ISRO's Web-Based Automatic Identification of Solar Bursts in X-RAY Light Curves Report

March 2022

1 Introduction

In this problem statement we have tried to make a model to detect Solar X-Ray bursts. The data was recorded by the Chadrayaan 2 orbiter which is in a polar orbit around the moon. In this project, we have detected the X-Ray bursts and classified them using a ML model. We have used the calibrated files among the 713 files of data which were available to us. Each file has data spread over one day.

2 Methodology

2.1 Reading and Plotting the data

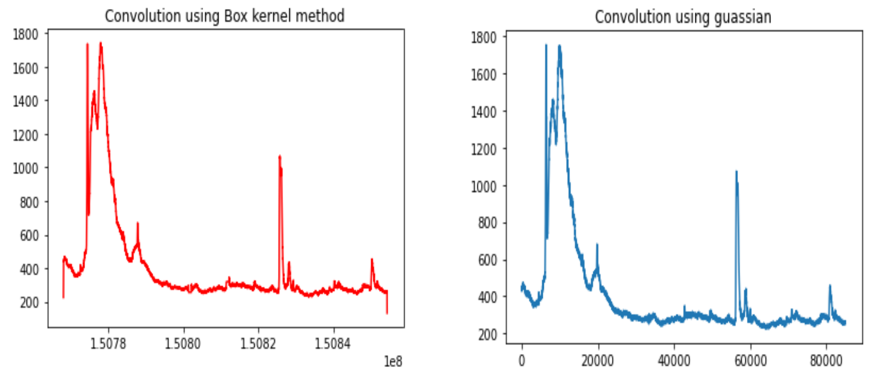
The data recorded by the Chandrayaan 2 X-Ray Solar monitor has a light curve with time on the X-axis and count rate on the Y-axis. This curve was created by combining the lc(light curve) files and the gti(good time interval) files. We used the fits.open function from astropy to open the two files and then combined them to get the light curve. We got an array with 4 fields(TIME, RATE, ERROR and FRACEXP). The time field contains information about time of observations. The rate field contains information about the flux in units of counts per second. We have plotted these two fields after considering the error field.

2.2 Smoothing the curve

The first thing we noticed once we plotted the graph was the high level of noise in the data. We could visually see a short 'burst' in almost all of the noise, but even the rise and the decay plot lines were filled with noise. So, our first task was to smooth out the noisy data curve. We got the idea for smoothing the curve from Ref.[1]. We had used the boxcar method, where we consider a box on n number of points and replace all the points in that box by the mean of the data points in that box. On doing the boxcar method we realised there were many flat lines in the smoothed curve. Hence, we decided to go with the Gaussian smoothing method. Here we use a Gaussian function whose equation would be in the form:

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right). \tag{1}$$

This function gives a higher weightage to the central points and lesser weightage to the points near the ends. Therefore we see better result as it will eliminate the flat lines as seen in the figure below.



(a)

(b)

Figure 1: (a) The convoluted graph on using Box kernel method. (b) The convoluted graph after using Gaussian Method.

2.3 Peak Detection

We detected the peaks by filtering all miscellaneous peaks by the threshold value. The threshold value is determined by calculating the midpoint in the prominence ranges. After detecting the relevant peaks, we use the position of previous and next peaks. We have imported electrocardiogram from miscellaneous routes of scipy to load light curves as 1D signal. Further, we have imported find-peaks and peak-prominences from scipy.signal. It helps us to find local maxima by analysing the neighboring values. We have imported the peak prominences module to calculate how much of the peak out stands the other values. The peaks are selected on the basis of a threshold value of 0.5 times the sum of minimum and maximum local peaks. This way we are able to detect the correct number and values of peaks.

2.4 Feature Extraction

In this Model, we have focused on 4 features, namely, Duration of burst, time of rise, time of decay and peak flux. To extract these features we have used a simple algorithm. To detect Peak flux and time of occurrence of peak, we used the method described in the previous section. To find the peak duration we imported the module named peak-widths from scipy.signal. This function calculates the width of the peak in the light curve at a relative distance to the peak's height and prominence. The output of this function contains 4 values, namely, peak duration/width, the start time, the end time and the height of the start/end point from the zero line. To find the rise time and decay time, we subtract the start and end points from the peak time.

2.5 The ML Model

After extracting the features, we create a data file for the ML model. In the file, each row represents data for a particular peak. The 4 columns represent the peak flux, rise time, decay time and duration of bursts. The features for each observation are then used to cluster the data using the k-means clustering algorithm. The data is firstly normalized and the elbow curve method is used to find the optimum number of clusters. On running the model on the data we got the elbow at around 3-4 clusters. TO get more accurate model, we took logarithmic of the peaks column.

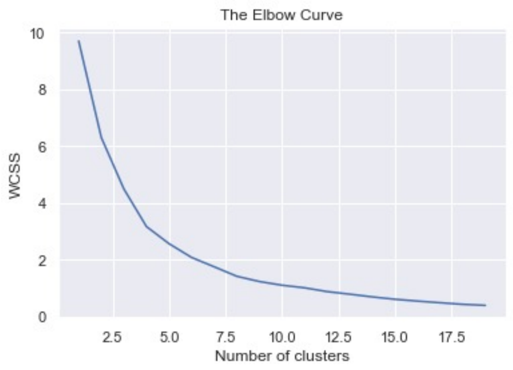


Figure 2: The model gives the same start and time of two peaks if they are close to each other such that the minimum count between them lies above the threshold value.

3 Limitations

3.1 Feature Extraction

One of the most challenging aspects of the model is extracting the features from the given data. Because X-Ray bursts show large variations in magnitude, are differently affected by noise and do not follow the same behavioural trends, it becomes difficult to apply a standard method (as described in the methodology) to find peaks across the data set. Among the limitations in our model, we cover the most prominent ones below.

Firstly, detecting peaks that arise close to the background noise level is not possible in our model when peaks with a count magnitude much greater then the background activity is also present in the same data file. This is because all the peaks with a rate close to the background end up below the 75th percentile in files which have peaks with a rate comparatively much larger than the background.

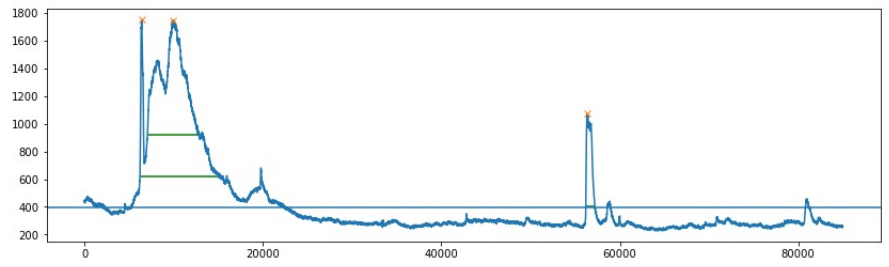


Figure 3: The model gives the same start and time of two peaks if they are close to each other such that the minimum count between them lies above the threshold value.

Second, the model gives the same start and end times of the two peaks if the minima between them lies above the set threshold. This can be seen in Fig.2 where the model gives the same start and end times for the first two peaks.

Third, the model is not able to process multiple files at once. This makes the data reading computationally slow. Also, the model is not able to process raw data files for now but only calibrated ones.

3.2 Future Scope

The first thing that can improved in the future version of this model is the curve fitting on the given data points to obtain a smooth curve. The Gaussian convoluted signal definitely makes the data processing simpler but there is scope for a lot of improvement. Next, we plan on implementing multiple-data thresholding procedures for processing all peaks (close or far from background) present in a data file. The idea is to detect peaks with count rates farthest from the background with a certain threshold, then to eliminate those peak points and set a new threshold value for peaks with rates close to the background. Lastly, we aim to connect all the time series data to reduce computational time in processing each day's data. We can then implement a multiple-thresholding technique, as just discussed, to find the peaks for the entire data set.

References

[1] Markus J Aschwanden and Samuel L Freeland. Automated solar flare statistics in soft X-rays over 37 years of GOES observations: The invariance of self-organized criticality during three solar cycles. *The Astrophysical Journal*, 754(2):112, 2012.