COS424 Project: Forecasting earthquakes

Sukriti Singh
Department of Computer Science
Princeton University
sukritis@princeton.edu

Abstract

Forecasting earthquakes is an active area of research in geological research. Current earthquake warning systems are severely limited in their ability since it often only provide a few seconds warning, which in many situations would not be enough. In this project, we try to use machine learning and optimization techniques to develop a method to forecast or predict an earthquake from the raw seismic data. Various regression methods are applied taking the raw fields as well as wavelet decomposition of the original seismic data as input. Through quantitative analysis, we show the random forest regression with wavelet based decomposition outperforms conventional overcomplete features computed over raw and Fast Fourier transforms of the data. We also observed the shortcomings of the implemented method in its ability to forecast distant earthquakes as well as immediate ones.

1 Introduction

Forecasting earthquakes is one of the most important problems in Earth science because of their devastating consequences. Current scientific studies related to earthquake forecasting focus on three key points: when the event will occur, where it will occur, and how large it will be [3]. To facilitate scientific progress on earthquake prediction, Los Alamos National Laboratory (LANL) [4] and its collaborators has developed an apparatus to study earthquake physics in a laboratory setup, and have created a dataset with real-time seismic data with approximate time to the next lab earthquake. Our goal in this work is to use machine learning to analyze such acoustic data and train models that can predict when the next lab earthquake will take place. If such a model can be obtained and the technology can be successfully transferred from lab setup to earthquake prediction, it would unlock the possibility of early prediction systems which in turn could help save lives. [14] suggests that there is indeed physical basis to expect the technology transfer on such a scale is indeed feasible.

Following initial work by LANL [14], we use feature extraction methods to represent the data blocks of acoustic data using a set of features that can capture signal's spatial distribution as well frequencies; specifically, this include computing statistical measures such as mean, median. In addition to computing the statistics on raw data, we also consider the Wavelet transform and compute similar statistics on the transformed signal. Next, we experiment with various regression algorithms - linear, support vector machines, random forests and gradient boosting from the SciKit-Learn Python library [12] to train models for prediction. In order to obtain the best model parameters, we would perform the grid search over the parameters by splitting the available training data into training and validation subsets.

2 Related Work

Advances in machine learning as well as improved sensing methodology over the recent years has triggered significant research activity focused on improving forecasting models, especially in

labquake prediction [14, 9, 13]. LANL's initial work [14] showed that the prediction of laboratory earthquakes from continuous seismic data is possible in the case of quasi-periodic laboratory seismic cycles. They used supervised machine learning to train a random forest regressor [5] to predict the duration to the next failure. More specifically, over 100 statistical features are computed on the raw acoustic signals over time windows and used as inputs to the random forest regressor which is trained to minimize the time to failure. The reported experimental results suggest that catastrophic earthquake failure may be preceded by an organized, potentially forecastable, set of processes. [9] use similar framework to process the raw seismic signals to estimate the fault displacement rate.

While the observations from the initial work at LANL [14] is indeed encouraging, however, it was limited to data with quasi-periodic laboratory seismic cycles. To further research in more challenging settings, Los Alamos has launched a Kaggle competition [3] in early 2019; in this competition, the team has provided a much more challenging dataset with considerably more aperiodic earthquake failures. The competition has indeed triggered significant experiments on the challenge website, where participants collaboratively develop solutions that would help solve the challenge. To this end, several methods to compute statistics of the acoustic signals, such as those proposed in LANL's initial work [14] are readily available [2, 1]. Given acoustic signal over a specific time interval, the set of features include statistical measures such as mean, median, running mean over different time window sizes, harmonic mean, Fast Fourier transform based features [7] and among many others. These methods then often employ various regression methods such as Random forest regression [5], Gradient boosting [6].

3 Dataset: LANL Earthquake Prediction

The data is provided as part of the Kaggle challenge, titled "LANL Earthquake Prediction" [3]. The seismic data is recorded using a piezoceramic sensor, which outputs a voltage upon deformation by incoming seismic waves. The seismic data of the input is this recorded voltage, in integers. This input data is essentially a chunk of 0.0375 seconds of seismic data (ordered in time), which is recorded at 4MHz, hence 150000 data points. The expected output is the time remaining until the following lab earthquake (in seconds), which is essentially the delta between current time and the time of failure, which is based on a measure of fault strength (shear stress).

The challenge dataset is split into training and test data. The training data is a single, continuous segment of experimental data. The test data is a set of several small segments. The data within each test file is continuous, but the test files do not represent a continuous segment of the experiment. Note that even though both the training and the testing set come from the same experiment, there is no overlap between the training and testing sets, that are contiguous in time. The time of failure information is not available for the test data, however, the predictions on the test data can be evaluated using challenge submission webpage [3], which reports the mean absolute error between the prediction and Ground truth.

A sample seismic data sequence is shown in Figure 1.

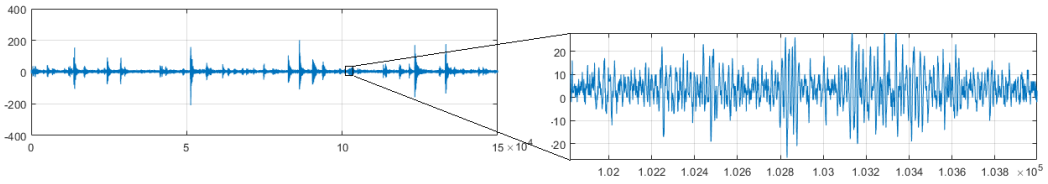

Figure 1: A sample input sequence of seismic data.

4 Benchmarking regression for failure prediction

Most, if not all, existing machine learning methods for failure prediction start by first computing a set of features on the data, and then train a regressor to predict failures. We take a similar approach in this work. In this section, we first present a brief summary of frequently used features for analyzing

seismic signals, followed by a comparison between various regression models. Instead of using statistical measures on raw data as features, we use statistics based on wavelet transforms on the raw data, which to the best of our knowledge is not being considered in the experiments we have seen so far. For regression analysis, we have considered linear regression, support vector regression, random forests regression as well as gradient boosting.

4.1 Feature extraction

Over the years, several statistical measures have been discovered that can capture different properties of the underlying signal. For instance, median of signal over time is more robust to outliers compared to mean. While these statistics are indeed relevant and potentially capture the key properties of the underlying signal, there is likely to be loss of information. To this end, a common practice is compute an overcomplete set of features and use machine learning to select a subset of relevant features. However, the overcomplete feature itself has constraints based on the size of the data; if the size of the data is small and number of features are significantly higher then its possible the machine learning may not train well due to overfitting. In this work, we suggest to use wavelet decomposition of the signal (which encodes both spatial and frequency features) and compute features on the wavelet coefficients; this feature space is not only smaller compared to the feature space being used by others, it would likely cover all areas of the signal and allow machine learning to discover more interesting patterns.

4.1.1 Baseline: Over-complete feature basis

Here we list frequently used features in the area of the earthquake prediction [14] (and Kaggle kernels).

- Mean, standard deviation, median, geometric mean, harmonic mean
- Percentiles for absolute values (at every 5% interval)
- Auto-correlation with different offsets (5, 10, 50, 100, 500, 1000, 5000, 10000)
- Entropy at various bin sizes (at every bin size of 5, starting from 1 to 99)
- Rolling mean and standard deviation with different window sizes e.g. 5, 10, 100, 500.

Besides computing above statistics over the entire signal or time window, it's common to also compute them over certain areas or segments of interest. For instance, a smaller windows close to the beginning or the end of the signal. Figure 2 shows several segments of interest (time windows) being used to compute the statistics. These statistics are then concatenated together to form the full feature vector.

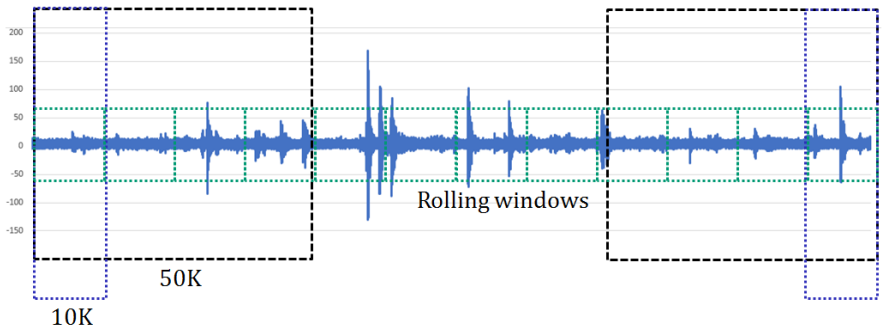

Figure 2: A sample input sequence of seismic data.

In addition to the above statistics on the raw signal, similar statistics are also computed on the Fast Fourier transform [7] of the signal.

4.1.2 Features from Wavelet decomposition

Instead of computing an over-complete set of sparse features over the raw signal, we use wavelet decomposition to represent the signal and compute statistics over the wavelet coefficients. The choice of wavelet decomposition to represent the seismic signal is motivated by the prior work [10, 8]; authors show that it is accurate to analyze the seismic oscillation by wavelet transform method, and has advantages of the conventional frequency-domain analysis and time-domain analysis.

In this work, we use the multilevel discrete wavelet transform with Mallat decomposition [11], which involves iterative decomposition of only the approximation subband at each subsequent level. This is illustrated in figure 3 (a). Here we use the Daubechies wavelets [8] due to its use in other recent works on seismic data. Figure 4 shows the wavelet decomposition of the 2 different raw signals. Notice that the responses are high around the spikes in the raw signal.

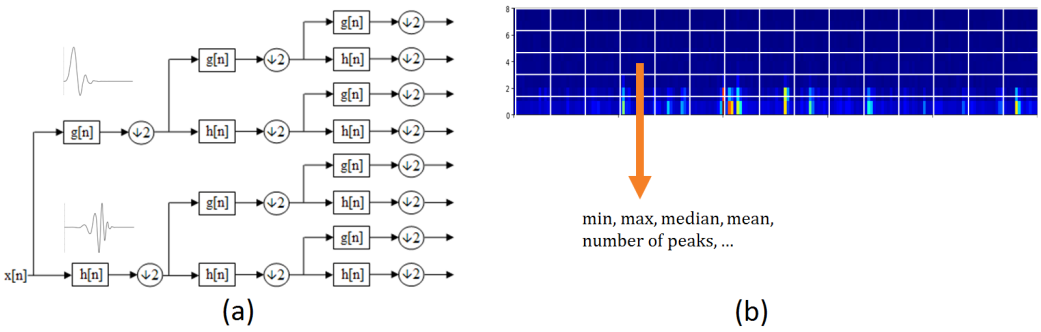

Figure 3: Wavelet based feature extraction (a) Wavelet decomposition at 3 levels; g[n] is the low-pass approximation coefficients. h[n] is the high-pass detail coefficients (b) Feature computation over the wavelet coefficients; this is done by imposing a grid and computing statistics for each cell.

The wavelet decomposition of the input signal is just as long as a the raw signal i.e about 150K time steps. Thus, for efficient analysis, we downsample this data and compute the statistics on the downsample data. Note that naive downsampling may lose the necessary detail information captured in the wavelet coefficient; hence, we impose a grid over the wavelet coefficients (8 x 15 cells) i.e. for each wavelet response channel, we have 15 evenly spaced cells, and then for each cell, we compute statistical features such as min, max, median, mean, number of peaks at window size 5 or 10 or 50; we used the Python tsfresh library to compute these features.

4.2 Regression analysis

After computing the feature using either of the feature extraction methods, next step is to train the regressor. Although prior works [14] used Random forest regressor, here we experiment with different regression methods, namely

- Linear regression with L1 and L2 sparsity (Elastic net)
- Support vector machine with linear kernel
- Random forest regression
- Gradient boosting regression

We use SciKit-Learn Python library [12] to train all the regression models. To select the best model, we used the *GridSearchCV*() [12] to perform grid search over the model parameters to select the best model. Note that since the time to next earthquake is not available for the test dataset, we split the available training dataset into separate train and test dataset, where test dataset is completely unseen during training including parameter selection. In the following section, we explain the experiment dataset, followed by quantitative comparison between various regression methods, and finally, a detailed analysis of the performance of the best performing regression method (random forest regression).

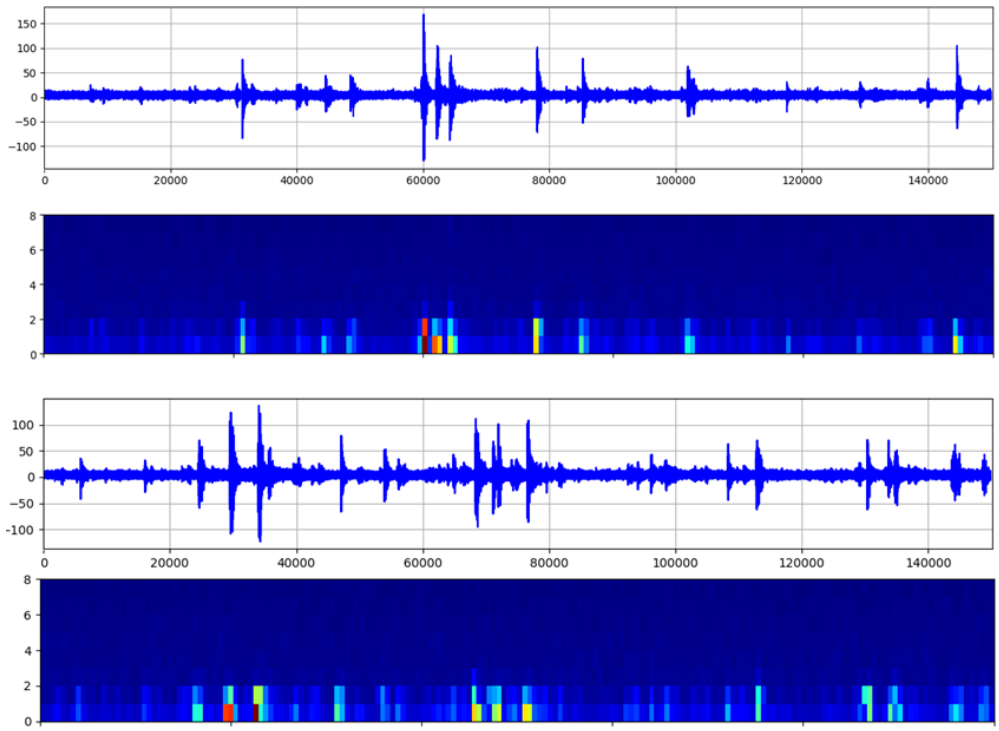

Figure 4: Wavelet decomposition of the raw signals. The wavelet coefficient map (shown in jet colormap) is rescaled to fit the width of the raw signal to easily appreciate the wavelet responses with the corresponding position in the raw data.

4.2.1 Experiment dataset

The provided dataset is a long sequence of seismic activity together with the time to the next lab earthquake known at every instant. We sample sequences of length 150000 time steps from this long sequence; the window size of 150K is used since it matches the size of the actual testing dataset from the challenge [3]. We sample at interval of 50K time steps, resulting in a total of 12574 samples. Given these samples, we use the first 90% for training (11320 samples) and rest for testing (1254 samples). We intentionally choose the train and test samples to be one after another, to avoid overlap between the time steps. For model selection by cross validation, only the training data is used. The testing dataset is completely unseen during training.

4.2.2 Quantitative analysis

Splitting the training sequence from the challenge into training and test dataset allowed use to have Ground truth (time to next quake) for both training and test datasets, and thus enable quantitative comparison between different methods. To select the best model, we used the Scikit learn *GridSearchCV*() [12]; due to computation resource constraints we limitted the grid search to only a certain number of estimators for random forest and gradient boosting methods. For both methods, using more estimators helped improve the performance and we ended up using 250 estimators.

We ran experiments with both feature extraction methods as well as with different regression methods. For feature extraction, we used the overcomplete feature basis (together with Fast Fourier transform) as the baseline and compared it with the wavelet based features. For wavelet based features, we compare various regression methods. Table 4.2.2 shows the comparison of various regression methods with different feature extraction methods. Note that the Random forest regression with wavelet based features perform better than all other methods. This, on one hand, validates the observation that Random forest regression seems to be performing better than other methods, but it also shows the added value of using wavelet based features in comparison to the other features.

Features	Regressor	Mean Absolute Error	80 percentile	90 percentile
FFT (1419)	Random forests	4.2	6.78	9.07
Wavelets (280)	Random Forests	**4.14**	**6.68**	**8.82**
Wavlets	Gradient Boosting	4.21	6.84	9.1
Wavlets	Linear SVR	4.31	6.91	9.37

For a more detailed comparison between the feature space, we also present correlation plots, shown in Figure 5. The points in blue and orange are the training and test samples respectively. Notice that the wavelet based features clearly performs better.

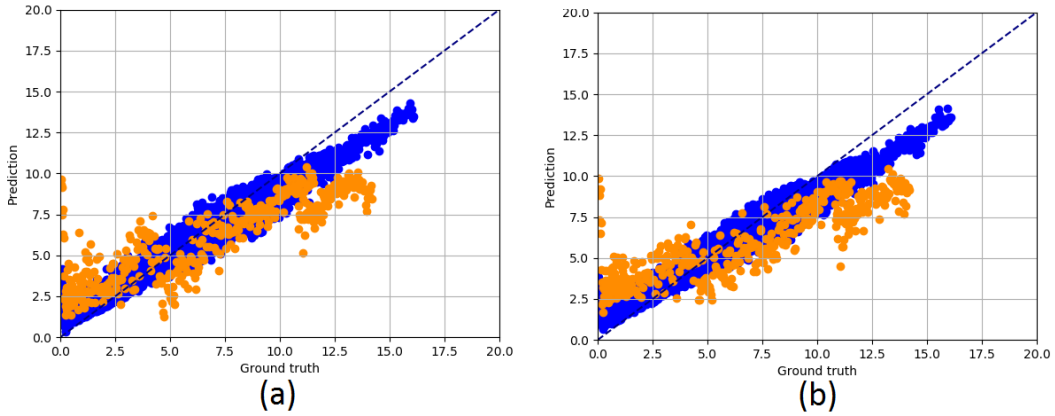

Figure 5: Correlation plot on training and test datasets using (a) Overcomplete feature basis with FFT (b) Wavelet based features.

4.2.3 Analysis of trained regressor

Having identified the random forest regressor with wavelet features as input to be the best performing method, we further analyzed the model and the practical impact of the results. To this end, we first studied which features are more informative towards the prediction. We studied the feature importance of the Random forest regressors and observed the number of peaks with neighborhood size of 10 as well as absolute maxima features were among the best features. This is somewhat expected since the absolute maxima of the wavelet coefficients, especially closer to the end of the video, would likely to capture the vibrations that are likely to correlated with a potential quake.

We also compared studied the prediction performance of the trained model by plotting the time to the next quake over samples. Figure 6 shows the time to the next earthquake for samples (every third sample) in the testing dataset. We notice that the regressor fails to the predict the distant as well as immediate events. While it is expected the the distant features may be hard to predict, however, its inability to predict immediate events is concerning. Further investigation is needed to understand what this may be the case.

5 Conclusion

In this work, we attempted to develop a method to predict the time to the next lab earthquake only from seismic data. We used the data provided as part of the Kaggle challenge and demonstrated that the using wavelet decomposition based features, together with Random forests, performs better than using an overcomplete set of the features (using Fast Fourier transform). We observed the current regressor did not perform as well as needed especially when the time to next quake is small. In order to improve the performance, an immediate step should be greatly increase the dataset by sampling the 150K windows at smaller intervals (5K) instead of 50K. Furthermore, we can try to use a less coarser representation of the wavelet coefficients.

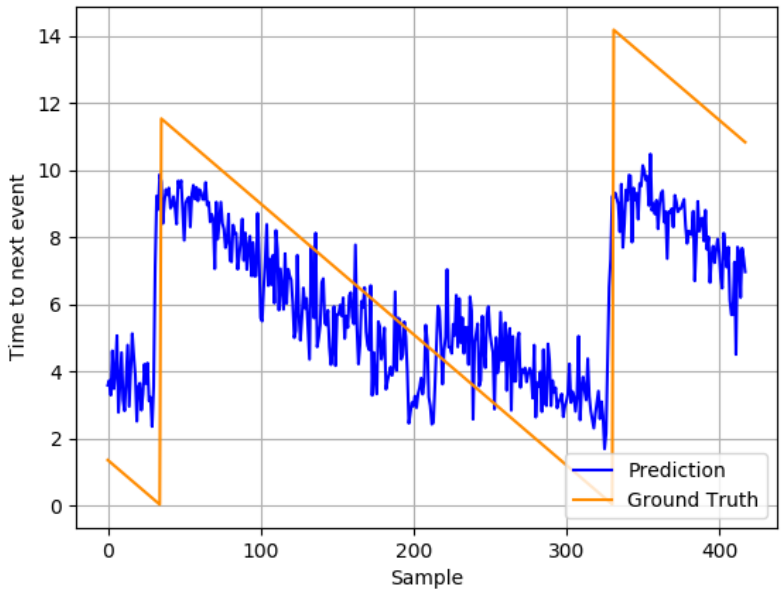

Figure 6: Time to next earthquake event over samples in the testing dataset.

References

[1] Kaggle kernel: Even more features. `https://www.kaggle.com/artgor/even-more-features`. Accessed: 2019-05-05.

[2] Kaggle kernel: Lanl earthquake eda and prediction. `https://www.kaggle.com/gpreda/lanl-earthquake-eda-and-prediction`. Accessed: 2019-05-04.

[3] Lanl earthquake prediction. `https://www.kaggle.com/c/LANL-Earthquake-Prediction/overview/`. Accessed: 2019-04-23.

[4] Los alamos national lab. `https://www.lanl.gov/`.

[5] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[6] Jerome H. Friedman. Stochastic gradient boosting. *Comput. Stat. Data Anal.*, 38(4):367–378, February 2002.

[7] W. M. Gentleman and G. Sande. Fast fourier transforms: For fun and profit. In *Proceedings of the November 7-10, 1966, Fall Joint Computer Conference*, AFIPS '66 (Fall), pages 563–578, New York, NY, USA, 1966. ACM.

[8] Ali Heidari, Jalil Raeisi Dehkordi, and Reza Kamgar. Application of wavelet theory in determining of strong ground motion parameters. *International Journal of Optimization in Civil Engineering (BHRC)*, 8:103–115, 01 2018.

[9] Claudia Hulbert, Bertrand Rouet-Leduc, Paul A. Johnson, Christopher X. Ren, Jacques Rivière, David C. Bolton, and Chris Marone. Similarity of fast and slow earthquakes illuminated by machine learning. *Nature Geoscience*, 12(1):69–74, 2019.

[10] Bi Jun-Wei, Wu Zuo-Ju, Wang Zhi-Jia, Ouyang Fang, and Cao Yuan. Wavelet transform and its application in earthquake engineering. pages 1126–1128, 06 2014.

[11] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(7):674–693, July 1989.

[12] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, November 2011.

[13] Bertrand Rouet-Leduc, Claudia Hulbert, and Paul A. Johnson. Continuous chatter of the cascadia subduction zone revealed by machine learning. *Nature Geoscience*, 12(1):75–79, 2019.

[14] Bertrand Rouet-Leduc, Claudia Hulbert, Nicholas Lubbers, Kipton Barros, Colin J. Humphreys, and Paul A. Johnson. Machine learning predicts laboratory earthquakes. *Geophysical Research Letters*, 44(18):9276–9282, 2017.