containing predicted values of N for sigma=0.01 as N_1 (which has 40 elements) and so on
#N is the array obtained from csv file for no. of events (Not really)

j=1
N=np.transpose(darkmattersignalcalc)
L=[0]*5
m=M=[[0.0]*5]*40
L[j] = M[0][j]

for j in range(0,5):
    for i in range(0,40):
        m[i][j]=(N[i][j] + abs(N[i][j]- data[i][1]))
        M[i][j]=data[i][1]*math.log(m[i][j])-math.log(data[i][1])-m[i][j]    #Calculated in log itself since product is too big
        L[j] = L[j]+M[i][j]               #LOG Likelihood
```

```

s=[-2,-1,0,1,2]
plt.plot(s,L)
plt.title('Maximum likelihood Estimate')
plt.xlabel('Log Sigma')
plt.ylabel('Log likelihood')
plt.savefig('LogLikelihood.png')
plt.show()

s1=np.array([-2,-1,0,1,2])
xnew=np.linspace(s1.min(),s1.max(),300) #interpolating the given data into a smooth curve to find the 1-sigma interval
L_new=spline(s,L,xnew)
q=0
j=0
for i in range(len(L_new)):
    j=abs(L_new[i]-(L[0]/math.sqrt(math.e)))  #since even after interpolation, the values of the likelihood don't exactly drop by sqrt(e), we found the only value present in the interpolated set which has an error of 54 points of L_max/sqrt(e).
    if j<54.75:
        q=xnew[i]
        print(q)#the value of the order of sigma(parameter) for the 1-sigma interval
        print("The Sigma interval Value")
    else:
        continue

plt.plot(xnew,L_new)
z=plt.axvline(q)
g=plt.axvline(-2)
plt.xlabel("Log sigma")
plt.ylabel("Value of Log Likelihood")
plt.show()
```

Finally we used the interpolate function from Scipy and interpolated the data points to fit a smooth curve through them for the sigma interval.

The code for part 2 is as given below. It is to be considered as an extention of the above code.

```
#Problem 2 begins here
#Log Likelihood mentioned below as defined in assigment 4
# def loglikelihood(x,d):
  #  return x*math.log(d)-math.log(math.factorial(math.floor(x)))
#backgroundsignalcalc has the data for different bins
#s=np.random.poisson(5,100000)
print(s)
#count, bins, ignored = plt.hist(s, 50, density=True)
#plt.show()
ts=[0]*1000
for i in range(1000):
    for j in range(40):
        z=0
        z=z+loglikelihood(backgroundsignalcalc[j],np.random.poisson(backgroundsignalcalc[j],1))
        #np.random.poisson is used to create a random dataset with poisson distribution around the background data.
    ts[i]=z # 1000 different datasets are used
tsdata=0
for z in range(40):
    tsdata=tsdata+loglikelihood(backgroundsignalcalc[j],dt[1][z]) #test statistic of the given data

a=sns.distplot(ts, hist=True, kde=False,
                bins=40, color = 'blue',
                hist_kws={'edgecolor':'black'})     # plotting test statistic against frequency of it's
```

```
occurence


heights=[0]*40      # array to store the frequency
g=min(ts)
for i in range(40):             #loop to calculate the frequency corresponding to different test statistic bins
    for j in range(1000):
        if (g < ts[j])&(ts[j] < g+0.76):
            heights[i]+=1
    g=g+0.76
Hits950=0
l=0
while Hits950 < 950:
                            #since 1000 datasets are used, the test statistic
                            #corresponding to 950th dataset (in increasing order of the test statistic)
  Hits950+=heights[l]
  l+=1
print(l*0.76+min(ts))
plt.axvline(l*0.76+min(ts))
plt.show()
```

# 3 Histograms

## 3.1 Problem 1

1. **Question a**:- Histogram Plot for the measured data

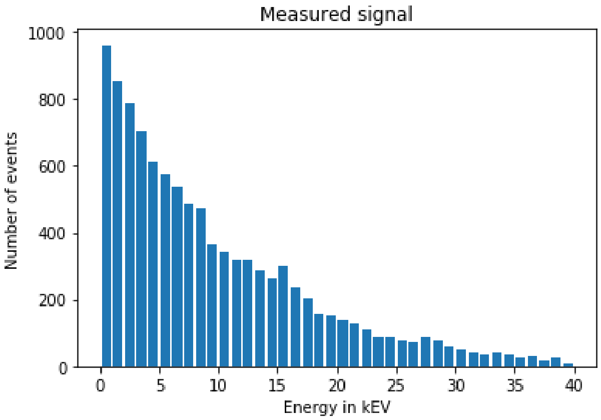

Figure 1: Measured Signal

2. **Question b**:- Histogram Plot for the background data

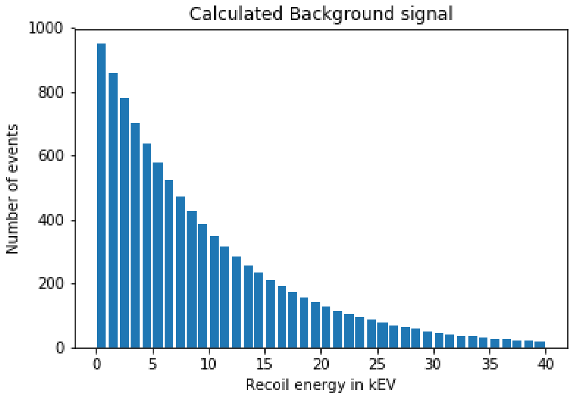

Figure 2: Background Signal

3. **Question C.1**:-Histogram Plot for $\sigma = .01$

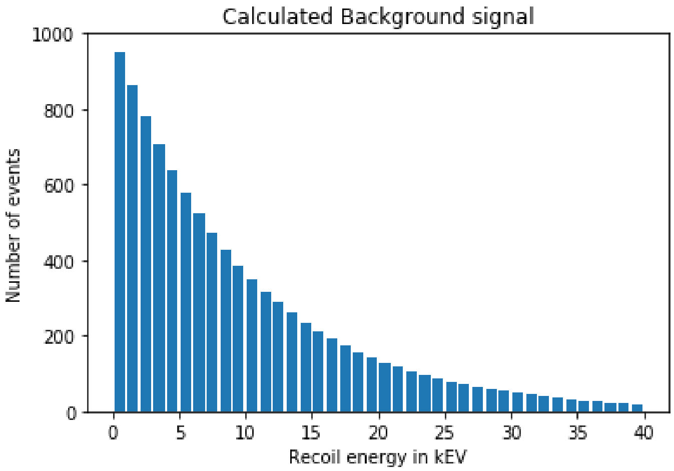

Figure 3: Calculated Signal for $\sigma$=0.01

4. **Question C.2**:-Histogram Plot for $\sigma \ = \ .1$

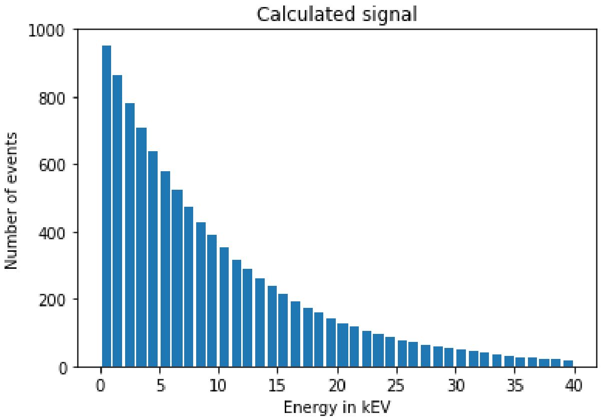

Figure 4: Calculated Signal for $\sigma$=0.1

5. **Question C.3**:-Histogram Plot for $\sigma \ = \ 1$

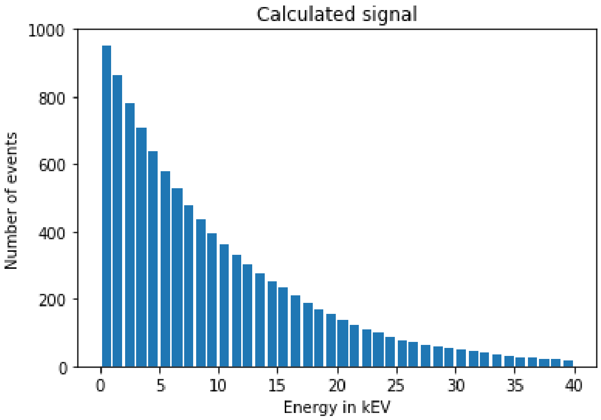

Figure 5: Calculated Signal for $\sigma$=1

6. **Question C.4**:-Histogram Plot for $\sigma\ =\ 10$

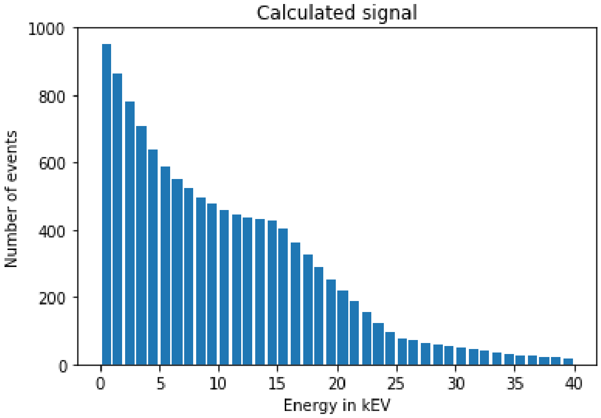

Figure 6: Calculated Signal for $\sigma$=10

7. **Question C.5**:-Histogram Plot for $\sigma\ =\ 100$

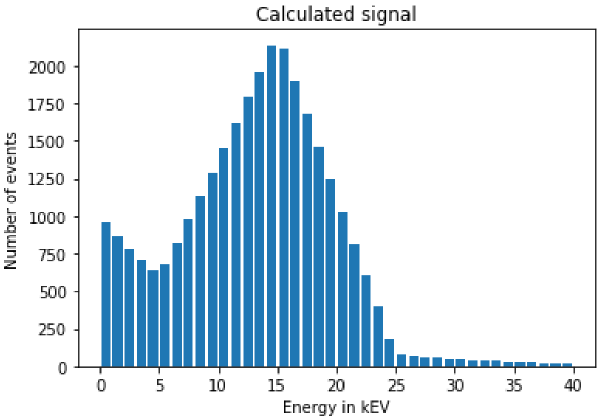

Figure 7: Calculated Signal for $\sigma$=100

We can clearly see the dark matter signal for $\sigma = 1, 10, 100$.

## 3.2 Problem 2

1. **Question 2.a**:-Histogram Plot of test statistic for the null hypothesis:

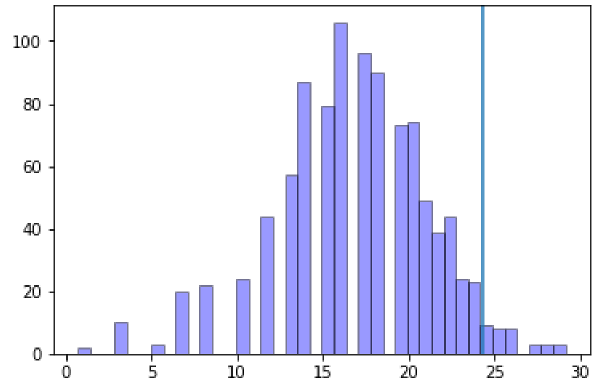

Figure 8: Test statistic for Null hypothesis

2. **Question 2.b**:-Histogram Plot of test statistic to display the threshold for 95% confidence compared against the given data.

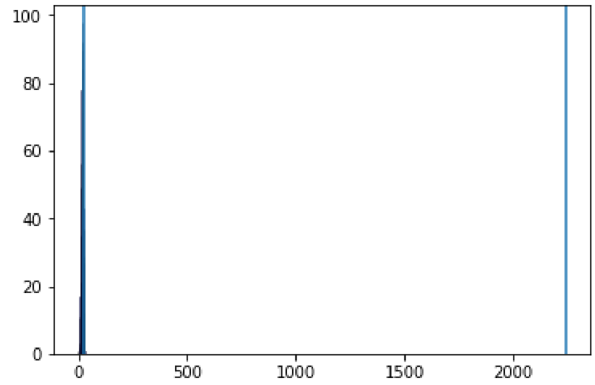

Figure 9: Null Hypothesis Test Statistic plotted with the given data

The vertical blue line in Fig: 8 corresponds to the 95% confidence level which was calculated to be around 24.74. As can be observed from Fig: 9, the given data is well beyond the 95% confidence threshold.

# 4 Log Likelihood

## 4.1 Log Likelihood Function

We claim that the Log Likelihood is of the following form:-

$$log\ \mathcal{L}\ =\ \sum_{n=1}^{40} m_i(log(B_i\ +\ dm_i))\ - B_i\ - dm_i\ - log(m_i!)$$

where

1. $m_i$ is the value of the measured number of events at the ith bin.
2. $B_i$ is the value of the number of events due to background distribution(calculated from the given distribution of Background Signal)
3. $dm_i$ is the value of the number of events due to dark matter signal.
   It depends on the parameter $\sigma$.

This corresponds to the $\mathcal{L}$ being a product of possion distribution probabilities with a mean at each bin being $B_i\ +\ dm_i$ and $m_i$ being the observed the measured number of final signal at each bin.

**Reason for assuming a possionian functional form:-**
Here the number of bins if fairly large(nearly 40), so for each individual event, the probability of falling into a particular bin,$p_i$, is very small. This correspsonds to saying that for each nuclei, the probability of having an energy between $E_R$ and $E_R + 1$ (both in KeV) is very low. Also, the total number of particles is very large. Therefore, under small Bernoulli probability of the particle falling in a particular bin and large number of particles, we can approximate the distribution of particles in ith bin as Poissionian distribution around a mean of $B_i\ +\ dm_i(\sigma)$.
Intrinsically we have used the fact that a binomial distribution $B_p^N$ under large N and small p, with Np = constant, tends to a Poissionian distribtuion.

# 5 Maximum Likelihood Estimation

From the histogram plots, we see that $\sigma\ = 1,\ 10,\ 100$ serve as a very poor estimation for the parameter. Also we see that $\sigma\ =\ 0,01\ and\ 0.1$ are nearly

identical in terms of approximating the measured signal. However, they fail to account for the abrupt changes in the measured signal completely and more or less resemble the background signal. Hence, among the given orders of parameters,we infer that $\sigma$ is definitely not of the order of 0, 1 or 2.
Report of the findings from the data:-

1. MLE:-
   The value of $\sigma$ was **0.01.**
2. The 1-$\sigma$ interval was [0.01,$10^{0.9164}$]. The interval extends on the RHS of the MLE.

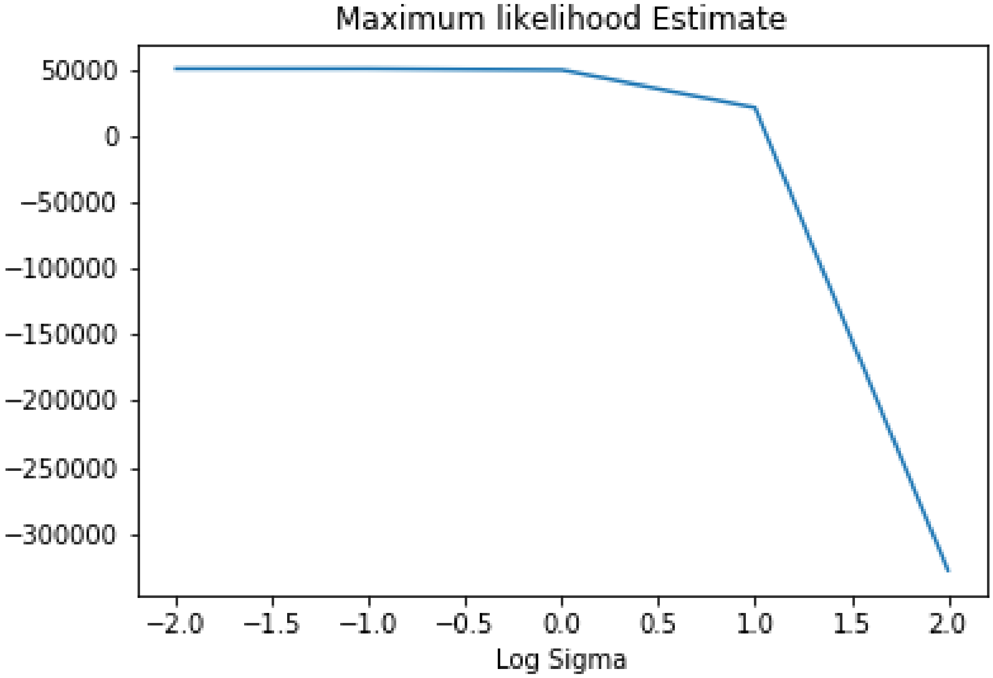

Figure 10: Log Likelihood

After interpolation using a smooth curve, the graph looked like this:-

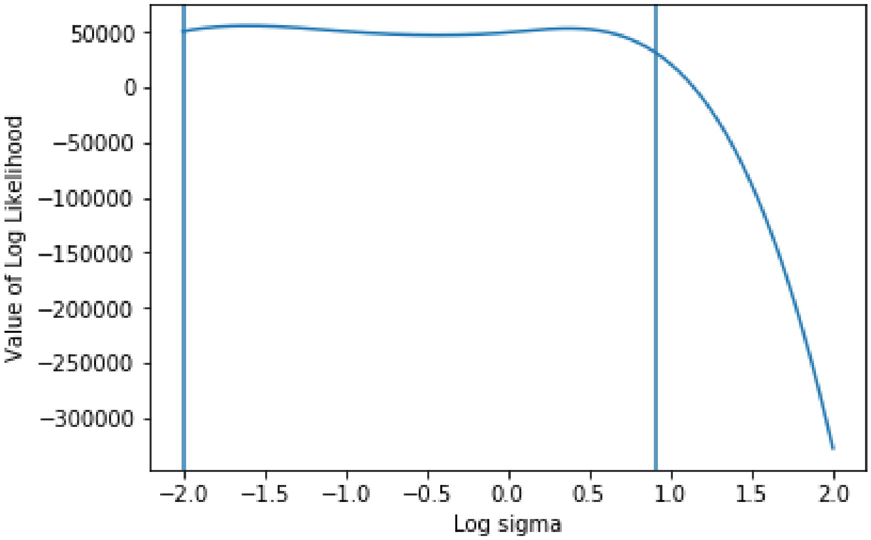

Figure 11: Interpolated Graph of Likelihood

The vertical line represents the 1-sigma interval from -2.0 to 0.9164.

# 6 Null Hypothesis Testing

Poisson distribution is assumed in coherence with the aforementioned reasons. A random number generator is used to generate data by converting the uniformly distributed random numbers to the required Poisson distribution by virtue of the predefined numpy function 'random.poisson'. The background signal is passed as the poisson parameters accordingly. Log likelihood the the Poisson given null hypothesis is used as the test statistic. As calculated an inferred from Fig: 8 and Fig: 9, the 95% confidence threshold lies around 24.74 for the test statistic, which is well below the observed 2238.65 corresponding to the given data. This suggests that the null hypothesis can be safely ruled out as a potential hypothesis describing reality, i.e. Dark Matter exists (with negligent uncertainty).

## 7 Team Responsibilties

- Project Leader - Aakash Marthandan
- Programmer - Raunak Dutta
- Web Manager - Ayush Bhardwaj
- Report Writer - Guru Kalyan Jayasingh

## 8 Website and Resources

This the link to our website.
Also, we have uploaded our code to Github repository. The link for the same can be found here.