LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Technical Note	**LIGO-T1700198–v1**	2017/07/12

Online Detector Characterization using Neural Networks

Roxana Popescu

California Institute of Technology
LIGO Project, MS 18-34
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project, Room NW22-295
Cambridge, MA 02139
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

LIGO Hanford Observatory
Route 10, Mile Marker 2
Richland, WA 99352
Phone (509) 372-8106
Fax (509) 372-8137
E-mail: info@ligo.caltech.edu

LIGO Livingston Observatory
19100 LIGO Lane
Livingston, LA 70754
Phone (225) 686-3100
Fax (225) 686-7189
E-mail: info@ligo.caltech.edu

http://www.ligo.caltech.edu/

1 Introduction

The data obained from LIGO has noise that comes from many sources. In order to be able to better distinguish signals from the noise, it is important to characterize the type of noise observed. Machine learning algorthms can be used to look for patterns within the data and to classify the data into different categories.

There are many sensors at the LIGO detectors that measure sources of noise. For example, there are several stations at each LIGO detector that measure seismic noise in different frequency channels in each of the X,Y, and Z directions. Within the data, there are different types of seismic noise such as earthquakes and anthropogenic noise.

In order to sort data, machine learning algorithms can use one of two approaches: classification or clustering. Classification algorithms search the data and sort the data into already defined categories. Clustering algorithms look for relationships within the data to create categories into which the data is sorted. Classification algorithms are part of supervised learning since the computer determines the structure of the data from data that is already provided. Clustering algorithms are part of unsupervised learning since the computer determines the structure of the data without any previous information. Clustering algorithms can be used to characterize the noise by identifying common characteristics within the noise depending on its sources and can further help with classification. [1]

Neural networks can be used to find relationships between the inputed data by using hidden layers of connections within the data. Recurrent neural networks are neural networks that use loops within them so that previous information can be retained. [1]

2 Objectives

The aim of this project is to characterize different sources of noise from LIGO using machine learning algorithms. First I will test clustering algorithms on seismic data, and then implement a neural network to sort through the seismic noise data, as well as other noise data.

3 Clustering Algorithms

3.1 K-means clustering

The k-means clustering algorithm creates clusters by separating data points into k number of groups. The value of k is inputted into the algorithm. The clusters are determined by minimizing the inertia, or the within-cluster sum-of-squares. The inertia is a measure of how coherent the clusters are. By minimizing the intertia, the algorithm tries to minimize the difference between the mean value of a cluster and the values of points in the cluster. If a set of n samples x are inputted, the algorithm divides the samples into k clusters C. Each cluster is described by its mean u_j, or centroid. The interia of a cluster is caluclated by the

following expression:

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(\|x_j - \mu_i\|^2)$$

The inertia is not normalized, but lower values are better and zero is the optimum value. The inertia assumes that the clusters are convex and isotropic, and would not work well to cluster irregular or elongated clusters. [2]

3.2 DBSCAN clustering

The DBSCAN clustering algorithm creates clusters out of areas in the data of higher density. Unlike kmeans, it does not consider clusters to have any particular shapes, and the algorithm determines the number of clusters based on inputted parameters. Core samples are points that are in areas of high densities. The algorithm creates clusters around core samples so that the clusters consist of core samples, and non-core samples that are close to the core samples. The core samples are determined by two input parameters, the minimum samples and a specified distance, ε. A point is in the ε-neighborhood if the distance d from a point p to a point q is within a radius of of ε, as determined below:

$$N_\varepsilon(p) : \{q | d(p,q) \leq \varepsilon\}$$

High density areas have the minimum sample of values within the ε-neighborhood. By increasing the number of minimum samples, and decreasing the distance, ε, a cluster's density is increased. [2][3]

4 Current Progress

I have used kmeans clustering from the scikit-learn python package [2] to examine seismic BLRMS data. Figure 1 shows a plot of clustered data from the six seismic bands in in the X direction. There are six clusters in this graph. The kmeans clustering algorithm has been able to pick out anthropogenic noise but does not identify earthquakes well. I tried to use kmeans to identify earthquakes as a cluster by using combinations of different data channels for the clustering and by clustering the derivatives of the data. However, I was unable to use kmeans to sort the earthquakes into identifiable clusters.

I then used the DBSCAN clustering algorithm from the scikit-learn python package [2] to cluster data from the earthquake channels. Figure 2 shows a plot of the earthquake channels clustered into eleven clusters by the DBSCAN clustering algorithm. The peaks, which indicate earthquakes are all in the same cluster. However, while the DBSCAN algorithm appears to work well with earthquake channel data alone, it did not cluster the earthquakes when data from other channels was combined with the earthquake channel data.

5 Future Progress

The next step in my project is to quantitatively exaluate how well the clustering algorithms work. For the earthquake band data, one way to evaluate how well the clustering works is to compare the data to a list of known earthquakes. I will also cluster data from more sensors and try out other clustering algorithms. After seeing how well clustering algorithms work, the next part of the project is to implement neural networks to characterize the noise.

References

[1] Aurlien Gron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O'Reilly Media Inc., (2017).

[2] http://scikit-learn.org/stable/modules/clustering.html

[3] https://www.cse.buffalo.edu/~jing/cse601/fa12/materials/clustering_density.pdf

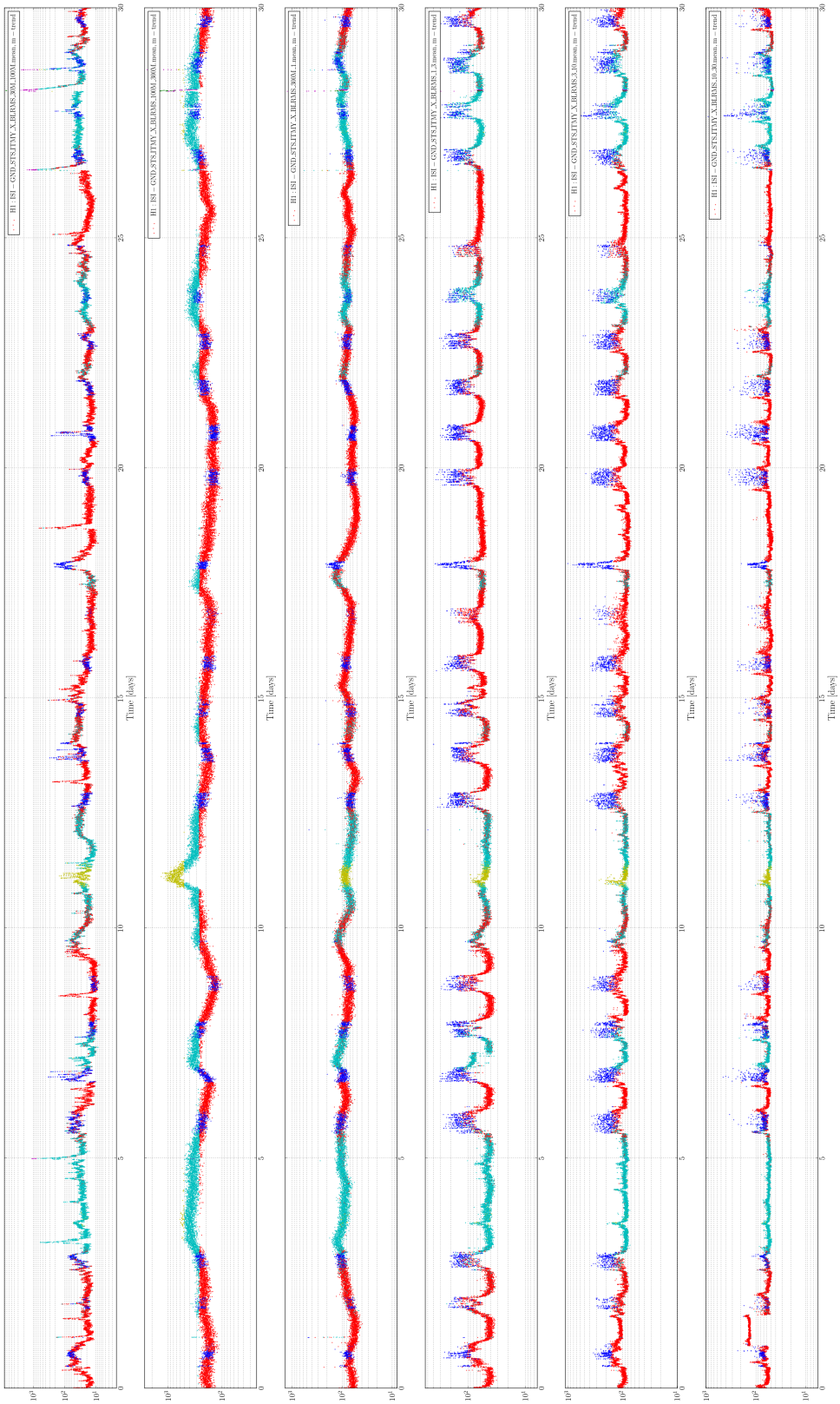

Figure 1: Plot of data from six seismic channels in the X-direction clustered into 6 clusters using kmeans

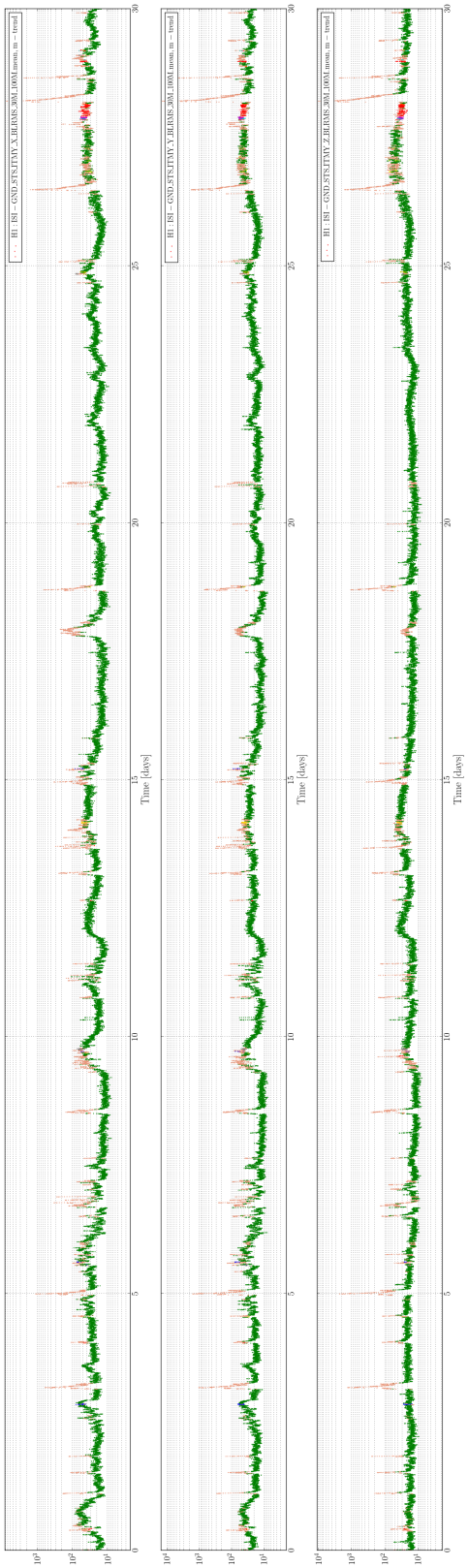

Figure 2: Plot of earthquake data from each directon clusted using DBSCAN