**Project** © Spring 2019

# Solar Events Prediction

Advisor:Dr. Daniel Pimentel-Alarcón Project By: Naga Jagadeesh Mutala & Ratanpriya Shrivastava

## 1.1 Introduction

The Sun produces several solar events which cause adverse effects on Earth's atmosphere such as disruptions in communication infrastructure, power grid failures and damage of satellites. The event prediction is a challenging task because there is no method available to accurately predict onset and duration of solar events. Therefore, it is important to gain scientific knowledge on the occurrence of solar events and different parameters that are responsible for such events; so it will help in understanding and predicting the solar events. Out of several events occurring, mainly the Active Region, Sigmoid, Coronal Hole, Flare have strong influence on the local space weather in the vicinity of the earth.

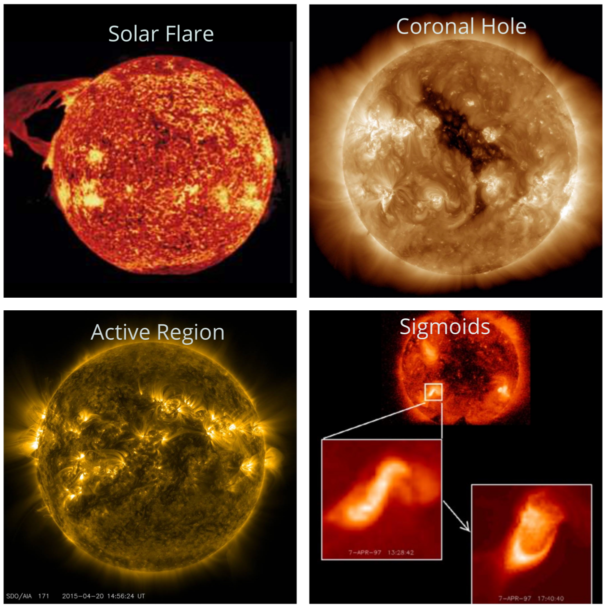

Figure 1.1: Powerful solar events occurring in the Sun

## 1.2 Goal

To implement Machine Learning Algorithms and identify the features which are responsible for the occurrence of solar events. With the help of these features we are predicting the occurrence of future solar events.

## 1.3 Review of Logistic Regression

Logistic regression is one of the most commonly used statistical techniques. It is applied on the data in which there is a binary (success-failure) outcome, or where the outcome takes the form of a binomial proportion. Like linear regression, one estimates the relationship between predictor variables and an outcome variable. In logistic regression, however, one estimates the probability that the outcome variable assumes a certain value, rather than estimating the value itself.

$$\mathrm{y} \in \{0,1\}$$

In the above equation, y is a categorical variable, which means that output can take only two values; "0" or "1", that represents outcomes such as True/False, Win/Lose, or Dead/Alive. If random variable y has the following equation:

$$\mathbb{P}(\mathrm{y}=1)=\mathrm{p}$$

$$\mathbb{P}(\mathrm{y}=0)=1-\mathrm{p}$$

$$\mathbb{P}(\mathrm{y},\mathrm{k})=\begin{cases}\text{p} \quad \mathit{if} \ \ \mathrm{k}=1,\\ \text{1-p} \quad \mathit{if} \ \ \mathrm{k}=0\end{cases}$$

For $0 < \mathrm{p} < 1$, y is called *Bernoulli* random variable and it is expressed as

$$\mathrm{y} \sim \mathrm{Bernoulli(p)}$$

Probability is used to compare two events. Assume p is the event of a success and $1-\mathrm{p}$ is the event of a failure. Then take probability of success and divide it by probability of a failure to find its **odds**

$$odds := \frac{\mathrm{p}}{1-\mathrm{p}}$$

**Odds** by definition, is the extent to which an event is likely to occur. It is measured by the real number of the favorable cases possible. So in the above equation, p must be between 0 and 1.

Now, taking the Logarithm of odds

$$log(\frac{\mathrm{p}}{1-\mathrm{p}})$$

Then formally, the model of logistic regression can be represented as:

$$log(\frac{\mathrm{p}}{1-\mathrm{p}})=\beta_1\mathrm{x}_1+\beta_2\mathrm{x}_2+.......+\beta_\mathrm{d}\mathrm{x}_\mathrm{d}$$

We have taken Log of the odds ratio. The Log of odds is given by the linear combination of the independent variable and the parameters. In a Logistic Model, the parameter of the model beta and x are crucial in determining the odds of success. The larger the number is, the higher the probability of success will be.

$\mathbb{P}$ represents the probability of 1 and $e$ represents the base of the natural logarithm. Finally, $\boldsymbol{\beta^T}$ and feature vector $x$ are the parameters of the model.

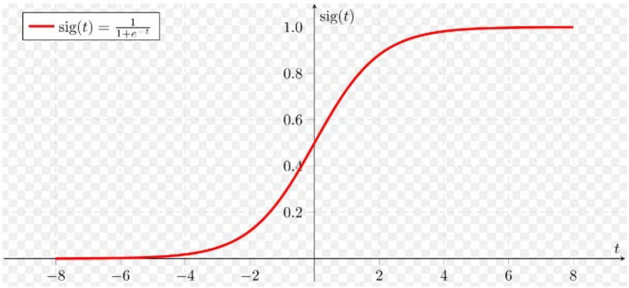

Figure 1.2: Figure describes the graph of the Probability function

The prediction of event is based on the following conditions:-
y = 1, if

$$(\frac{1}{1+e^{(-\boldsymbol{\beta}^T X)}}) > (1-\frac{1}{1+e^{(-\boldsymbol{\beta}^T X)}})$$

And, y=0 if

$$(\frac{1}{1+e^{(-\boldsymbol{\beta}^T X)}}) < (1-\frac{1}{1+e^{(-\boldsymbol{\beta}^T X)}})$$

**Maximum Likelihood**

$$L(\boldsymbol{\beta}|\mathbf{Y},\mathbf{X}) = \sum_{i=1}^{N} y_i log(\frac{1}{1+e^{(-\boldsymbol{\beta}^T x_i)}}) + (1-y_i)log((1-(\frac{1}{1+e^{(-\boldsymbol{\beta}^T x_i)}}))$$

We are applying Logistic Regression on several time frames. We got the $\boldsymbol{\beta}$ values for each time frame and calculated the mean of all $\boldsymbol{\beta}$ values and compared the accuracy of the model. In few instances, we got zeros for all coefficients of $\boldsymbol{\beta}$. So instead of ignoring those values we want to calculate l2-norm and check if l2-norm value is greater than zero. If l2-norm value is non-zero, it can impact the overall accuracy of the model.

**LASSO** (Least Absolute Shrinkage and Selection Operator) is a powerful method that performs two main tasks: Regularization and Feature Selection. The LASSO method puts a constraint on the sum of the absolute values of the model parameters, the sum has to be less than a fixed value (upper bound). In order to do so, this method applies a shrinking (regularization) process where it penalizes the coefficients of the regression variables, shrinking some of them to zero. During feature selection process the variables that still have a non-zero coefficient after the shrinking process are selected to be part of the model. The goal of this process is to minimize the prediction error.

In practice the tuning parameter $\lambda$, that controls the strength of the penalty, has a great importance. Indeed when $\lambda$ is sufficiently large then coefficients are forced to be exactly equal to zero, this way the dimensionality can be reduced. The larger the parameter $\lambda$ is, the more number of coefficients are shrinked to zero. On the other hand if $\lambda = 0$ we have an OLS (Ordinary Least Square) regression.

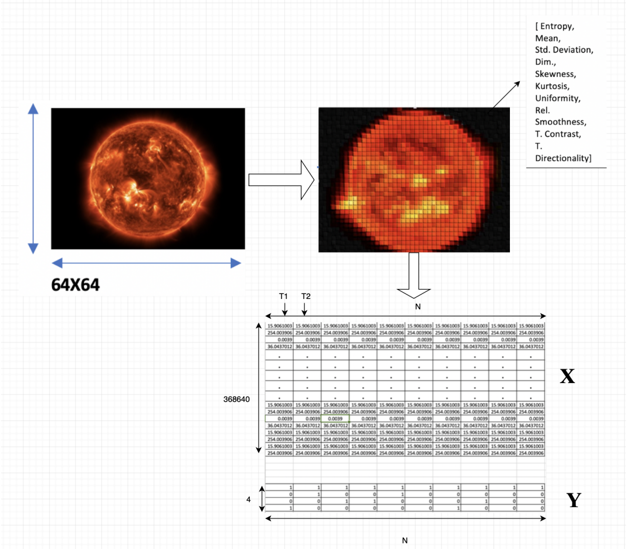

Figure 1.3: Figure describes the implementation of Logistic Regression

There are many advantages in using LASSO method. First of all it can provide a very good prediction/accuracy, because shrinking and removing the coefficients can reduce variance without a substantial increase of the bias, this is especially useful when you have a small number of observation and a large number of features. In terms of the tuning parameter $\lambda$ ,we know that bias increases and variance decreases when $\lambda$ increases, indeed a trade-off between bias and variance has to be found.
Moreover the LASSO helps to increase the model interpretability by eliminating irrelevant variables that are not associated with the response variable, this way also overfitting is reduced.

The cost function for Log LASSO (Least Absolute Shrinkage and Selection Operator) regression can be written as

$$L(\boldsymbol{\beta}|\mathbf{Y},\mathbf{X}) = \sum_{i=1}^{N} y_i log(\frac{1}{1+e^{(-\boldsymbol{\beta}^T x_i)}}) + (1-y_i)log((1-(\frac{1}{1+e^{(-\boldsymbol{\beta}^T x_i)}})) + \lambda\sum_{i=1}^{N}|\beta_j|$$

## 1.4 Dataset

In order to predict Solar events, we have used Solar Images taken between January 1, 2012 and December 31, 2014. These images are stored in DM Lab repository which can be accessed using various APIs. Using APIs, we access the image data at 9 different Wavelengths [94,131,171,193,211,304,335,1600,1700]. These APIs provide us image parameters [ Entropy, Mean, Std. Deviation, Dim., Skewness, Kurtosis, Uniformity, Rel. Smoothness, T. Contrast,T. Directionality] and event occurrence data which we are storing in **X** and **Y** Matrices. We have used Logistic regression and Logistic LASSO to predict the event occurrences.

Image Parameter XML [GET]:
http://dmlab.cs.gsu.edu/dmlabapi/params/SDO/ AIA/64/full/?wave=171&starttime=2012-02-13T20:10:00

TEMPORAL EVENT SEARCH QUERY [GET]:
http://isd.dmlab.cs.gsu.edu/api/query/temporal?starttime,endtime, tablenames,sortby,limit,offset,predicate

## 1.5 Related Work

Most of the current methods for solar events prediction are data-driven approaches rather than purely theoretical modeling. Data-driven approaches are divided into two categories : Linear statistical [3],[5] and Non-Linear statistical (mostly machine learning)[4],[7],[8]. These two categories can be subdivided into two subcategories : Line-of-sight magnetogram-based [2] models and Vector magnetogram-based models.

Some methods by Piana, et al., 2017 [8] involved use of Deep learning technology for the prediction of solar flares from GOES data. The Units are integrated in this system to predict solar flares. The system takes the input which is GOES data, and generates the possibility of a solar flare. So, the system starts from encoded GOES X-ray flux 1 minute data time series data transformed to 64 by 64 MTF images. After that, the MTF images are fed to the Deep Learning Convolutional Neural Network to predict a solar flare.

Learning from magnetograms proposed by Huang, et al.,2017 [2] is another notable method used for solar event forecasting. A deep learning model is implemented to automatically learn the solar flare forecasting features from the magnetograms of active regions. The architecture of the convolutional neural network consists of the convolutional layer, pooling layer, normalization layer, and fully-connected layer. It is trained using stochastic gradient descent.

Another approach is by integrating Support Vector Machine (SVM) with Case-Based Genetic algorithm (GA) to improve the precision of the space weather forecast cited by Yamamoto, et al., 2016 [4]. It provides an effective way for the imbalanced forecasting by assigning different cost parameters to the two classes offered by SVM. The SVM model was trained and tested, and its performance was estimated using forecast verification metrics called True Skill Statistic (TSS) and integrated into the GA as the evaluation step. The fitness evaluation is performed by executing the SVM using the feature (pattern file) combination specified by the gene and cross-validation with using half of the population for building the model and the other half for its validation.

A multivariate time series nearest neighbor search classifier applied on the solar flare data and ranked the importance for different solar magnetic parameters. In MVNN approach by Boubrahimi, Angryk, 2018 [5], Taniguchi et al.[7], parameter weights have to be calculated. To do that, a univariate KNN search is applied on every parameter on a given domain of k. Correlation between parameter and class labels have been recorded and the corresponding weight vector is obtained. In the next step, the nearest neighbor search is performed to aggregate the votes for classification.

Apart from SVM, KNN researchers did analysis by means of the hybrid LASSO algorithm to find the occurrence of the flare. Few analysis by Hamdi, et al.[6] involved construction of a training set by randomly extracting 66 percent of Active Regions from the overall set of Active Regions and label the feature vectors associated with each extracted Active Regions by annotating whether a flare with an intensity sufficiently high would occur in the next 24 hours or not. Solar flares are classified according to their intensity i.e. the amount of energy they release: from low to high energy we have flares of class C, class M, and class X. Some researchers focused on predicting whether the least intensity C flare would occur in the next 24 hours.

Also, Distance Density clustering and multivariate time series decision tree are used for flare data prediction by Ma, et al., 2017[1]. Dynamic Time Warping(DTW) algorithm has been used over Euclidean distance for one-to-many mappings and measuring the similarity between two temporal sequences. Then Distance Density clustering is applied to the idea of cluster split on sparse regions based on a distance plot. Later univariate time series clustering and multivariate time series decision tree were applied to the flare data. The performance was evaluated using Accuracy, F-score and Rand index.

Some analysis by Suk, et al.,2016 [3] was done on determination of trigger mechanisms of the flares which significantly contribute to space weather prediction. Emphasis was mainly on the pre-flare phase where frequency analysis is made and principal component analysis (PCA) is applied using high-order moments. The analysis includes a search for and identification of the turning point at which the pre-flare phase changes to the flare itself, or the frequency analysis of solar plasma oscillations that affect the flares.

All of the above mentioned works focussed on the parameterization of the Active Regions by line-of-sight or vector magnetograms for flare detection. In this work, we evaluated the Logistic Regression based on timestamps to distinguish flaring and non-flaring Active Regions.

## 1.6 Challenges

Since $\mathbf{Y}$ is very sparse in a specified time frame, we got a sequence of either all 0's or all 1's. Therefore, Logistic Regression was not able to predict as it needs a combination of 0's and 1's. Because of this, we had to select a time frame (occurrence of an event), where we can find the combination of 0's and 1's so that Logistic Regression can be obtained to get better values of $\boldsymbol{\beta}$.

Out of several coefficients that we have got, selecting significant coefficients was a difficult task. So, we implemented Logistic LASSO for coefficient selection and regularization, and this improved our model's accuracy.

## 1.7 Ideas

We took the time frame as large as possible so that we get a combination of 0s and 1s and Logistic Regression can be applied. We choose only significant coefficients by using Log LASSO so that our accuracy can be improved.

## 1.8 Results

Following are the results which we have achieved:-

Logistic Avg. Accuracy: - 0.66514867
Log Lasso Avg. Accuracy: - 0.694925106

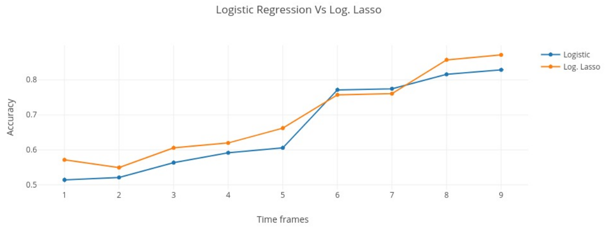

Figure 1.4: Figure describes the graph of the Probability function

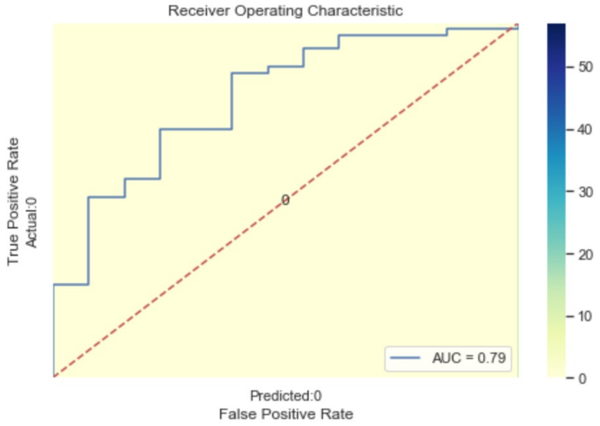

Figure 1.5: Figure describes the ROC curve for Logistic Regression

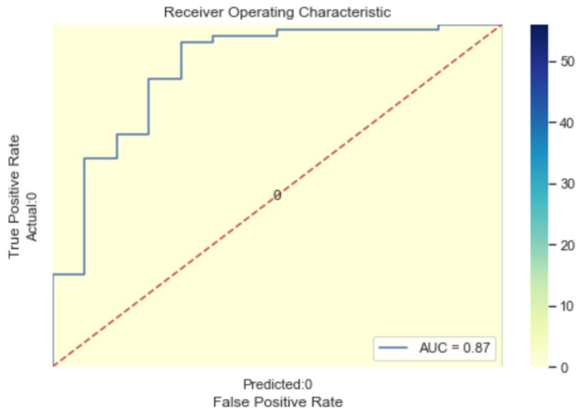

Figure 1.6: Figure describes the ROC curve for Log LASSO Regression

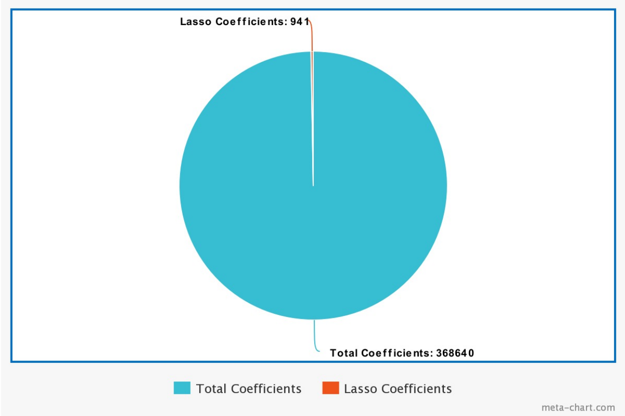

Figure 1.7: Figure describes reduction of the coefficients by applying Log LASSO

## 1.9 References

[1] Ruizhe Ma, Soukaina Filali Boubrahimi, Shah Muhammad Hamdi, Rafal A. Angryk, Solar flare prediction using multivariate time series decision trees, 2017.
[2] Xin Huang, Huaning Wang, Long Xu, Wenqing, Sun Learning solar flare forecasting model from magnetograms, 2017.
[3] Tomas Suk, Stanislava Simberova by Solar flare retrieval, detection and analysis, 2016.
[4] Yukiko Yamamoto, Daichi Itoh, Setsuo Tsuruta, Takayuki Muranushi, Yuko Hada-Muranushi, Syoji Kobashi, Yoshiyuki Mizuno, Rainer KnaufSolar, flare prediction by SVM integrated CBGA with dynamic mutation rate, 2016.
[5] Soukaina Filali Boubrahimi, Rafal Angryk, Multivariate Time Series Nearest Neighbor Search: A Case Study on Solar Flare Prediction, 2018.
[6] Shah Muhammad Hamdi, Dustin Kemptin, Soukaina Filali Boubrahimi, Rafal A. Angryk, A time series classification-based approach for solar flare prediction, 2017.
[7] Yoshio Taniguchi, Yoshihiko Kubota, Setsuo Tsuruta, Takayuki Muranushi, Yuko Hada-Muranushi, Yoshiyuki Mizuno, Syoji Kobashi, Yoshitaka Sakurai, Rainer Knauf, sAndrea Kutics, A SVM integrated Case Based Learning Data GA for Solar Flare Prediction, 2018.
[8] Tarek A M Hamad Nagem, Rami Qahwaji, Stan Ipson, Deep learning teachology for the prediction of solar flares from GOES data, 2017.
[9] Michele Piana, Federico Benvenuto, Anna Maria Massone, Cristina Campi, FLARECAST: An I4.0 Technology for Space Weather Using Satellite Data, 2018.
[10] Ddenis Ullmann, Slava Voloshynovskiy, Lucia Kleint, Sam Krucker, Martin Melchior, Cedric Huwyler, Brandon Panos, DCT-Tensor-Net for Solar Flares Detection on IRIS Data, 2018.
[11] Ahmet Kucuk, Juan M. Banda, Rafal A. Angryk, Data Descriptor:A large-scale solardynamics observatory imagedataset for computer visionapplications, 2017.