STATS 841: Final Project Report
Melbourne University Seizure Prediction

Ruifan Yu & Shivam Kalra
{ruifan.yu,shivam.kalra}@uwaterloo.ca
Team: *Overfit*

December 2, 2016

Abstract

We are team `Overfit` and worked on Melbourne University AES MathWorks NIH Seizure Prediction. For the competition, we are building a model to classify EEG signal into "Interictal" or "Preictal". Based on literature review, we are extracting features by FFT, Butterworth filters and correlation. We are using 3 models to classify EEG. Our first model is window based with SVM, Gradient Boost and Random Forest as classifiers. Second is spectrogram based with Convolutional Neural Network (CNN) as classifier. Third is based on Radon projections with SVM as classifier. Per Kaggle results, our first model outperformed other two. We've scored **0.73757** ranking **69th** in the public leader-board and **0.72461** ranking **65th** in the private leader-board.

1 Introduction

Seizure prediction is popular field of research, enabled by statistical analysis methods applied to features derived from intracranial Electroencephalographic (EEG) recordings of brain activity [5]. Seizure forecasting systems have the potential to help patients with epilepsy to lead more normal lives. For that reason, the Kaggle competition aims at developing a stable and accurate seizure classifier. In this competition, we are provided with labeled EEG recordings from 3 patients to train our classification models. We further use our trained models to predict on the unlabeled testing data, which is submitted to Kaggle for evaluation.

1.1 Data Description

Data for the competition consists of multiple EEG recordings from three patients. Each recording (known as **clip**) is 10 minutes long, consisting of 16 data channels from 16 different electrodes sampled at 400 Hz. Each clip in training data is categorized as Interictal or Preictal. Number of clips in the training and testing data for each patient is shown in Table 1. Kaggle uses Area under ROC curve (AUC) as method of evaluation.

	Patient 1	**Patient 2**	**Patient 3**
Train	1302	2346	2394
Test	216	1002	690

Table 1: Number of data clips in train and test data-sets for each patient.

1.2 Problems in Data-set

By analyzing the data-set, we found two major issues that influences the training of our models.

1. **Categorical Imbalance:** Number of positive (preictal) and negative (interictal) samples are highly unbalanced. It decreases the recall rate and results in high false negative rate when proper regularization/preventive measures are not applied.
2. **Random Dropouts:** Some portion of clips have random dropouts (all zeros). In some rare cases, clips are entirely empty. This brings noise to our classifiers and further influences their accuracy.

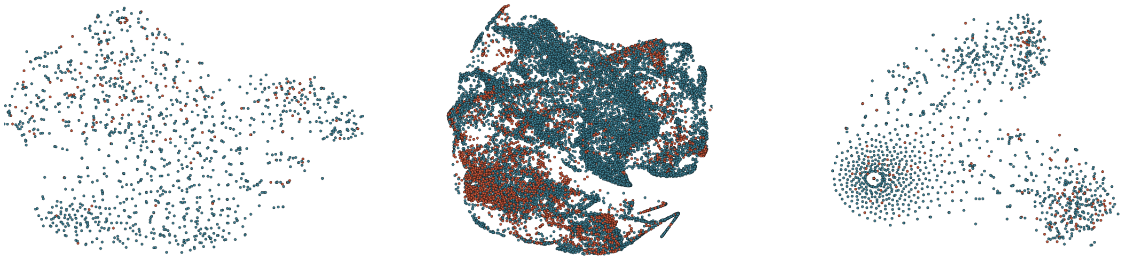
2 Feature extraction

All our feature extractions rely on 6 useful frequency intervals as mentioned in [2]: Delta (0.1 – 4 Hz), Theta (4 – 8 Hz), Alpha (8 – 12 Hz), Beta (12 – 30 Hz), Lowgamma (30 – 70 Hz) and Highgamma (70 – 180 Hz). We are using four kinds of features to train our models as summarized in Table 2.

Domain	**Feature**	**Used By**
Frequency Domain	Mean & Variance of magnitudes in six frequency bands	Model 1
Time Domain	Mean power ($amplitude^2$) for six frequency bands	Model 1
Time Domain	Pairwise correlations among 16 channels	Model 1
Frequency Domain	Spectrograms (binned in frequency and time domain)	Model 2 & Model 3

Table 2: Different features extracted in different domains for training models.

We apply Fast Fourier Transform to channel's data for collecting the magnitude information in all six frequency bands. We also use Butterworth band pass filter to extract average power ($amplitude^2$) in the time domain for all six frequency bands. Then we extract pairwise correlation channels in time domain, as in [4]. All these features are extracted from a *window* (see Def 2.1). Spectrograms are non-windowed, however we still apply binning in time and frequency domain to make them more useful (used by Model 2 & Model 3).



(a) Raw channel data (b) Window features (c) Radon features

Figure 1: TSNE visualization of features (blue is interictal and red is preictal) from (a) raw channel data, (b) features from windows based model and (c) radon features

Definition 2.1. *Window: A non-overlapping sub-sample from the clip in all 16 channels.*

We tried different window size such as: 5, 10, 20 and 60 seconds for the training purposes. Window enables down-sampling of the data and filtration of the noise & dropout areas.

TSNE is a technique to visualize the manifolds in higher dimensional data in low dimension (for example 2D or 3D). Figure 1 shows the separability of different features extracted for different models. We can see that features for window based model are better distributed for the classification task which is evident from the results discussed in later sections.

3 Models

Since seizure activity is highly individual, we train one model per patient. Following section explains details of three of our models.

3.1 Window Based Model

Motivation

SVM, Gradient Boost and Random Forest are well suitable for binary classification with hand-crafted features. Therefore, we used them as the classifiers in our first model which is based on features extracted from *window*. We call this model as **Window based model**.

Design

As shown in Figure 2, window based model consists of three parts: 1) Feature extraction, 2) Training and 3) Prediction using aggregation. In the feature extraction step, we extract the features according to the Table 2. For collecting data to train classifiers, we randomly select 20 windows per **clip** and extract feature of size 408 per **window**.

Then, we feed our classifiers with all the training data. By our design, instead of directly predicting the final probability, this model will first predict the probability for each window then generate a final probability by aggregating the predictions for all windows (refer to Figure 2). Ideally, this may bring more robustness to our model.

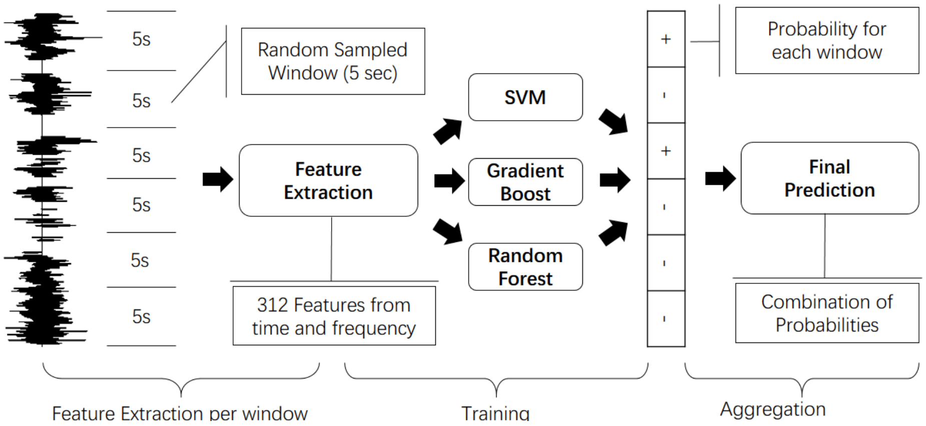

5s
5s
5s
5s
5s
5s
Random Sampled Window (5 sec)
Feature Extraction
312 Features from time and frequency
SVM
Gradient Boost
Random Forest
+
'
+
'
'
'
Probability for each window
Final Prediction
Combination of Probabilities
Feature Extraction per window
Training
Aggregation

Figure 2: Window Based model.

Analysis

Through experiments, we found that unbalanced data lead to bad recall rate. Although the general AUC score was good in cross-validation set, we were not doing well in detecting “Preictal” cases. Besides, since we did not know the ratio in test data, we should focus more on the sensitivity. Therefore, we did some tuning works on parameters to fix these problems.

Data is unbalanced, therefore, we manually set a bigger weight to the positive samples penalizing misclassified positive observations, henceforth brings the better results.

As for the AUC score, we found an interesting issue that we could not get a accurate estimation of our model. Even though the model performed so well in cross-validation set, the Kaggle score was much lower. After some analysis, we realized that this was caused by shuffling and splitting the train–test data-set.

Some 10 min clips may come from same hour segment and they are very similar. Therefore, the window features extracted from these clips could be similar too. After we created the cross-validation set, there were some observations that were identical to some others in training set. It is a kind of data leakage. As the data is time-series, it needs some tricks to set up the suitable test set. After being aware of that, we manually selected test data set so that there was no overlapping between the train–test data sets.

Over-fitting is very common for this kind of unbalanced data-set. In order to reduce the influence of over-fitting, we added many methods to punish the complexity of our model. In SVM, we limited the flexibility of decision boundary, giving us less penalty to wrong predictions. For Gradient Boost, we limited the max depth of decision tree, the min child weight and running iterations. Besides, during training, instead of using full train set, we randomly sub-sampled the data-set to introduce more robustness to our model.

3.2 Spectrogram based model

Motivation

Convolutional Neural Network (CNN) has been used in seizure detection in various studies, for ex. [3, 4]. These studies have shown that CNN is a very competitive method for seizure prediction, meanwhile CNN is state-of-art in computer vision domain nowadays.

Design

We generate spectrograms for each channel with 60 time bins and 6 frequency bins. 6 frequency bins are selected as mentioned earlier in Section 2. 60 time bins are equal sized windows of 10 seconds each. Since there are 16 channels, we get (16 x 6 x 60) image for each clip (first dimension is channel depth and (6 x 60) is resolution of image).

With spectrogram, the prediction is turned into image classification problem. The architecture of CNN that results in best AUC score is shown in Figure 3.

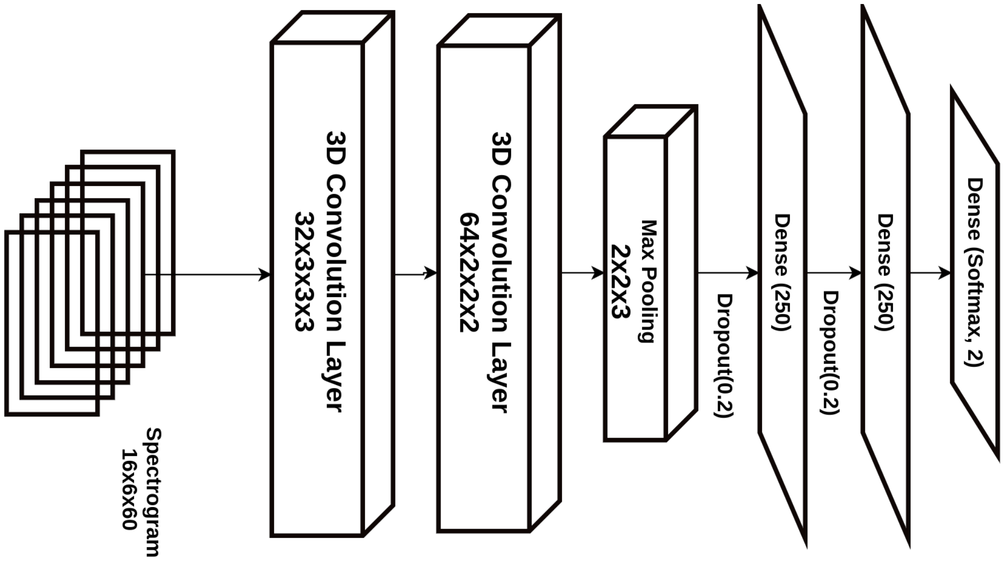


Figure 3: Architecture of CNN using 3D convolution kernels.

Analysis

Problem of unbalanced data and random dropouts is solved using data augmentation. Data augmentation enables us to create extra data during training phase. We regularize our CNN predictor by adding the dropout layers [6], which prevent the over-fitting.

3.3 Radon projections based model

Definition 3.1. *Radon Transform adds up the pixel values in the given image along a straight line in a particular direction and at a specific displacement.*

Motivation

In existing literature, Radon transform has been used on spectrograms to extract the effective acoustic features from the speech data [1]. It is vastly studied and reputed feature extraction technique in computational medicine. It forms basis of CT scans, tomography and used in compression of medical images.

Design

For every clip, spectrograms (16x10x60) are stacked in vertical direction to create a single spectrogram (160x60) called **stitched spectrogram**. Stitched spectrogram is used to calculate the Radon projections from 16 equidistant angles (between 0° – 180°), giving 2D vector of 16x171 as a feature. These 2D features are flattened into 1D vector (size of 2763) which are subsequently used to train SVM classifier.

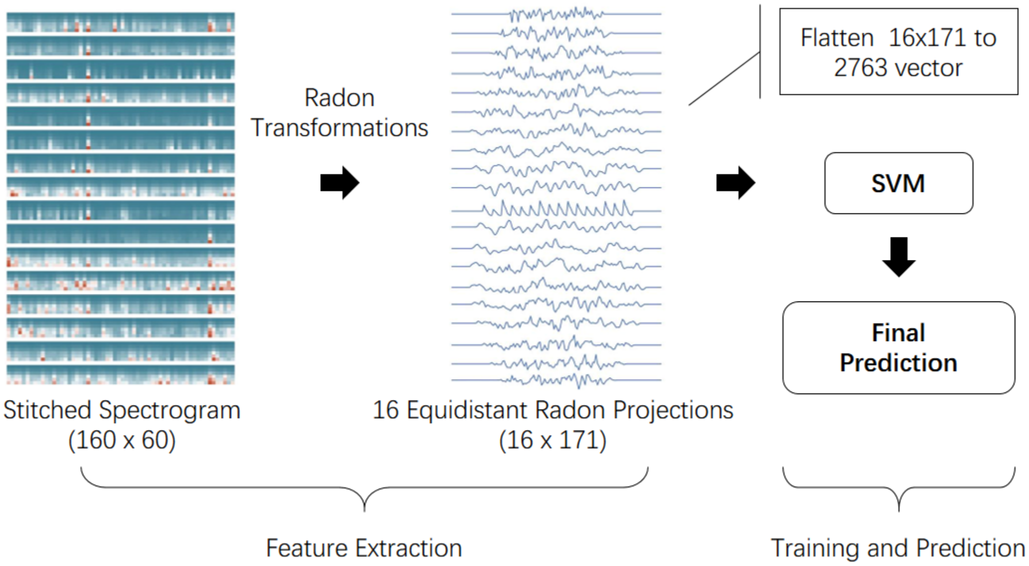

Radon Transformations
Flatten 16x171 to 2763 vector
SVM
Final Prediction
Stitched Spectrogram (160 x 60)
16 Equidistant Radon Projections (16 x 171)
Feature Extraction
Training and Prediction

Figure 4: Overview of the Radon Projections based model.

Analysis

Keeping unbalanced labels in mind, we trained SVM with more emphasis on positive cases (preictal). We trained using 5-fold cross validation to prevent wrongful learning due to data dropouts. Other problems and their solutions during the training of SVM have been discussed in previous Section 3.1.

4 Results

Table 3 shows our Kaggle scores for submissions using different models. We are not surprised to see that Model 1 out-performed from all models. For the spectrogram based model, we suffered over-fitting. It also shows that Radon transformation based model is not suitable for this data-set. We tried some modifications not no improvement in Radon model.

We selected average ensemble and weighted ensemble as our two submissions for the final evaluation.

Remark. *Our team **Overfit** scored **0.67062** ranked at* 158^{th}. *However, our last submission was made using account **RuifanYu** scoring **0.72461** ranking* 65^{th}. *Since we have access to all our final private leader-board scores, our best entry scored **0.74115** by Model 1 (without ensemble).*

Model	Public Leader-board	Private Leader-board
Model 1	0.69443	0.74115
Model 2	0.71466	0.64398
Model 3	0.65910	0.65460
Average Ensemble	0.73757	0.67062
Ensemble (Weighted)	0.70371	0.72461

Table 3: Kaggle scores from all three models (including two ensembles)

5 Conclusions

During this project, we learned a lot in application of machine learning and how handle parameter tuning and large data-sets. In real world scenarios, the feature engineering is the most important aspect. From this project, we also realize that deep learning is not very suitable for every kind of problem. So far, tree-based classifiers and SVM are still the dominant methods, they balance out well between over-fitting and under-fitting and are usually significantly faster than training deep neural networks.

Remark. *Source codes/presentation can be accessed at: Github Repository*

References

[1] Ajmera, P. K., Jadhav, D. V., and Holambe, R. S. Text-independent speaker identification using Radon and discrete cosine transforms based features from speech spectrogram. 2749–2759.

[2] Howbert, J. J., Patterson, E. E., Stead, S. M., Brinkmann, B., Vasoli, V., Crepeau, D., Vite, C. H., Sturges, B., Ruedebusch, V., Mavoori, J., et al. Forecasting seizures in dogs with naturally occurring epilepsy. *PloS one 9*, 1 (2014), e81920.

[3] Korshunova, I. Epileptic seizure prediction using deep learning.

[4] Mirowski, P., Madhavan, D., LeCun, Y., and Kuzniecky, R. Classification of patterns of eeg synchronization for seizure prediction. *Clinical neurophysiology 120*, 11 (2009), 1927–1940.

[5] Mirowski, P. W., LeCun, Y., Madhavan, D., and Kuzniecky, R. Comparing svm and convolutional networks for epileptic seizure prediction from intracranial eeg. In *2008 IEEE Workshop on Machine Learning for Signal Processing* (2008), IEEE, pp. 244–249.

[6] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research 15*, 1 (2014), 1929–1958.