Early Forecasting of Quakes via Machine Learning

Chester Holtz, Vignesh Gokul

1 Abstract

An important goal in seismology is the ability to accurately predict future earthquakes before they occur. Anticipating major earthquakes is very important for short-term response - i.e. preparation of emergency personnel and disaster relief. In seismology, earthquake prediction is well defined: the identification of severity, bounded geographic region, and time window in which a quake will occur with high probability. We plan to approach earthquake prediction from the perspective of computer science. In particular, we will apply efficient techniques from predictive machine learning and statistics to a restricted version of this problem - prediction of the time-to-failure or time-to-fault.

2 Introduction

Earthquake prediction is a well-studied problem. However, there is a gap between the application traditional statistics-based modeling and modern machine learning-based methods. In this project, we explore the application of a broad set of approaches and techniques from machine learning, statistics, and optimization including deep neural networks (LSTM, CNN, WaveNet), sparse quantile regression, and other online regression algorithms. In addition to applying these techniques to do prediction on the raw signal, we also experimented with various pre-processing and feature learning algorithms (Robust PCA, MFCC features, and spectral embeddings). Our intent is to design algorithms that are effective for forecasting quakes, but also to make sure that they are efficient (fast, low footprint) enough to potentially run on embedded monitoring devices in the field. Thus, we propose to evaluate the efficiency of inference of our algorithms by calculuating a normalized sparsity metric. We leveraged a gold-standard synthetic & real dataset released by Los Alamos National Laboratory. The data is hosted at `https://www.kaggle.com/c/LANL-Earthquake-Prediction/data` and consists of simulated acoustic waveform signal.

In summary, our contributions include:

- Design an accurate end-to-end framework design for earthquake prediction
- Validate a variety of prediction algorithms with special focus on systems that can adapt online - i.e. learning with only one pass through the data.
- Investigate techniques to deal with a signal that is grossly corrupted by noise.

In section 3 we review the technical material behind our project and give an overview of the data, our preprocessing techniques, feature selection algorithms, and prediction algorithms. In section 4 we describe the milestones we accomplished over the quarter. In section 5 we conclude by summarizing our results and offering some interesting ideas for future work.

2.1 Prior Work

The data for this challenge comes from a gold-standard laboratory earthquake experiment that has been studied in depth as an synthetic analog of seismogenic faults for decades. A number of physical laws widely

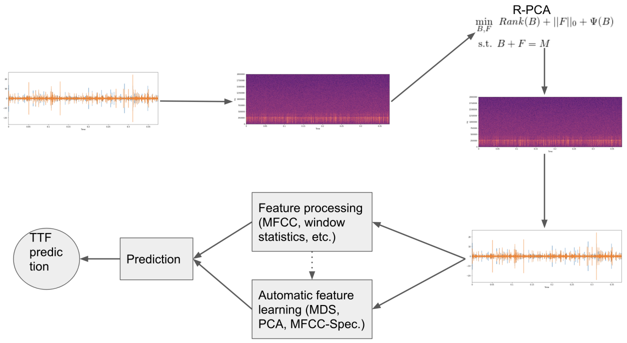
R-PCA
$\min_{B,F} \; Rank(B) + ||F||_0 + \Psi(B)$
s.t. $B + F = M$
Feature processing (MFCC, window statistics, etc.)
Automatic feature learning (MDS, PCA, MFCC-Spec.)
Prediction
TTF predic tion

Figure 1: Prediction pipeline.

used by the geoscience community have been derived from this earthquake data that have been validated on real earthquakes.

Earthquake Prediction and Analysis

There has been significant prior work on earthquake prediction. [20, 18, 8] performed a machine learning-based analysis using the same techniques used to produce our data and evaluated their methods on time-to-failure prediction. They conclude that random forests are effective for this task. [19] provided initial results on applying the methods developed on lab data to field data with success, on a particular type of earthquakes known as slow earthquakes. [13] leveraged lab data to predict earthquake magnitudes via regression using random forests.

Orthogonal to learning-based methods, earthquakes have traditionally been modeled as point processes, e.g. [2, 15].

An important facet to seismic wave analysis is pre-processing of the waveform. Traditional approaches include a manual analysis of the waveform in its frequency domain. Additional approaches include denoising via Robust Principal Component Analysis (R-PCA) [3] as in [1, 6, 12] and dictionary learning [5]. Recent methods have expanded this prior work to leverage nonlinear methods to analyze seismic data including Spectral Graph Laplacian-based approaches [16, 16] and Multiscale Principal Component Analysis (M-PCA) [9].

Neural network-based methods applied to modeling earthquake events have also show promise [17]. We leveraged convolutional neural networks (CNNs) [10] long short-term memory networks (LSTMs) [7] and their joint architecture [23] for representation learning and modeling earthquakes. In addition we evaluate a recent technique developed for audio processing: Wavenet [22].

3 Technical Material

In this section we review the technical details of the algorithms we applied to this problem and detail our analysis and numerical results. We decompose the prediction problem into stages as in Fig 1.

3.1 Data Overview

Recall that our goal was to build a model that predicts the time remaining before failure from a chunk of seismic data.

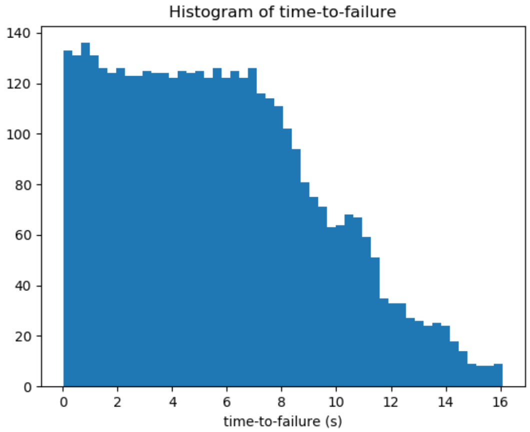

Figure 2: TTF distribution.

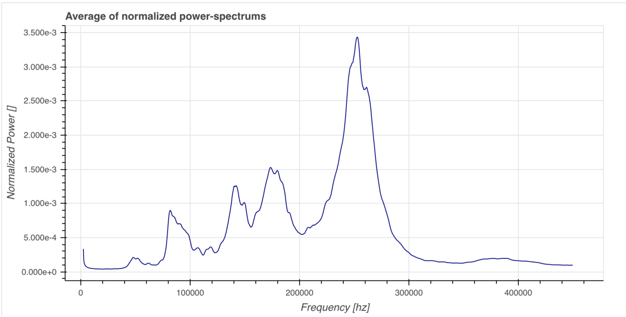

Figure 3: Power-spectra distribution.

The input is a chunk of 0.0375 seconds of seismic data (ordered in time), which is recorded at 4MHz, hence 150'000 data points, and the output is time remaining until the following lab earthquake, in seconds. 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, as integers. Both the training and the testing set come from the same experiment. The data is recorded in bins of 4096 samples. Withing those bins seismic data is recorded at 4MHz, but there is a 12 microseconds gap between each bin, an artifact of the recording device. The distribution of time-to-failures is provided in Fig 2.

3.2 Preprocessing

We experimented with two approaches to pre-processing: manual noise filtering and an automatic denoising technique.

High-frequency Noise Filtering

The first approach we took to pre-processing involved plotting the distribution of power spectra of earthquake signal segments and manually isolating a range which contains high spectral density. From the plot in Fig 3, we see that the regions of high power density are localized in a range between Khz and 300 Khz.

R-PCA for automatic de-noising

Principal component analysis (PCA) [3] is an effective tool for random noise attenuation. It has been widely used in seismic data processing for the enhancement of the signal-to-noise ratio of seismic data. However, PCA lacks robustness to gross outliers. We adopt a robust PCA (RPCA) framework that can be utilized in the frequency-space domain to automatically filter erratic noise typical in seismic data. The method adopts a nuclear norm constraint that exploits the low rank property of the desired data while using an l_1 norm constraint to properly estimate erratic (sparse) noise. Our seismic data is natively represented as a temporally varying waveform. We apply Robust PCA on data transformed via Short-Time Fourier Transform (STFT) into it's frequency domain. It's representation is a matrix D. As mentioned before, a reasonable assumption is that natural data, regardless of it's extrinsic representation, exhibits intrinsic low-dimensional structure. The underlying assumption of Robust PCA is that the desired signal is low-rank and corrupted by noise. i.e. Robust PCA suggests the following decomposition:

$$D = L + S \tag{1}$$

Where L is the low-rank signal we wish to recover, and S corresponds to sparse, additive noise. The problem of separating L and S can be formulated as an optimization problem:

$$\min_{L,S} rank(L) + \lambda ||S||_0 \text{ s.t. } L + S = D \tag{2}$$

where $||S||_0$ denotes the number of nonzero entries of S and λ is a balancing parameter. Due to the combinatorial nature of both terms in the objective, this problem is NP-Hard. A a relaxed version which substitutes convex surrogates for both terms is typically solved instead:

$$\min_{L,S} ||L||_* + \lambda ||S||_1 \text{ s.t. } L + S = D \tag{3}$$

where $||L||_*$ corresponds to the sum of the absolute values of the eigenvalues of L and $||S||_1$ corresponds to the sum of the absolute values of the entries of S. To solve this problem, we adopt the method of Augmented Lagrange [11] relaxes the hard equality constraint and reformulates the problem to facilitate unconstrained optimization:

$$\min_{L,S,Y} ||L||_* + \lambda ||S||_1 + \frac{\mu}{2} ||D - L - S||_F^2 + \langle Y, D - L - S \rangle \tag{4}$$

For Lagrange multipliers $\mu \in \mathbb{R}, Y \in \mathbb{R}^{d \times d} \geq 0$ whose magnitudes influence the satisfaction of the equality constraint in (2).

3.3 Feature Selection

In this section we describe two approaches to feature selection we evaluated. Statistical features are selected by deriving various statistical metrics to represent the underlying signal. Acoustic features are is based on an audio-centric interpretation of seismic signals. Learning features encompasses a set of ambitious techniques to learn discriminative features exclusively from the raw data.

Statistical Features

As mentioned above, statistical features are selected by deriving various statistical metrics that represent different local and global aspects of the underlying signal. We compute sliding-window features for a variety of different window sizes to capture local and global descriptors of the earthquake signal. In total, we compute 180 features including moment statistics about the waveform distribution (mean, variance, skewness, kurtosis) as well as quantile information.

Acoustic Features

To model acoustic features of earthquake seismic signals, we utilize the widely adopted Mel-frequency cepstral coefficients (MFCC) [4] representation as implemented in Librosa [14]. MFCC-based representations are an effective and popular representation that has been used for a variety of speech and audio processing tasks [4]. We briefly summarize the model:

Spectrum-to-MFCC computation is composed of invertible pointwise operations and linear matrix operations that are pseudoinvertible in the least-squares sense. This leads to a straightforward reconstruction process: Let the MFCC sequence C be computed as

$$C = D\log(MS) \tag{5}$$

Where S is a pre-emphasized Short-time fourier transform (STFT) magnitude spectrogram, M is a mel-filterbank matrix, and D is a truncated discrete cosine transform matrix. The reconstruction of the magnitude spectrum is obtained simply by

$$\hat{S} = M^{+}\exp(D^{+}C) \tag{6}$$

Where A^{+} denotes the pseudoinverse of A. We used the first 20 components, and first downsampled earthquakes to 40kHz. Without performing this initial downsampling step, the useful information is distributed significantly more components.

Feature Learning

As mentioned before, when dealing with high-dimensional natural data, it is common and reasonable to assume that there is an intrinsic dimensionality, or an underlying & unobserved small number of relevant degrees of freedom. A variety of linear and nonlinear methods for discovering this intrinsic dimensionality are used in practice. We apply two algorithms: one based on Multi-Dimensional Scaling (PCA/MDS) and spectral analysis.

We first evaluated linear embeddings derived from a Multi-Dimensional Scaling Algorithm [?] which is nothing but an application of PCA to a $n \times n$ matrix of euclidean distances between points. projecting our data onto the top k eigenvectors resulting from this process yields k-dimensional embeddings. A visualization of this process is provided in Fig. 4.

A manifold structure formalizes this idea for nonlinear dimensionality reduction. A manifold can be summarized as a mathematical space that looks and behaves locally like a Euclidean space of some fixed dimension.

Spectral methods have emerged as popular set of techniques for nonlinear dimensionality reduction, and they can learn representations that facilitate clusters which do not form convex regions in the embedding space. We will briefly summarize the idea. Spectral embeddings are derived using eigenvectors of an affinity matrix A (nonnegative and symmetric) which is built to represent the distance between points. If D is the diagonal matrix whose (i,i)-th entry is the sum of the entries of row i in matrix A, the top k eigenvectors of the Discrete Graph-Laplacian,

$$L = D^{-1/2}AD^{-1/2} \tag{7}$$

can clustered (i.e. via K-means) to compute a discrete partitioning of the points. These eigenvectors can be interpreted as a reduced dimension representation of the original samples and encodes the connectivity of the network. Intuitively, points whose embeddings are close are connected close in the graph representation. A visualization of the first two eigenvectors is provided in Fig. 5.

3.4 Prediction Algorithms

We evaluated several different classes of predictors on our learned features. In this section, we will briefly summarize and discuss the relative performance of each algorithm we applied and address their advantages and disadvantages in the context of seismic modeling. A summary of the algorithms is given in Table 2, and

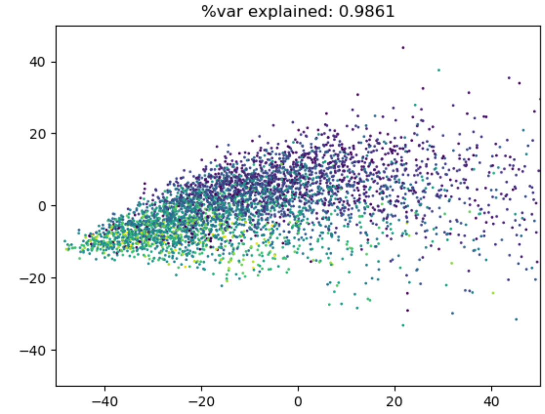

Figure 4: MDS.

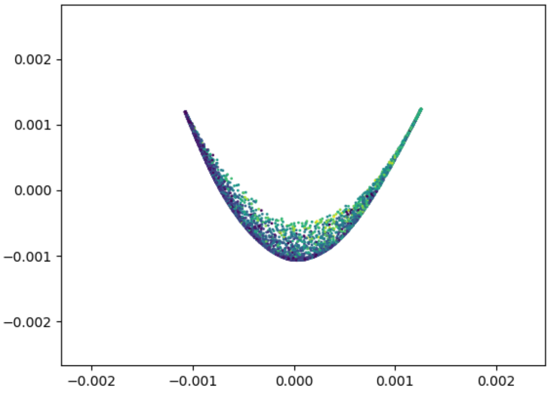

Figure 5: Spectral embedding of MFCC vectors from seismic activity (non-linear reduction from 48 to 3 dimensions). Color of points shows the TTF structure of the projection (higher TTF in one side in opposition to lower TTF).

reports for 5-fold cross validation experiments are given in Table 1. SVR gives the best performance, while the sparse linear algorithms provide good performance and efficiency with respect to normalized sparsity. We characterize the algorithms we applied into three groups:

3.4.1 Offline Algorithms

We do offline prediction using two algorithms: Linear Least Squares Estimation (LLSE) and, XGBoost (Gradient Boosting). We adopt LLSE as a baseline - any algorithm we apply should do better. We set XGBoost to minimize the least squares error, defined to be $\frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$ where $\hat{Y}_i$ is a prediction on the i-th example. XGBoost is an implementation of the Gradient Boosting algorithm. Gradient Boosting constructs an ensemble - or weighted averaging of a set of decision trees by iteratively adding new trees - or updating old trees - to the ensemble such that the new ensemble is guaranteed to reduce the loss. More formally, assuming the set of decision trees is differentiable, we update our mode according to the following step on the m-th iteration:

- $F_m(x) = F_{m-1}(x) - \gamma_m \sum_{i=1}^{n} \nabla_{F_{m-1}} L(y_i, F_{m-1}(x_i)$
- $\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} \nabla_{F_{m-1}} L(y_i, F_{m-1}(x) - \gamma_m F_{m-1}(x_i))$

Since we restrict the set of decision trees to be finite, a decision tree is chosen at each iteration which represents the projection of the gradient into this set. Fast implementations exist for both algorithms, and we were able to apply them our dataset.

3.4.2 Online Algorithms

We also adopted a set of online algorithms to evaluate. These algorithms are desirable because they can learn in a streaming fashion out of core - i.e. they do not require the entire dataset be stored in memory and they only need a single pass through the dataset to do learning. We used the Vowpal Wabbit library implementations of these algorithms, and implemented the Online-Lasso algorithm independently with Fast Iterative Shrinkage Thresholding (FISTA) [21]. To summarize, quantile regression is a robust mean estimator. We had hoped that this kind of algorithm would be more robust to noisy seismic data. The Lasso formulation is identical to LLSE, but includes a regularization term that induces sparsity in the model. By inducing sparsity, our model learns to select a small number of predictive features. We observe for both algorithms the selected features are identical - 4 MFCC features and 2 statistical features. We applied Gaussian Kernel-SVR which learns the unique line that minimizes the deviation from all points to a region around the line.

3.4.3 KNN-type Algorithms

Finally, we wanted to evaluate our learned features. We do this by applying the K-Nearest-Neighbors-type algorithm for regression to our learned features. K-NNR learns a weighted combination of nearby points, where the weights are proportional to euclidean distance.

3.5 Deep Learning

3.5.1 Recurrent Models

We also approached this problem, by using deep learning methods that have shown the potential to learn a strong feature space. Our first approach was to use a Recurrent Neural Network that takes in the sequential data and outputs predictions. Since, the dataset is huge, with all sequences of length 150,000, training recurrent models was not feasible. It is also shown that such long sequences would arise to the problem of vanishing gradient, which hinders the training of RNNs. In order to avoid that, we split each sequence into smaller sequences of length 1000 each. For example, a single sequence of length 150000 would have

Alg\Metric	MAE (mean, var)	Normalized S ($s/\|W\|$)	Runtime (s)	Training-time
LSE	[2.3, 0.22]	194/200 = 0.97	0.10	0.0016
XGBoost	[2.21, 0.216]	n/a	0.004	0.59
Online-SQR	[2.29, 0.02]	**6/200 = 0.03**	0.00026	n/a
Online-SVR	**[2.1, 0.1]**	2110/4194 = 0.5	0.000231	n/a
Online-Lasso	[2.26, 0.11]	**6/200 = 0.03**	**0.000209**	n/a
PCA-KNN	[2.3, 0.16]	n/a	0.000359	n/a
MFCC-Spec-KNN	[2.18,0.352]	n/a	0.000356	n/a

Table 1: Numerical results of our predictors. Bolded entries represent comparatively superior results. PCA-KNN and MFCC-Spec-KNN are restricted to learned features, while the rest of the algorithms operate on staistical + audio features.

Alg\Summary	Model-type	Model	Minimizer	Algorithm
LSE	nonlinear	$\frac{1}{N}\|\|\hat{Y}-Y\|\|_2^2$	Expectation	XGboost/GD
Quantile Regression	linear	Q-Loss	Median deviation	LP/GD/OGD
Lasso	linear	$\frac{1}{N}\|\|\hat{Y}-Y\|\|_2^2+\lambda\|\|\beta\|\|_1$	Regularized Expectation	FISTA/O-FISTA
Kernel-SVR	nonlinear	$\max(0, 1-Y\cdot\hat{Y})$	0-1 Approximation	SMO/OGD
Robust-PCA	Factorization	$\min_{L+S=D} rank(L)+\|\|S\|\|_0$	Factorization	Aug. Lagrangian

Table 2: Review of objective, loss, and optimization algorithms. Q-Loss = $\tau(y-p)\mathbb{I}(y \geq p)$.

150 smaller sequences of length 1000. For each of the smaller sequence, we replace the sequence by four features: mean of the sequence, minimum, maximum and the standard deviation. By doing this processing, our 150000 length sequence is now transformed to 150 × 4. This allows us to use a recurrent model, where at every timestep, a feature vector of length 4 is being given, for a total of 150 timesteps. This model gave us a MAE of 2.034

3.5.2 Convolutional Neural Nets

Another approach was to leverage Convolutional Networks to learn features from the sequential data. CNNs do not suffer from the vanishing gradient problem. We convert the sequences to the frequency domain and obtain spectrograms, which are then fed to the CNN architecture. We use 5 layers of Convolutions, along with pooling and batch normalization layers. We were able to achieve a MAE of 2.3. This was surprising, because we were hoping the CNN model would perform better than the recurrent model.

Method	MAE
RNN	2.034
CNN	2.3
Wavenet + LSTM	1.8

Table 3: Review of Deep Learning Approaches

3.5.3 WaveNet + LSTM

One of the main reasons for the previous models' low performance, was that those models were not able to extract a meaningful and useful latent space for making predictions. We ascribe this to the high dimensionality of the data. In order to extract meaningful features, we take inspirations from audio generation models. The intuition behind this is that, a stronger generative model should also be able to understand the structure of the data better. We use WaveNet [19] as our model and transform it from a generative model to a feature extraction model. WaveNet uses causal dilated convolutions to build a generative model for

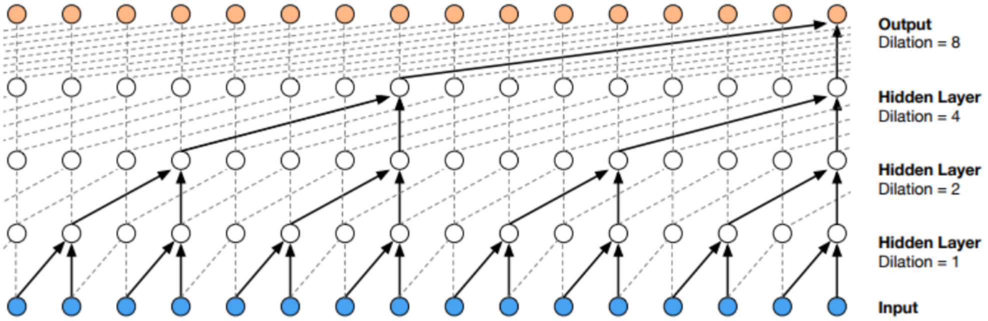

Figure 6: Architecture of WaveNet

audio. Figure 6 shows the architecture of Wavenet. As we can see, causal dilated convolutions can represent sequences (blue nodes in the bottom layer) by a single node (right most orange node in the top layer). Using this, we extract 150 nodes (for every 1000 nodes) for one sequence of length 150000. These features are then passed to a LSTM which makes the predictions. This method gave us the best results, with MAE of 1.8. Table 3 shows the consolidated results of the deep learning approaches.

4 Milestones

4.1 Milestone 1: Literature Review

Our first milestone was to survey existing works in predicting earthquakes. We performed a comprehensive review of the literature and compiled a short summary of the prior work. The literature survey gave us a lot of inspirations and ideas, that ultimately lead to our final results in conjunction with the audio-recommendation made by Professor Kastner around week 4.

4.2 Milestone 2: Exploratory Analysis

Exploratory Analysis was challenging because of the high dimensionality of the data. We used a lot of denoising techniques and downsampled the signal. Our initial approach to denoising was based on manually looking at several spectrograms derived from earthquake data. Noting that the majority of meaningful signal is concentrated at frequencies below 30k HZ we downsampled the signal to 90hz.

Motivated by prior work, we also evaluated an alternative approach based on robust PCA (`https://en.wikipedia.org/wiki/Robust_principal_component_analysis`): by observing the spectrogram of the signal, we note that the speckled noise exhibits a sparse structure. Through decomposition of the spectrogram (after thresholding as before) into a sparse part (corresponding to the speckled noise) and low rank part (the underlying signal), we can recover the true signal and then undo the DFT operation on the low rank component to recover the earthquake waveform.

4.3 Milestone 3: Feature Selection

We evaluate two methods for feature selection. Our first approach is based on manually computing rolling-window statistics (i.e. rolling mean, rolling variance, window-quantiles, etc.) for windows of sizes [10,50,100] and concatenating them. Furthermore, we concatenated spectral features (real + imaginary coefficients of the DFT transform) to our feature vector. In total, our representation of an earthquake is a vector of 180 numbers.

Our second method is based on automatic feature engineering. We compute a euclidean distance matrix between earthquake segments in our dataset and apply PCA. This technique is known as Multi Dimensional Scaling (MDS) and produces low dimensional representations that preserve distances. By observing a 2-d projection of these representations, we find clear cluster structure (point-intensity corresponds to time-to-failure), and we can use these representations directly as features (see our application of KNN below). Experimentation using other pairwise metrics is future work.

4.4 Milestone 4: Model Implementation

We evaluated several non-neural network-based methods for predicting runtime. In particular, we leveraged batch-regressors implemented in SK-Learns library [`https://scikit-learn.org/stable/`] on the 180 features derived during the previous milestone. The methods include an LSE baseline, XGBoost, Kernel-SVR, Random Forest, and Lasso. Pre-feature engineering, the data is approximately 10 gb. Even after feature engineering, the data remains on the order of gigabytes. We adopted several fast online algorithms from Vowpal Wabbit [`https://github.com/VowpalWabbit/vowpal_wabbit`] including Online Kernel-SVR, quantile regression, and online lse. In an attempt to improve upon algorithms utilizing our previous analysis, we also implemented an augmented online-lasso formulation from scratch in Python+Numpy. Our initial experiments validate the performance of this approach with respect to runtime and normalized sparsity. Due to online-lassos nature, no training time is required and the parameters are updated in a streaming fashion. In particular, we implemented augmentations to the standard Lasso formulation including a nonconvex SCAD penalty, a locally linear approximation algorithm to produce an optimal penalty multiplier, and an optimal algorithm (fast proximal gradient method (A Fast Iterative Shrinkage-Thresholding Algorithm - FISTA)). The implementation of this algorithm has been made available here: `https://github.com/choltz95/LASSO-SCAD`

We applied a Recurrent Network using the features mentioned above per timestep. We also worked on applying Convolutional Networks to the spectrogram resulting from application of a DFT. Using a distinct spectrogram for each earthquake in the training set, we applied a CNNs to model time-to-failure. All the implementations used Python, Tensorflow and Keras. Our best approach was using WaveNet to learn a strong feature representation and then use a LSTM. All our notebooks will be posted on our github (Please check link above)

4.5 Milestone 5: Evaluation Results

We carried out a variety of experiments - recording results based on 5-fold cross validation. We present the CV-mean and CV-variance of MAE for each algorithm as well as the mean normalized sparsity, runtime (to predict on a new sample), and training time for our offline algorithms. The reported KNN results is computed on MDS features, and we are surprised at how competitive it is.

5 Conclusions and Future work

We explore and evaluate audio based approaches to processing and machine learning and deep learning to analyze seismic patterns in order to predict the time remaining till the next earthquake. We believe that this would help save a lot of life and property if it can be incorporated in a real life scenario. We also hope that more research in this direction is pursued, as it looks really promising, based on our results.

Through our analysis, we have developed a number of ideas to pursue as future work. We have divided them into the following categories

Manifold Regularization

Inspired by the cluster structure we have observed from our PCA analysis, we hope to further explore the low dimensional structures of earthquakes. Additionally, we were surprised at the performance of KNN on learned feature representations, and this motivated us to further explore nonlinear dimensionality reduction algorithms for feature learning including MDS and spectral embeddings. In the future, we would like to more closely integrate these techniques with our predictive algorithms via spectral regularization.

Efficiency Analysis

Many of the techniques we have applied perform well for the task of earthquake time-prediction. However we would also like to explore the efficiency of algorithms for prediction. In many cases, earthquakes are sudden and early prediction of earthquakes increases the effectiveness of early response. It would be great if we could facilitate on-device prediction to eliminate time required for latency and data transfer. Since seismic sensors in the field are often power-efficient, we are motivated to explore methods and algorithms to reduce the energy and time required to do prediction. Although we have performed an additional analysis of the normalized sparsity and run-time of each algorithm, there are many interesting ideas worth pursuing.

More Deep Learning Architectures

We found that deep learning models were able to perform considerably better in comparison with classical machine learning models. With the rise of newer models like Attention etc, we definitely would love to experiment with such techniques, which are powerful sequential models.

Alternative Objectives

To successfully predict an earthquake and take evacuationary measures, it is important to predict not only the time, but also the place and the magnitude of the earthquake. Without one of these three main factors, evacuation would not be possible. It would be interesting to see if the current dataset has traits that can effectively predict the other two factors as well. It would also be interesting if we could induce learning more general representations - i.e. by including a term in the objective which corresponds to forecasting error.

References

[1] H. Akhondi-Asl and J. D. B. Nelson. M-estimate robust pca for seismic noise attenuation. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1853–1857, Sep. 2016.

[2] A. Bray and F. P. Schoenberg. Assessment of point process models for earthquake forecasting. *Statistical Science*, 28(4):510–520, Nov. 2013.

[3] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, June 2011.

[4] P. M. Chauhan and N. P. Desai. Mel frequency cepstral coefficients (mfcc) based speaker identification in noisy environment using wiener filter. In *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE)*, pages 1–5, March 2014.

[5] Y. Chen. Fast dictionary learning for noise attenuation of multidimensional seismic data. *Geophysical Journal International*, page ggw492, Jan. 2017.

[6] J. Cheng, K. Chen, and M. D. Sacchi. Application of robust principal component analysis (RPCA) to suppress erratic noise in seismic records. In *SEG Technical Program Expanded Abstracts 2015*. Society of Exploration Geophysicists, Aug. 2015.

[7] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[8] C. Hulbert, B. Rouet-Leduc, P. A. Johnson, C. X. Ren, J. Rivière, D. C. Bolton, and C. Marone. Similarity of fast and slow earthquakes illuminated by machine learning. *Nature Geoscience*, 12(1):69–74, Dec. 2018.

[9] J. R. Jr and F. G. Meyer. Machine learning for seismic signal processing: Phase classification on a manifold. In *2011 10th International Conference on Machine Learning and Applications and Workshops*. IEEE, Dec. 2011.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[11] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. 2010.

[12] O. Lindenbaum, Y. Bregman, N. Rabin, and A. Averbuch. Multiview kernels for low-dimensional modeling of seismic events. *IEEE Transactions on Geoscience and Remote Sensing*, 56(6):3300–3310, June 2018.

[13] N. Lubbers, D. C. Bolton, J. Mohd-Yusof, C. Marone, K. Barros, and P. A. Johnson. Earthquake catalog-based machine learning identification of laboratory fault states and the effects of magnitude of completeness. *Geophysical Research Letters*, 45(24):13,269–13,276, 2018.

[14] B. McFee, M. McVicar, S. Balke, C. Thomé, C. Raffel, D. Lee, O. Nieto, E. Battenberg, D. Ellis, R. Yamamoto, J. Moore, R. Bittner, K. Choi, P. Friesch, F.-R. Stöter, V. Lostanlen, S. Kumar, S. Waloschek, S. Kranzler, R. Naktinis, D. Repetto, C. F. Hawthorne, C. Carr, W. Pimenta, P. Viktorin, P. Brossier, J. ao Felipe Santos, J. Wu, E. Peterson, and A. Holovaty. librosa/librosa: 0.6.1, May 2018.

[15] Y. Ogata. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical Association*, 83(401):9–27, 1988.

[16] J. Ramirez. Learning from manifold-valued data: An application to seismic signal processing. 2012.

[17] Z. E. Ross, Y. Yue, M. Meier, E. Hauksson, and T. H. Heaton. Phaselink: A deep learning approach to seismic phase association. *CoRR*, abs/1809.02880, 2018.

[18] B. Rouet-Leduc, C. Hulbert, D. C. Bolton, C. X. Ren, J. Riviere, C. Marone, R. A. Guyer, and P. A. Johnson. Estimating fault friction from seismic signals in the laboratory. *Geophysical Research Letters*, 45(3):1321–1329, 2018.

[19] B. Rouet-Leduc, C. Hulbert, and P. A. Johnson. Continuous chatter of the cascadia subduction zone revealed by machine learning. *Nature Geoscience*, 12(1):75–79, Dec. 2018.

[20] B. Rouet-Leduc, C. Hulbert, N. Lubbers, K. Barros, C. J. Humphreys, and P. A. Johnson. Machine learning predicts laboratory earthquakes. *Geophysical Research Letters*, 44(18):9276–9282, 2017.

[21] S. Tao, D. Boley, and S. Zhang. Local Linear Convergence of ISTA and FISTA on the LASSO Problem. *arXiv e-prints*, page arXiv:1501.02888, Jan 2015.

[22] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. In *Arxiv*, 2016.

[23] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu. CNN-RNN: A unified framework for multi-label image classification. *CoRR*, abs/1604.04573, 2016.