Datamining and Machine Learning

# Earthquake aftershock analysis

This is another interesting dataset that contains data on aftershock activity that followed after the large 4 September 2010 Darfield Earthquake in Canterbury, New Zealand. The objective in this dataset is to conduct a spatial analysis of Earthquake. The objective is to track shock magnitude and shock depth by latitude and longitude.

The data is sourced from GNS Science New Zealand https://www.gns.cri.nz. Some pre-processing has been done however, further preprocessing needs to be done.

The data will be mined through a clustering approach. Experiment with 3 different clustering algorithms, one of which must be the Self Organizing Map (SOM). If you are using Weka, please note that this algorithm does not come built into Weka and so you will need to download it using the Package Manager.

After pre-processing the data apply each of the clustering algorithms and visualize the clusters in order to gain a visual understanding of the quality of the clusters produced – bear in mind that we seek a spatial understanding of aftershock activity.

For a fair comparison we will keep the number of clusters the same across all clustering algorithms. In order to set the number of clusters, first apply SOM. SOM will automatically determine the number of clusters from the grid that you specify and you can then use this number with your other clustering methods (for e.g. k means).

Once you have compared the 3 algorithms on the basis of visualization you need to compare cluster quality on the basis of two quantitative measures:

1. Average sum of squares taken over all clusters.

2. Cluster Silhouette measure, (see https://en.wikipedia.org/wiki/Silhouette\_(clustering) for an explanation on how this is computed).

Now compare the 3 algorithms on these two measures. In your analysis section of your report you can comment on consistency between the visual comparison and the quantitative comparison.

Part 2:

In the last part of your experimentation take the winner algorithm (i.e. the best clusterer) and perform a classification analysis in order to find out the spatial location of the largest aftershocks. In order to classify aftershocks we can convert clusters to classes by applying the AddCluster filter in Weka.

Once the conversion is done apply a classifier such J48 or Naïve Bayes to generate a model and answer the question: Where in space (i.e. latitude and/or longitude) does the highest magnitude earthquakes lie?

# Introduction

The objective of the data mining assignment is to conduct knowledge discovery on aftershock activity data from the 2010 Darfield Earthquake in NewZealand to establish patterns of magnitude in relation to longitude and latitude and depth in relation to longitude and magnitude.

The task involves descriptive mining to establish patterns and meaningful clusters within spaces that may have multiple dimensions i.e. longitude, latitude + magnitude or longitude, latitude + depth. The idea is to establish association rules between location of highest magnitude and the distributions of shocks and magnitude of shocks in the surrounding area. This is to establish spatial understanding of shock areas. Temporal monitoring although vitally important to track the progress and evolution of Earthquakes is not taken into consideration.

3 algorithms are considered for to achieve the knowledge discovery:

* Self-Organising Map (SOM) / Self-Organising Feature Map (SOFM)
* K- means clustering
* DBSCAN

The Algorithms are to be evaluated

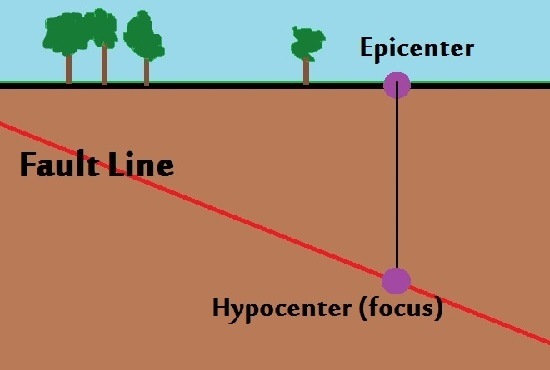
The data for this assignment was recorded from the fateful day of 3rd of September 2010 until the end of the month on the 30th of September 2019.

# Literature Review

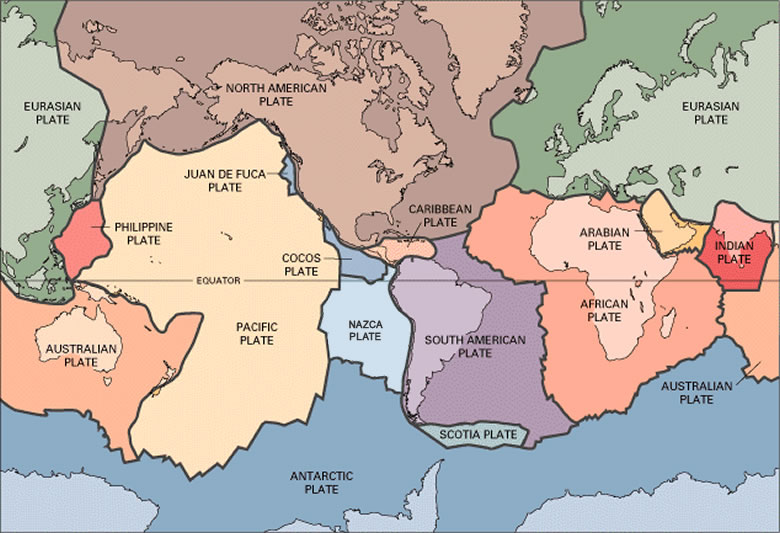
Earthquakes are natural phenomena where the surface of the Earth experiences violent shaking following the collision of continental plates that make up parts of the Earths lithosphere. Depending on the magnitude/ seismicity of these events their impact can range from hardly being felt to being capable of toppling buildings and causing severe damage to cities and although they are a cause of much human suffering and loss earthquakes are a natural part of living on planet Earth.

The primary source of the seismic activity of an earthquake is the release of strain energy from the focus also known as the hypocentre. As two continental plates meet each other faults which can be thought of as cracks began to build up in the rock to a point where it is released. This release of energy produces propagating waves that travel to the surface where it causes vibrations and thus shaking effects. The surface point directly above this hypocentre focus is called the epicentre and is the point of highest magnitude because it is the place where these waves first reach the surface in the shortest distance (straight line).

The Epicentre is given a significant focus in the data mining operation as we wish to discover the distribution of Earthquakes with respect to the point of highest



Because of their origin from colliding tectonic plates, earthquakes only happen in selected parts of the world. By observing a simple mapping of Tectonic Plates in figure 1 it can be observed that New Zealand is located at the meeting points of Pacific and Australian plates thus making that region prone to earthquakes and aftershocks that follow the main event.



Data Mining with Clustering

Data mining earthquakes

# Data

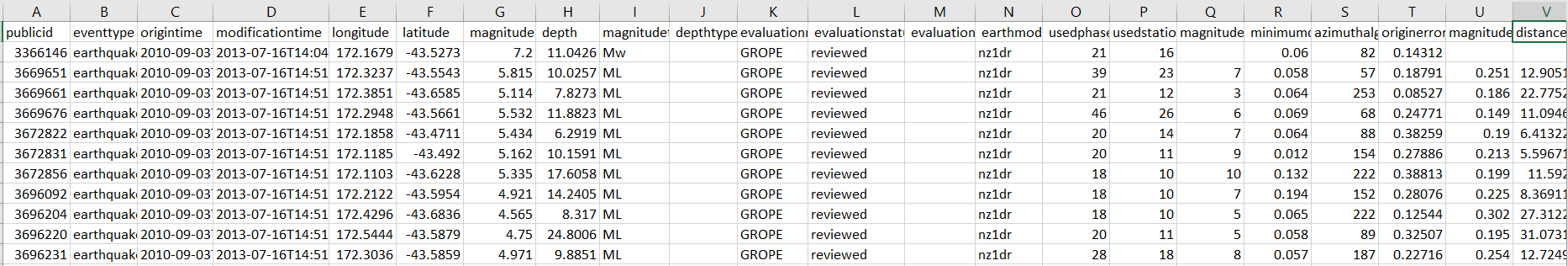
The data supplied was

## 

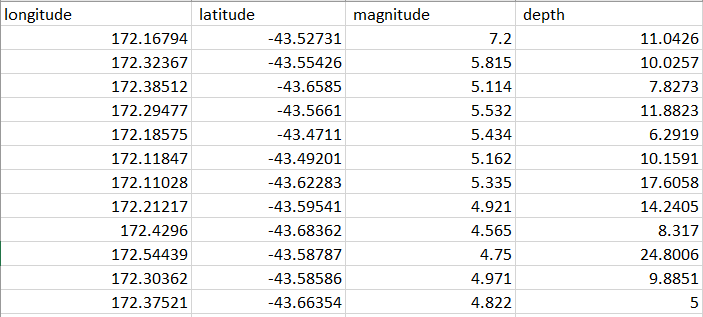
# Pre-processing

## Trimming of Data Table

The original dataset provided contained 22 attributes as shown below.



To ease data mining task, we parse the attributes into the only 4 that matter: longitude, latitude, magnitude, depth. The final table looks like this. This is saved into a new csv file.



After loading the data into python and converting into nd array with this code:

dFrame = pd.read\_csv('aftershockLLMD.csv')  
quakeRecArr = dFrame.to\_records(index=False)  
quakeLLMD = np.array(quakeRecArr.tolist())

We can then easily access the attributes we need each parameter with the following python code:

quakeLLMD[:,0] #for all values of magnitude  
quakeLLMD[:,1] #all of latitude  
quakeLLMD[:,2] #all of magnitude  
quakeLLMD[:,3] #all of depth

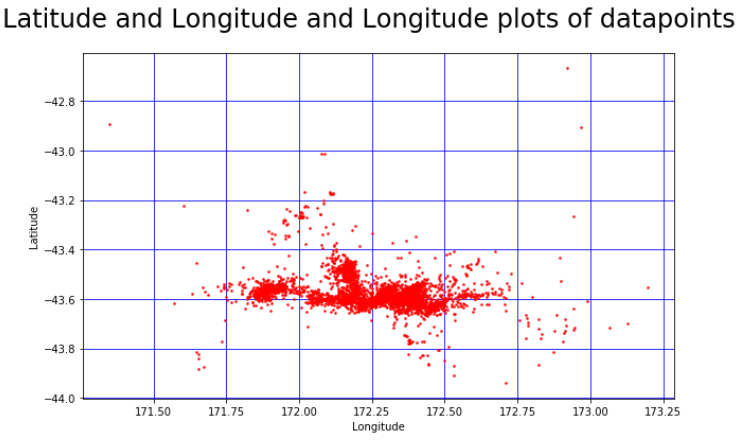
Ndarray data type is chosen because it is faster and most of our models interact best with this data structure.

## Rounding of Figures

We want to make sure that all data is rounded to significant figures.

# Pre-visualizing

Before Any algorithm is implemented visualization is done to better understand the data prior to clustering. The simplest object we wish to understand is the spatial location of the data points which is a mapping of longitude and latitude



# Algorithm and Mining

## Self-Organizing Feature Map

Self-Organizing Feature or SOM is a type of Artificial Neural Network that maps higher dimensional input layer vector data to a grid of output neurons. Unlike convolutional neural network, it has no hidden layers.

It is most commonly used in unsupervised machine learning knowledge discovery to find cluster relations in data without any prior knowledge of the data itself.

SOM works by mapping close instances from a in multidimensional data set to the same neuron or a local group of neurons. The result is a 2D weight/ intensity map that approximates the distribution of multidimensional data.

Advantages vs Disadvantages

|  |  |
| --- | --- |
| Advantages | Disadvantages |
|  |  |

The SOM implementation used in this mining assignment is SOMPY for python. Currently released for python 2 however, to make it work for python 3 some features are disabled.

After downloading the package, we import to notebook. The only parameter supplied to the SOM is the map size n x m.

#we have agreed that this looks like 4 lcusters. So trace the

map\_labels = oneSevOneSevSom.cluster(n\_clusters=4)

data\_labels = np.array([map\_labels[int(k)] for k in oneSevOneSevSom.\_bmu[0]]) # mapping labels from grid to original data

The trick to implementing the SOM is to find an ideal map size that determines a right number of clusters. There is no perfect number of clusters especially for real world datasets. So, for this experiment we will iterate from small map size, to ideal, and a large size.

We find the clusters based on U matrix. U matrix Definition:

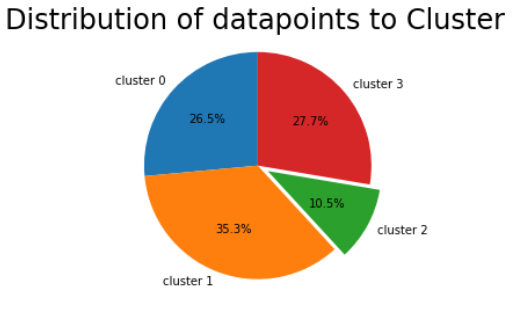
### Longitude Latitude Magnitude

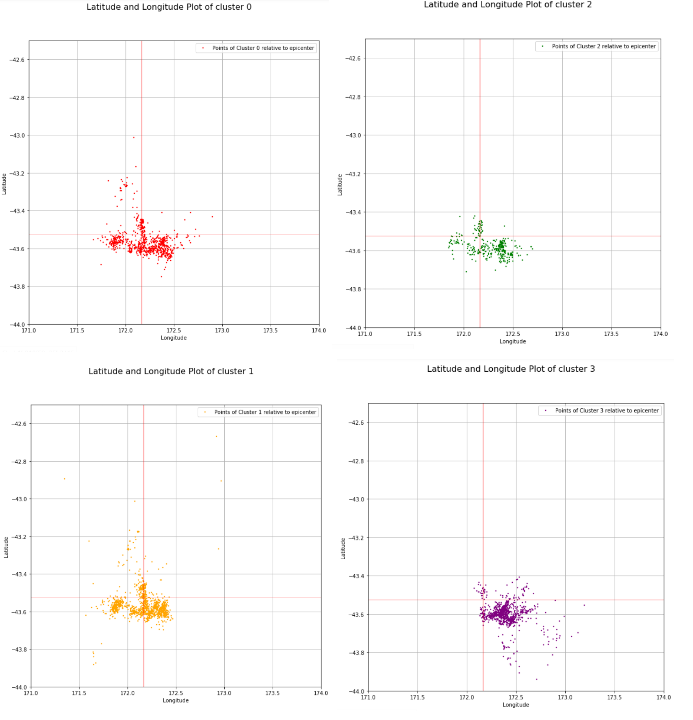
We wish to understand the relationship between Location (Longitude/ Latitude) of data to the magnitude. We implement SOM of various sizes

|  |  |  |  |
| --- | --- | --- | --- |
|  | 6x6 | 17x17 | 50x50 |
| Umatrix |  |  |  |
| N clusters discovered | 1 ~ inconclusive | 4 | 5 |
| Training Time (s) | 0.253000 | 0.801000 | 17.24500 |
| Quantization Error | 0.746467 | 0.320556 | 0.138652 |

We think 4 clusters from map size 17 by 17 is sufficient. SOMPY allows us to assign clusters to datapoints.  
  
Based on the Result

map\_labels = oneSevOneSevSom.cluster(n\_clusters=4)  
data\_labels = np.array([map\_labels[int(k)] for k in oneSevOneSevSom.\_bmu[0]])   
map\_labels



Now visualize clusters:

Interpretation.

We can see there is some similarity

### Longitude Latitude Depth

|  |  |  |  |
| --- | --- | --- | --- |
|  | 6x6 | 17x17 | 50x50 |
| Umatrix |  |  |  |
| N clusters discovered | 2 | 4 | 4 |
| Training Time (s) | 0.272000 | 0.835000 | 17.192000 |
| Quantization Error | 0.561950 | 0.217925 | 0.080829 |

Visualize, we choose 4 clusters but the cluster labels are derived from the 50x50 SOM.

## K Means Clustering

K means is the most popular

The disadvantage of the K Means clustering is that we have to supply it with how many clusters it should divide the data into. This prevents us from achieving true knowledge discovery of finding the ideal clusters that the data should be divided into.

K = 4

## DBSCAN Density-based spatial clustering of applications with noise (DBSCAN)

K = 4

# Evaluation

## Self Organizing Map

### Average sum of squares taken over all clusters.

### Cluster Silhouette measure

## K Means Clustering

### Average sum of squares taken over all clusters.

### Cluster Silhouette measure

## DBSCAN Density-based spatial clustering of applications with noise (DBSCAN)

### Average sum of squares taken over all clusters.

### Cluster Silhouette measure

# Part 2

Conversion

# Conclusion

\*Insight Dr.

# References

[1] Vahid Moosavi SOMpy