Seizure forecasting

Xiaoying Pang
∗ xypang@stanford.edu

Abstract

The focus of this project is to predict impending seizure occurrence for epilepsy patients using intracranial EEG recordings. Reliable prediction of seizures can provide the patients with not only warnings but also new therapeutic possibilities. The goal is to correctly differentiate the pre-seizure brain state (the preictal state) from the baseline state (the interictal state) using just the iEEG signals. Both frequency domain and time domain features have been extracted and explored and ML algorithms such as logistic regression, lasso logistic regression, svm and knn have been applied. A kaggle score of 0.77486 was achieved using the logistic regression and frequency domain features.

Introduction

Seizure afflicts nearly 1% of the world's population. When it strikes, it can disable a person temporarily and cause serious injuries. Its unforeseeable nature makes a patient's daily life even harder. Many patients take preventive drugs daily. However, only a fraction of them can actually benefit from the medication, some of them can even suffer from physical and cognitive side effects from these drugs. The ability to forecast seizures will provide the patients with some certainties in their lives and enable on-demand treatments so that patients only need to take medicines when seizures are about to happen.

Clinical studies have suggested that a patient's brain activities can be classified into four states: interictal (between seizures, or baseline), preictal (prior to seizure), ictal (seizure), and postictal (after seizures). In this Kaggle challenge, we were ask to correctly identify the pre-seizure brain states, i.e. the preictal states from the baseline states i.e. the interictal states using the intracranial EEG data.

Due to the fact that iEEG patterns varies greatly across patients, in fact, one patient's preictal signals can look very similar to another patient's interictal signals, it is almost impossible to construct a single generic algorithm/model that will work for all patients. A more realistic approach is to find a patient-specific classification method based on features extracted from each individual patient's iEEG data.

In this study, I have explored both time and frequency domain features extracted from the iEEG signals and tested with a few ML algorithms. Details about the features extracted and the machine learning algorithms applied will be discussed in the following sections. Matlab was adopted as the scripting language for this project.

Data and Features

Data

Training and testing data for five animal and two human patients are provided by Kaggle. Each data file contains a 10 min long iEEG recording collected using various numbers of electrodes. Each row in the data file is the signal picked up by one of the electrode. The sampling frequency of the electrodes or channels was 400 Hz for the animals and 5000 Hz for humans. The training data are very unbalanced, e.g. for the first animal patient, 480 interictal training clips are provided, whereas only 24 preictal clips are provided. The total data size is about 60 GB.

Features

Inspired by [1–3] and the Kaggle discussion forum, the following features have been extracted from the iEEG data.

1. Shannon entropy for different frequency bands using windowed data
2. PCA transformed feature 1
3. Correlation between channels for windowed data
4. Correlation between channels for the whole 10 min data

To calculate feature 1, each 10 min data clip is first divided into 60(30)-second long epochs with 50% overlapping. Then the Hamming window is applied to each epoch to reduce the FFT power leakage. The FFT power spectral density (PSD) of one of the epochs is shown in Fig. 1. One can see that most of the

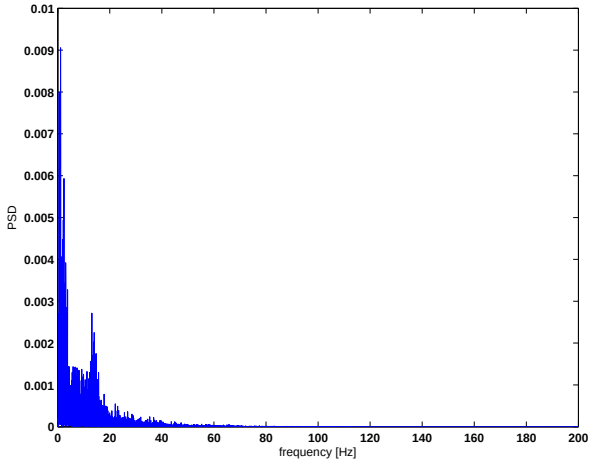

Figure 1. Power spectral density of a 60 second epoch.

spectral powers are concentrated in the low frequency region. Once the FFT PSD is calculated, it is then normalized by its summation over all frequencies, and used for the calculation of the Shannon entropy for different frequency bands. Shannon entropy [4] is a measure of the 'uncertainty' in a signal. It can be calculated according to Eq.(1)

$$entropy = \frac{\sum_k p_k \log p_k}{\log N} \tag{1}$$

where p_k is the PSD of the kth frequency, and N is the total number of frequencies in a frequency band. Finer frequency band widths have been chosen for lower frequencies due to the fact that most of the spectral powers are concentrated in these regions. Different frequency band settings have been tested and the most effective one is the 1 Hz bands from 0.25 Hz up to 30 Hz and 10 Hz bands from 30 to 180 Hz. To construct the final training matrix the band entropy for all the channels are concatenated. For example, there are 480 interictal and 24 preictal training clips for the first dog patient. By using a 60 second window, each 10 min clip can be divided into 19 epochs with 50% overlapping. The total number of epochs is $(480 + 24) \times 19 = 9576$. Then for each epoch, there are number of channels $\times$ number of frequency bands $= 16 \times 46 = 736$ features. So the final training matrix would be 9576 by 736. The same procedures have also been applied to the testing data. Since each testing clip has been divided into

several epochs, the final prediction of one clip is simply the average prediction across all epochs within the clip.

One of the advantages of the windowing technique is that it can help enlarge the training samples, which can be especially useful for this highly unbalanced data. Another advantage is that it helps to preserve more detailed structures of the signals rather than averaging them out in feature calculations.

In order to further decouple the features, PCA has been applied to get the demixing matrix and then used to rotate the features in order to obtain the decoupled final forms.

Besides the frequency domain, features like the variances and the correlations between different channels/electrodes are computed in the time domain. They are calculated for both the windowed data and the original 10 min clip data.

So far, feature 1 with 60 second window turns out to be the most effective kind.

ML Algorithms and Results

To formulate the classification problem, the preictal state is labeled as 1 and the interictal state is labeled as 0. The metrics precision = $\frac{TP}{TP+FP}$, recall = $\frac{TP}{TP+FN}$ and accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$ have been calculated during the cross validations. The Kaggle leader-board score is quoted as the testing accuracy. Although it is not very clear how the score is calculated, it may be obtained using the area under the ROC curve for all the patients. ML algorithms such as logistic regression, logistic regression with lasso regularization, svm and knn have been tested. The results shown in this section are all calculated using feature 1 mentioned in the previous section.

Logistic regression

Logistic regression works surprisingly well for this problem. Table 1 listed the training accuracies for all the subjects using 10-fold cross validation. Although the training metric values of the logistic regression are lower than those obtained by other methods, a Kaggle score of 0.77486 have been achieved, the highest among all the algorithms I have tried. Part of the reason might be that during the training, posterior probabilities have been mapped to 0 or 1 by comparing the probability to 0.5, which is probably not what Kaggle did when it calculated the scores.

Table 1. 10-fold CV results for logistic regression, Kaggle score = 0.77486

Subject	Precision(%)	Recall(%)	Accuracy(%)
Dog 1	30.3	43	92.8
Dog 2	92.3	90.1	98.7
Dog 3	75.8	62	97.2
Dog 4	73	65	93.6
Dog 5	71.4	79	96.7
Human 1	64.1	81.8	83.1
Human 2	36.7	55.8	57.7

Logistic regression with lasso regularization

The Matlab function 'lassoglm' with 'lambda' = 0.0001, 'alpha' = 0.9 is used. Parameter 'lambda' controls the magnitude of the l-1 norm, while the 'alpha' parameter controls the weight of lasso (l-1

norm) vs. ridge (l-2 norm) optimization. The cross validation accuracies are presented in Table 2. A Kaggle score of 0.7576 has been achieved by using this algorithm.

Table 2. 10-fold CV results for lasso LR, Kaggle score = 0.7576

Subject	Precision(%)	Recall(%)	Accuracy(%)
Dog 1	64.6	11.2	95.5
Dog 2	96.4	86.6	98.7
Dog 3	90.6	47.4	97.3
Dog 4	80.3	51.3	93.4
Dog 5	87.1	66.5	97.3
Human 1	97.9	97.4	98.8
Human 2	89.7	82.6	92.0

SVM

The Matlab command 'fitcsvm' with a Gaussian kernel was applied to the training matrix. It yields the best results when 'BoxConstraint' is set to 2.0 and 'Cost' is set to [0 1; 2 0] where the 'BoxConstraint' controls the magnitude of the regularization term, and the cost matrix indicates how different classification scenarios will be penalized. A larger 'BoxConstraint' value can force the svm to generate fewer support vectors, therefore reduce overfitting. A cost matrix of [0 1; 2 0] means that there will be no penalty if the prediction is correct, however, the penalty will be 1 if the prediction is an false alarm (prediction is preictal and the data is actually interictal) and 2 if the prediction is false negative (prediction is interictal, however, the data is actually preictal). Since for seizure prediction, a false negative is much more detrimental than a false alarm, more cost has been assigned to it. The cross validation accuracies of the SVM listed in Table 3 are the best among all the algorithms I have tried, however, it didn't achieve a better Kaggle learderboard score (0.75996) than that of the logistic regression (0.77486).

Table 3. 10-fold CV results for SVM, Kaggle score = 0.75996

Subject	Precision(%)	Recall(%)	Accuracy(%)
Dog 1	79.9	51.3	97.1
Dog 2	97.1	93.7	99.3
Dog 3	88.5	79.9	98.5
Dog 4	91.7	81.3	97.2
Dog 5	93	86.7	98.8
Human 1	96.8	97.1	98.4
Human 2	89.4	80.8	91.5

KNN

When training the KNN, twice weight have been assigned to the preictal data than to the interictal data since much less preictal data has been provided and we are more interested in the preictal state. This seems to be able to improve the prediction accuracy slightly. Table 4 lists the CV metric values of the KNN algorithm. The highest Kaggle score I was able to get by using KNN was 0.69451.

Table 4. 10-fold CV results for KNN, Kaggle score = 0.69451

Subject	Precision(%)	Recall(%)	Accuracy(%)
Dog 1	46.1	59.4	94.7
Dog 2	83.1	89.7	97.8
Dog 3	62.3	79.5	96.7
Dog 4	70.5	87.9	94.7
Dog 5	74.2	85.4	97.2
Human 1	91.4	99.1	97.3
Human 2	68.6	88.0	84.3

Discussion and Future Work

For this project, feature selection is the key. When I first started, I was using the variances and correlations between channels (feature 3 and 4 mentioned in the previous section) as my primary features. The best Kaggle score I was able to achieve using these time domain features never exceeds 0.665 which was obtained by using logistic regression. The frequency domain FFT features (feature 1) I tried later on turned out to be much better. In order to further decouple the features, I applied PCA to the extracted feature 1. However, that didn't help in terms of getting better CV accuracies and a higher Kaggle score. Besides, I also tried to concatenate all the frequency and time domain features for training, but the CV metric values and Kaggle scores I was able to get are all worse than just using the frequency domain features. Genetic algorithm has been applied to feature selection too, but no substantial gain has been achieved.

Logistic regression, lasso logistic regression and SVM all work very well for this problem. SVM produces the best CV accuracies, however, the logistic regression produces the best Kaggle score. This indicates a difference between the approach Kaggle used to calculate error metrics and mine.

In the future, more features can be included, e.g. the hjorth fractal dimension, mobility, complexity, and eigenvalues of the covariance matrix between channels, etc. And ML algorithms such as the neural network and random forest will be applied to this problem. A scheme combining predictions from an ensemble of models will be tested to see if better and more consistent performance can be achieved.

References

1. Howbert JJ, Patterson EE, Stead SM, Brinkmann B, Vasoli V, Crepeau D, Vite CH, Sturges B, Ruedebusch V, Mavoori J, Leyde K, Sheffield WD, Litt B, Worrell GA (2014) Forecasting seizures in dogs with naturally occurring epilepsy. PLoS One 9(1):e81920.

2. Ark Y, Luo L, Parhi KK, Netoff T (2011) Seizure prediction with spectral power of EEG using cost-sensitive support vector machines. Epilepsia 52:1761-1770.

3. Mormann F, Andrzejak RG, Elger CE, Lehnertz K (2007) Seizure prediction: the long and winding road. Brain 130: 314333.

4. C. Shannon, A mathematical theory of communication, Bell System Technical Journal, vol. 27, pp. 379423, 623656, 1948.