Extracting Events from a Super Imbalanced Time-series

Shekh Ahammed Adnan Bashir

The University of New Mexico

Department of Computer Science

adnanbashir@unm.edu

Abstract

Developing a supervised classifier requires expensive labeled data. Overfitting is another challenge and often times it is difficult to use such classifier to detect novel patterns. Unsupervised classifiers solve these two problems. However, it is difficult to design unsupervised classifier because of their requirement of profound knowledge of patterns in the data. In other words, designing a useful unsupervised classification algorithm requires vast domain knowledge. In this work, we would see how fuzzy logic-based probabilistic clustering algorithm can be used to detect tremors from Marsquake [2] seismic traces in a semi-supervised manner. This semi-supervised algorithm makes use of a new similarity measure and detects 21 new events out of around 235 events.

Introduction

A growing number of natural and induced earthquakes are taking place every year over the world. The situation now is we have a lot of seismic traces by the blessing of a plethora of stations and not a sufficient number of experts to extract events from them. This calls for automating the process. Automating the extraction of events from seismic datasets is a difficult problem. Moreover, this is also a problem which is not much explored as other problems in computer science. So not that much development in this area so long has not taken place.

Autocorrelation and Fingerprint And Similarity Thresholding (FAST) [1] are two well-known algorithms for detecting events from a seismic trace. Both of them makes use of previously seen tremor traces to detect new events. There is one more algorithm called ConvNetQuake which makes use of recently much hyped Convolutional neural networks[3]. Training time for this algorithm is very long and requires large labeled dataset.

Methods

Because of the lack of understanding of seismic traces, I did not try to design an unsupervised algorithm. Rather I designed a semi-supervised algorithm. I plotted the seismic trace and visually detected 10 events. Then used them to detect some more events. We call those events seeds. And used all these events together to detect several other events. Not all of these detections were correct. I plotted them and visually recognized which of them are tremors and which other are not. Figure 1 shows such a tremor.

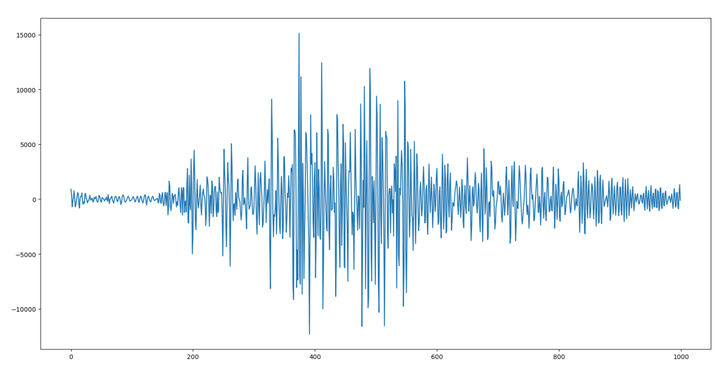

Figure 1: Trace of a tremor

Data Preprocessing

Marsquake seismic trace has a frequency of 2 Hz. The dataset has some periodicity as we see in Figure 2. This periodicity was removed using a Butterworth filter at bandpass frequency 0.2 Hz to 0.99 Hz and corner frequency of 4 Hz. A part of filtered trace is plotted in Figure 3. Only the traces from the seismic velocity channel along x-axis has been used. Empirically, traces from rest of the seismic velocity channels were found to be well agreed with the former and was thus discarded from consideration to keep things simple.

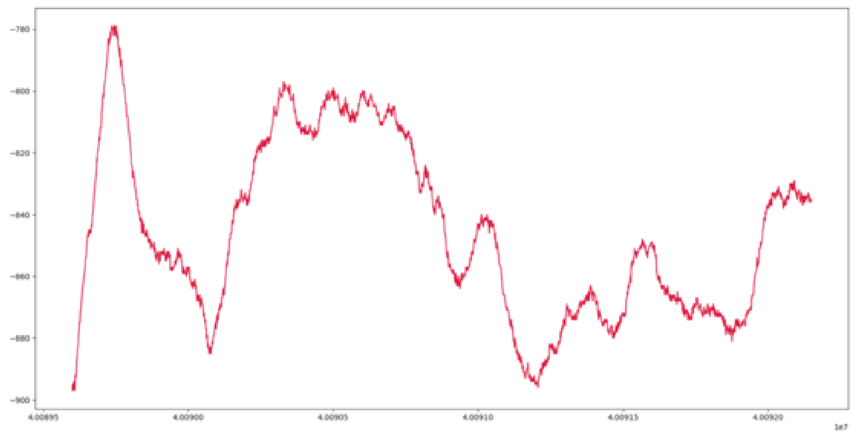

Figure 2: MHU trace before filtering from August 20, 2019, to August 31, 2019

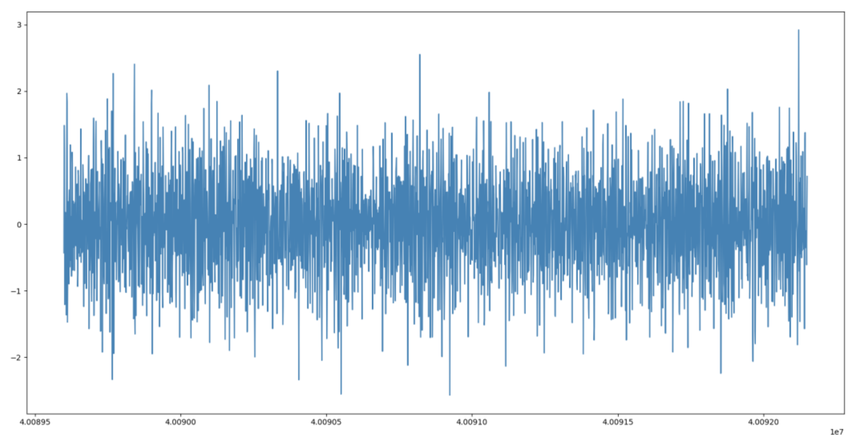

Figure 3: Filtered MHU trace from August 20, 2019, to August 31, 2019

Algorithm Outline

We first divide the entire trace in windows of length 10 seconds. This is a design choice made arbitrarily. Then we compute the probability of a window containing a tremor as following:

$$Pr(Event|w_j) = \frac{dist(w_j, C_{NEC})}{dist(w_j, C_{EC}) + dist(w_j, C_{NEC})}$$

where $dist(w_j, C_{EC})$ is the similarity between a j^{th} window in the trace and a window containing a tremor.

A new Similarity Measure

I introduced a new similarity measure to calculate similarity between windows. Let's call it *adist* and is defined as follows:

$$adist = \Sigma_i(sorted(window_i^{(a)}) - sorted(window_j^{(b)}))$$

In other words, this is the manhattan distance between two sorted windows. The intuition behind this similarity measure is that we ignore the time order and capture the magnitude order. The motivation is that, due to arbitrary change in the earth structure due to previous tremor, magnitude might vary. I admit that, this is something, I need to investigate further.

Results

I have found 21 new events beginning around at the following positions 6242400, 11037600, 14954400, 19382400, 2099
I have filtered out all the consecutive tremors within an hour. I have confirmed these to be tremors by visual inspection. We note at this point that, I plotted a span of an hour to visually recognize these to be tremors.

Conclusion and Future Work Directions

Unavailability of proper similarity measure and an attempt to keep the algorithm trivially parallelizable are the root of all the problems in this work. A similarity graph for each "tremor-like" window might alleviate this problem. For approaches requiring a loss function, the Loss function then is to be defined in such a way that it is insensitive to spikes in a window and also takes into account of the fact that S-wave reaches seismograph later than P-wave. The similarity measure used here could be improved by involving some rectification in its definition.

References

[1] K. J. Bergen C. E. Yoon, O. OReilly and G. C. Beroza. Earthquake detection through computationally efficient similarity search. *Science*, 2015.

[2] Clinton et al. Preparing for insight: an invitation to participate in a blind test for martian seismicity. *Seism. Res. Letters*, 2017.

[3] T. Perol, M. Gharbi, and M. A. Denolle. Convolutional neural network for earthquake detection and location. 2017.