Seperate Earthqukaes from Nuclear explosions

Sabber Ahamed

November 13, 2017

1 Project Overview

The Comprehensive Nuclear-Test-Ban Treaty (CTBT) is a multinational treaty aim to ban all nuclear activities including explosions, for both civilian and military purposes. More than 180 countries have agreed and signed the agreement. But, forty-four countries that have the existing nuclear capabilities must sign and approve the deal for it to have the force of law. Several recognized monitoring systems are used to control the potential violations of the treaty. Seismic wave observation is one of them and is considered the most reliable method for verifying an underground explosion. Seismic stations typically detect as many as 800,000 earthquakes worldwide. Compare to this number, explosions especially the number of nuclear explosions are very-low. Seismologists use several observations such as polarization, p-wave, S-wave and surface wave, depth of the events, etc. to distinguish explosions signal from earthquakes. All the process are complicated and requires the integration of physical and statistical techniques and human interpretations. Not always all the reviews are right. For example, recent quake from North Korea with magnitude 3.0 was first identified as nuclear explosions both by China and South Korea, but the detailed study of the seismic waves later confirmed that the quake was indeed a natural earthquake [6]. Machine learning approach has been adopted to minimizing error in discriminating mine seismic events from mine blasts [2]. In this project, my primary goal is to develop an improved machine learning model that can separate natural earthquake from nuclear explosions using raw seismogram.

2 Problem Statement

I plan to use supervised learning techniques toward the final solution. The primary goal is to develop a machine learning model that can distinguish natural earthquakes from nuclear explosions. Since there are only two types of data (earthquakes and nuclear explosions), the model would be binary classification problem. I plan to use support vector machine (SVM) because the preliminary studies show that SVM outperforms the other supervised algorithms such as random forest, Xgboost, LightGBM, etc. More details of SVM is described in algorithoms and techniques sections. The justification for using SVM is discussed

in benchmark section. Performance of the model will be evaluated by accuracy, recall, F-1 and ROC-AUC scores based on the predicted and actual values of the test dataset.

3 Metrics

In this project, I will use the following evaluation metrics:

1. **Confusion matrix:** The confusion matrix is a convenient presentation of a model performance have multiple classes. The matrix presents predictions on the x-axis and accuracy outcomes on the y-axis. The table helps us understanding the model performance showing positive and negative classes prediction status.

2. **Accuracy:** Accuracy measures the proportion of the number of correct predictions of the entire data sets. The metric is the most common and straightforward evaluation metric for classification problems. However, this parameter would not give us very detail information about the model performance and therefore, can be misused. As a result, I also use the other evaluation metrics listed below.

3. **precision:** The metric measures the proportion of the positive-class among the total number of positive predictions. Precision would help us to analyze how the model is doing in each category.

4. **Recall:** Recall on the other hand measures the proportion of the positive-class among the total number of positive-class. The recall is the ideal metric to judge the model performance on positive-class (Nuclear explosions).

5. **F-1 Score:** This metric combines merely precision and recall just taking the harmonic mean of them. F-score would be my auxiliary evaluation metric after precision and recall because it explicitly represents recall and precision under the hood.

6. **ROC-AUC score:** Area under ROC Curve (or AUC for short) is a performance metric that represents the model's ability to discriminate between positive and negative classes. An area of 1.0 represents a model that made all predictions perfectly. Since the project is a binary classification problem, this score would help to evaluate the model performance. I would also use this metric as secondary as F-1 Score.

4 Analysis

4.1 Data Exploration

I will use the facilities of Incorporated Research Institutions for Seismology (IRIS) Data Services, and specifically, the IRIS Data Management Center, to access waveforms, related metadata, and derived products used in this study. IRIS Data Services are funded through the Seismological Facilities for the Advancement of Geoscience and EarthScope (SAGE) Proposal of the National Science Foundation under Cooperative Agreement EAR-1261681. Table-1 shows the events I use for this study.

Table 1: Seismic events used for this study

Event Origin	Date	Magnitude	Seismograms	Event type
Banda Sea	Oct. 24, 2017	6.7	3382	Earthquake
Sumatra, Indonesia	Dec. 26, 2004	9.0	1193	Earthquake
Myanmar India border	Mar. 12, 2010	5.5	3021	Earthquake
Ryukyu islands	Aug. 15, 1017	4.9	2644	Earthquake
Chiapas Mexico	Sep. 08, 2017	8.1	2722	Earthquake
Total			12962	
India	May. 11, 1998	5.2	459	Nuclear explosions
Pakistan	May. 28, 1998	4.8	470	Nuclear explosions
Pakistan	May. 30, 1998	4.6	398	Nuclear explosions
North Korea	Feb. 12, 2013	5.1	3763	Nuclear explosions
North Korea	Jan. 06, 2016	5.1	2640	Nuclear explosions
North Korea	Sep. 03, 2017	6.3	2804	Nuclear explosions
Total			10534	

Table 2: Features for the model

Feature Name	Description	Features
MFCCs	Mel-frequency cepstral coefficients.	40
Chorma-stft	Chromagram representing the 12 distinct chromas.	12
Amplitude	Maximum and mean amplitude of a waveform	2
Spectral centroid	Measures the 'center of mass' of a waveform.	1
Statistical parameters	Moment, variation, skew, variance, auto-corelation, kurtosis.	6
Total Number of features		61

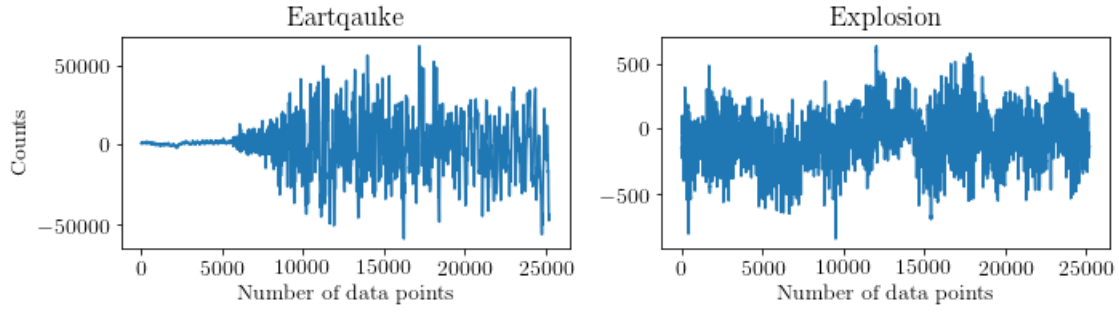Figure 1: The illustration shows the raw east-west componet of earthquake and explosions seismogram recorded at station 'MLAC'. The earthquake occured in Sumatra, Indonesia in 20014. The right pannel shows the seismogram due to underground nuclear explosion in India Pakistan border in 1998.

Each seismogram is 21 minutes long and starts 1 minute before P-wave arrival and ends 20 minutes after P-wave arrival. Fig. 1 shows the raw east-west component of seismogram of an earthquake and a nuclear explosion. Characteristics signal features that will be extracted using `Librosa` [3]- a python package for music and audio analysis. To improve the performance of the model, the number of features may vary in the final solution. I use 20% f the entire data for testing the model performance.

4.2 Exploratory Vizualization

Fig. 2 and 3 shows the stacked seismograms of earthquakes and explosions. Stacked earthquake seismograms show an exciting pattern. For example around 3000 data points S-wave arrival is distinctive followed by surface waves after 4000 data points in east-west and north-south component. P-wave arrival time, on the other hand, is characteristic in vertical-component. The explosions seismograms do not show any interesting pattern like earthquake seismograms. The pattern shown by the natural earthquake seismograms would be the key for machine learning models.
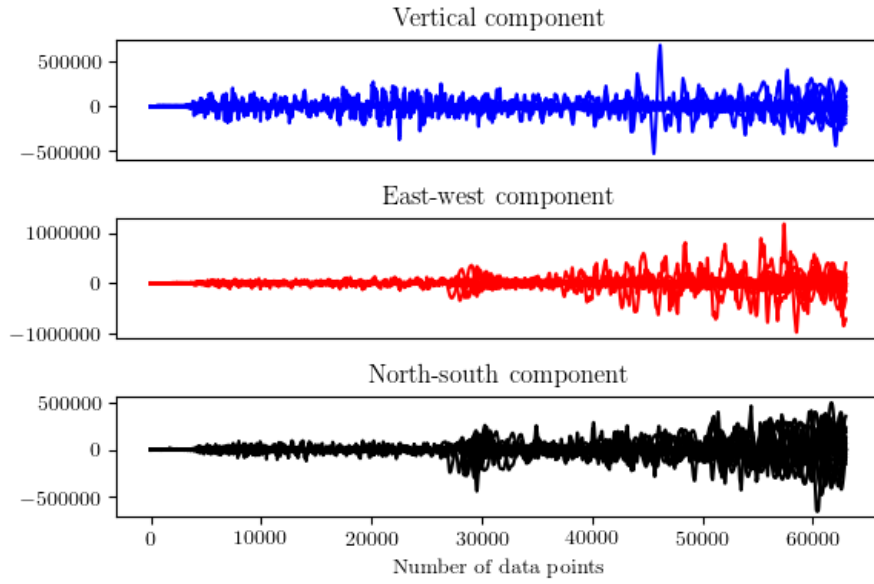
Figure 2: The illustration shows the 200 stacked seismograms of different earthquakes. Since the seismograms are raw and unprocessed, x and y label do not show usual label like velocity and time respectively. The illustration shows an interesting pattern among the seismograms. For example around 3000 data points S-wave arrival is distinctive followed by surface waves after 4000 data points in east-west and north-south component. On the other hand, P-wave arrival is distinctive in vertical component.
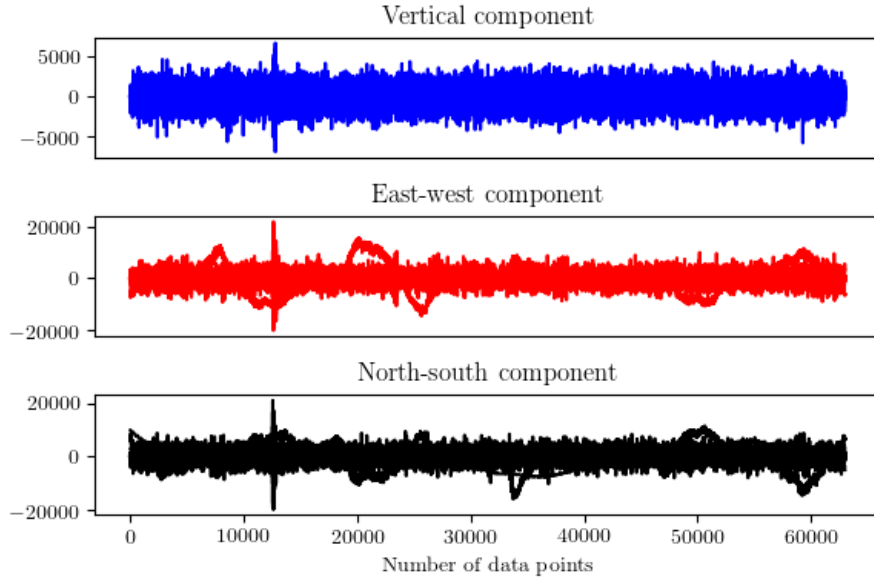

Figure 3: The illustration shows the 200 stacked seismograms of different underground nuclear explosions. Sicne seismograms are raw and unprocessed x and y label do not show usual label like velocity and time. The seismograms do not show any interesting pattern like earthquake seismograms.

4.3 Algorithoms and Techniques

4.3.1 Algorithom

In this project, I use support vector machine (SVM) algorithm [1] to discriminate natural earthquakes from nuclear explosions. The algorithm uses kernel trick to transform the data and based on these transformations; it finds an optimal boundary (line or hyperplane) between the possible outputs. An SVM model is a representation of the data points in space, creating a line or hyperplane so that the examples are divided by reasonably. SVM then maps new instances into that same space and predicts a class based on which side of the hyperplane they fall. Vladimir N. Vapnik and Alexey Ya. Chervonenkis in 1963 first present the idea of SVM algorithm. Since then, SVMs have been used successfully in different real-world problems such as text categorization, image classification, protein classification, Cancer analysis, handwritten character recognition, etc.

4.3.2 Tuning parameters

In this project I use most widely used the library for implementing machine learning algorithms in python is `scikit-learns` [4]. The following parameters are the significant parameters need to be tuned to get best model performance :

1. C: It is the regularization parameter, C, of the error term.

2. kernel: It specifies the kernel type of the algorithm. It can be 'linear,' 'poly,' 'rbf,' 'sigmoid,' 'precomputed.

3. Gamma: Kernel coefficient for 'rbf,' 'poly,' and 'sigmoid.' If gamma is 'auto,' then 1/n-features will be used instead.

To find the best parameters I will perform an exhaustive search over specified parameter values using `scikit-sklearn` implemented `GridSearchCV`.

4.4 Benchmark

Table-3 listed five supervised algorithms used to create the benchmark. In these benchmarks, I used sklearn [4] default parameters. It is pretty obvious that the support vector machine (SVM) performs better than all other algorithms. Therefore, I use SVM as the final model while its scores set the benchmark for further improvement.

Table 3: Model performance in five algorithoms

Algorithom	Accuracy	ROC-AUC	Avg. F1-score	Avg. recall
Random Forest	0.64	0.64	0.64	0.65
Xgboost	0.61	0.60	0.61	0.61
LightGBM	0.48	0.41	0.42	0.48
Gaussian Naive Bayes	0.50	0.63	0.41	0.50
Support Vector machine	0.77	0.76	0.78	0.78

5 Methodology

5.1 Data processing

In this project data processing is two-step processing (1) First removing bad seismograms and then apply z-score in each seismogram and (2) Process features extracted from seismogram. The following subsection has the details of these two steps.

5.1.1 Seismogram processing

One of the primary objectives of this project is to use raw seismograms to see if the unprocessed seismogram can providing us the information we want. Processing techniques vary person to person and purpose to purpose. In this project, I standardize the seismograms removing the mean from each seismogram so that they are centered around and zero. Other than the standardization I did not use any additional processing necessary for a seismic signal such as instrument corrections and filtering.

5.1.2 Extracted feature processing

Standardization of a dataset is a standard requirement for many machine learning estimators: they might misbehave if the individual feature does not more or less look like standard normally distributed data. After I extract all the features from seismograms, I standardize the features by subtracting the mean and divide by standard deviation.

5.2 Implementation

The implementation process is split into two main steps:

1. Training and

2. Measuring the performance

```python
from sklearn import svm

svm_model = svm.SVC(C=4, cache_size=200, coef0=0.0, class_weight='balanced',
    decision_function_shape=None, degree=3, gamma=0.08, kernel='rbf',
    max_iter=-1, probability=False, random_state = random_state, shrinking=True,
    tol=0.001, verbose=True)

svm_model.fit(X_train, Y_train)

Y_pred = svm_model.predict(X_test)

print('1. The accuracy of the model is {}\n'.format(accuracy_score(Y_test, Y_pred)))
print('2. Classification report \n {} \n'.format(classification_report(Y_test, Y_pred)))
print('3. Confusion matrix \n {} \n'.format(confusion_matrix(Y_pred, Y_test)))
print('4. Roc_Auc score \n {}'.format(roc_auc_score(Y_pred, Y_test)))
```

Figure 4: The illustration shows a portion of code snippet implemented to create the model using SVM. The first part of the snippet is importing SVM module from `scikit-sklearn`. SVM model is created using the parameters obtained by another interesting `scikit-sklearn` implemented module `GridSearchCV`. Finally I print out the model performance using evaluation metrics.

4000 preprocessed training datasets were used to train the model. I use Jupyter-notebook for its interactive and friendly environment. In this project, I use `scikit-sklearn` implemented API for `LIBSVM` - an integrated software for support vector classification [5]. Each time I get results from the model trained with new sets of parameters I use the evaluation metrics to measure the model's performance. Fig. 4 shows a code snippet of the SVM model.

5.3 Refinement

After I train and build the model, I improved the model accuracy while tuning parameters feature engineering, seismogram processing. The followings are the steps I followed to refine the model performance :

1. **Step 0 Benchmark:** At this initial step I used five of supervised algorithms to find out the best one that gives the highest model performance. I have found Support vector machine is the one provides best model performances regarding accuracy, Recall, F-1, and ROC score. The total number of features is 61 while the accuracy is 77%.

2. **Step 1 Grid search:** After I select SVM as the final algorithm, I use `scikit-sklearn` `GridSearchCV` module to find best C and gamma parameters. After an exhaustive

search for many hours, the best C and gamma are 4.0 and 0.08. The total number of features is 61 while the accuracy is 84.5%.

3. **Step 2 Feature manipulation:** At this stage, I manipulate some features. I have tried adding some extra features like amplitude of the signal, Mel coefficients, etc. But the did not improve while removing the feature *moment* increases the accuracy and recall little bit and becomes 85.8% and 82% respectively.

4. **Step 3 Feature manipulation:** At this stage, I add ten more features which improve the model accuracy and positive recall to 87.85% and 86% respectively.

6 Results

6.1 Model evaluation and validation

Table 4: Confusion matrix based on 4692 testing data

	Negative	Positive
Natural Earthquakes	True Negative(TN) = 2317	False negative(FN) = 301
Nuclear Explosions	False positive(FP) = 269	True Positive(TP) = 1805

I use 4692 testing data to estimate generalization error of the model created by support vector machine. The table 4 shows the confusion matrix contains information about actual and predicted classifications. In this project the positive class is *nuclear explosion* and *natural earthquake* is the negative class. The SVM classifier accurately predicts 2322 as earthquakes out of 2594 earthquakes datapoints and 1760 datasets as nuclear explosions out of 2098 explosions data points while rest of the data are misclassified.

Table 5: Classification results based on 4692 testing data

Class	Precision	Recall	F-1 score	support
Natural Earthquke	0.89	0.90	0.89	2586
Nuclear explosion	0.87	0.86	0.86	2106
Average/Total	0.88	0.88	0.88	4692

Table 5 shows the classification report using three evaluation metrics. The overall testing accuracy of the model is about 88%. Accuracy measures the proportion of correctly classified classes out of total test datasets. In addition to accuracy, we also consider precision, recall, and f-1 scores. Precision measures how many datasets are correctly classified as

true positive among the true and false positive combined. Whereas, the recall measures the proportion of the correctly identified true positive instances among the total positive datasets. The F-1 score is the harmonic mean of recall and precision.

6.2 Justification

The final ROC-AUC score of the model is 87.76% which is 10.22% more than that of the benchmark. To judge the model performance on positive-class (nuclear explosions), we need to look at the misclassification. In the benchmark, the model misclassified 30.11% atomic explosions while the final results misclassified 16.67%. The model also successfully identifies the recent nuclear explosion in North Korea (Fig. 5), which is a great performance. In summary, the current model is helpful to discriminate the signals recorded only in the USA, however, to solve the bigger problem we need more data hence powerful hardware with multiple CPUs or GPUs.

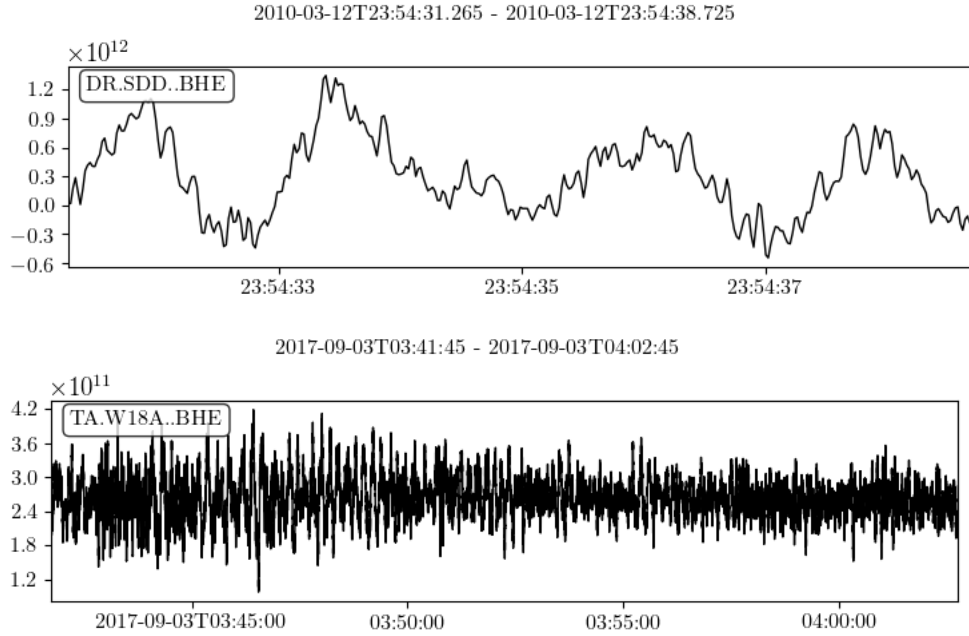# 7 Conclusion

7.1 Free-Form Vizualization

Figure 5: The illustration shows a correctly predicted earthquake and nuclear explosions. The earthquake occurred in Myanmar India border in 2004 and recorded in station 'SDD,' USA. The underground explosion occurred recently in 2017 in North Korea and recorded by 'W18A' station of TA network in the USA.

7.2 Reflection

The processed used for this project can be listed as follows :

1. Select earthquake and nuclear explosions events.

2. A request has been submitted to IRIS to get the data.

3. Processes the seismograms and remove bad and noisy ones.

4. Extract features from each seismogram

5. Create a benchmark for the classifier and find the best algorithm.

6. The classifier was trained using trained data.

7. `GridSearchCV` was used to find the best parameters to improve the model performance.

8. Manipulate the features to increase the model performance.

I found step four is the most difficult since I had to familiarize the sound signal processing library `Librosa` and several techniques. Another problematic but exciting part of the project was to manipulate the features (step eight). I'm also happy and grateful for getting to use open source libraries like `scikit-sklearn`, `Librosa`, `Jupyter-notebook`, `numpy`, `scipy` as I believe open-sourcing scientific works helps our researchers to discover essential things for humankind.

7.3   Improvement

Distinctive and useful features are required to achieve the optimal model performance. Using raw seismograms was one of the most challenging. If I had used processed and three components seismogram together to determine features, the model performance would be improved definitely. However, my motivation was to use raw seismograms to see if they are capable of discriminating between earthquakes and nuclear explosions. I believe feature selections are the key to improve the model performance.

References

[1] Corinna Cortes and Vladimir Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.

[2] Longjun Dong, Johan Wesseloo, Yves Potvin, and Xibing Li. Discrimination of mine seismic events and blasts using the fisher classifier, naive bayesian classifier and logistic regression. *Rock Mechanics and Rock Engineering*, 49(1):183–211, 2016.

[3] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, pages 18–25, 2015.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[5] Bernhard Schölkopf, Alex J Smola, Robert C Williamson, and Peter L Bartlett. New support vector algorithms. *Neural computation*, 12(5):1207–1245, 2000.

[6] The Telegraph. Earthquake in north korea was natural, south korean agency says. *The Telegraph*, 2017.