Predict seizures in intracranial EEG recordings

Linyu He (linyu90@stanford.edu) and Lingbin Li (lingbin@stanford.edu)

1 Abstract

This project aims to predict seizures in intracranial electroencephalography (iEEG) recordings using four algorithms. The data are a series of 10-minute iEEG clips labeled "preictal (positive)" for data recorded prior to seizures or "interictal (negative)" for data recorded between seizures. Our goal is to distinguish between the two states. The major challenge is that the data are highly imbalanced, i.e., the number of positive examples is less than 10% of that of negative examples. Our work is to make modifications to each of the four models and analyze the corresponding performance gain.

2 Introduction

Spontaneous seizures are the typical symptom of epilepsy, which is a common but refractory neurological disorder that afflicts nearly 1% of the world's population. Anticonvulsant medications are administered to many patients at high doses to prevent seizures, but their effectiveness is limited and patients often suffer their side effects. Even for patients whose epilepsy-causing brain tissue is removed via surgery, spontaneous seizures still persist. Due to the seemingly unpredictable occurrence of seizures, patients with epilepsy experience constant anxiety [**1**].

This project aims to make it possible that devices designed to monitor patients' brain activity can warn them of impeding seizures so that patients are able to take appropriate precautions. It is also helpful to reduce overall side effects caused by anticonvulsant medications taken by these patients. By providing them with devices with the ability to predict an impending seizure, anticonvulsant medications could be administered only when necessary, thus lowering the doses given to patients.

Multiple researches support the notion that the occurrence of seizures is not random. According to evidence shown by related researches, for patients with epilepsy, the temporal dynamics of brain activity can be classified into 4 states: interictal (between seizures), preictal (prior to seizures), ictal (seizure) and postictal (after seizures). The brain activity of each state can be recorded by iEEG [**1**]. Our goal is to employ machine learning techniques to learn from iEEG data the characteristics of preictal states and then distinguish these states from the interictal states. After one preictal state is identified, a warning should be sent to the patient to prepare him or her for an impending seizure.

3 Data and Feature Extraction

Kaggle provides iEEG data collected from canine subjects. The data of each subject is organized into 10-minute clips labeled "preictal (positive)" for data recorded prior to seizures or "interictal (negative)" for data recorded between seizures. Each clip contains 16 channels of iEEG data where each channel corresponds to one electrode implanted in the subject's brain. For each training example $(x^{(i)}, y^{(i)})$, $y^{(i)}$ is the label and $x^{(i)}$ is a clip in which each row corresponds to one channel and each column corresponds to iEEG readings at one sampling time point.

Since seizures in most patients are associated with a stereotypic EEG discharge with characteristic spectral pattern, we employed the following feature extraction procedure [**2**]:

Apply fast Fourier transform to each channel in a clip and divide the resulting power spectrum into 6 bands: delta (0.1 – 4 Hz), theta (4 – 8 Hz), alpha (8 – 12 Hz), beta (12 – 30 Hz), low-gamma (30 – 70 Hz), and high-gamma (70 – 180 Hz). In each band, sum the power over all band frequencies to produce a power-in-band (PIB) feature. Therefore, 6 features are obtained in each channel and 96 features are obtained in one clip. The procedure above is also illustrated in Fig 1, where $p(f)$ is the power spectrum.

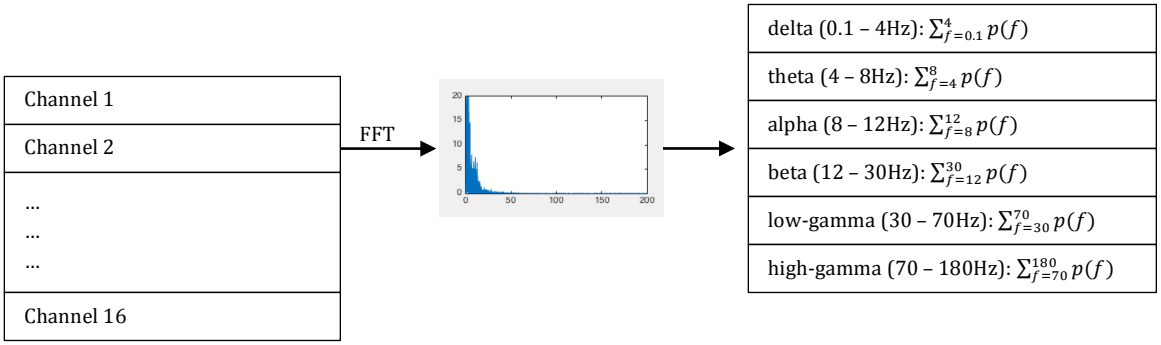

Fig. 1 Feature Extraction using FFT

4 Cross Validation

There are 3939 examples in total, in which 3674 are negative and 265 are positive. For each time of cross validation, we randomly pick 70% of the negative examples and 70% of the positive examples for training and use the rest for testing. This process is repeated for 100 times to calculate the average evaluation.

Besides the training error and the testing error, the following values are adopted to evaluate the performance of each model since the data are highly imbalanced.

Table 1. Values chosen to evaluate the performance

Name	Definition
Accuracy (ACC)	$\frac{TP+TN}{TP+TN+FP+FN}$
Positive Predictive Rate(PPV)/Precision	$\frac{TP}{TP+FP}$
True Positive Rate (TPR)/Recall	$\frac{TP}{TP+FN}$
False Negative Rate (FNR)/Miss Rate	$\frac{FN}{TP+FN}$
False Positive Rate (FPR)/Fall-out	$\frac{FP}{FP+TN}$

In Table 1, TP, TN, FP and FN are the number of true positives, true negatives, false positives and false negatives, respectively. Finally, the receiver operating characteristic (ROC) curves and precision-recall curves will be plotted based on values in the table above.

5 Learning Algorithms

In out attempt to seek a solution, three models covered in CS 229 were first adopted, which are logistic regression, naïve Bayes classifiers and support vector machines (SVMs). Later, we employed a model inherited from communication systems, which makes a prediction based on correlation coefficients between the test example and all training examples. Modifications are made to each model to improve their performance on an imbalanced data set. In the following discussion, we use $(x^{(i)}, y^{(i)})$ to denote each training example where $x^{(i)} \in R^{96}$ is the set of features and $y^{(i)} \in \{0, 1\}$ is a label (0 corresponds to a negative label and 1 corresponds to a positive label. In SVMs, $y^{(i)} \in \{-1, 1\}$ where -1 corresponds to a negative label and 1 corresponds to a positive label). We use $\hat{X}$ to denote a query point (test example) and $\hat{Y}$ to denote the label predicted by a model.

5.1 Logistic Regression

In logistic regression, the hypothesis is $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ where θ is the parameter. The probability of $\hat{Y} = 1$ conditioned on $\hat{X}$ and parameterized by θ is $P(\hat{Y} = 1|\hat{X}, \theta) = h_\theta(\hat{X})$.

Usually, the prediction that $\hat{Y} = 1$ is made if $h_\theta(\hat{X}) \geq 0.5$. Since the data set is highly imbalanced, i.e., the number of negative examples is much larger than that of positive ones, we consider it more important to correctly classify more positive test examples than to have a few false positives. Therefore, the decision threshold of $h_\theta(\hat{X})$ can be less than 0.5, which makes it more likely to classify a test example as positive. We set the decision threshold to be η where $\eta \in [0, 0.5]$ and plot the cross validation results when choosing different values of η, which is shown in Fig. 2.

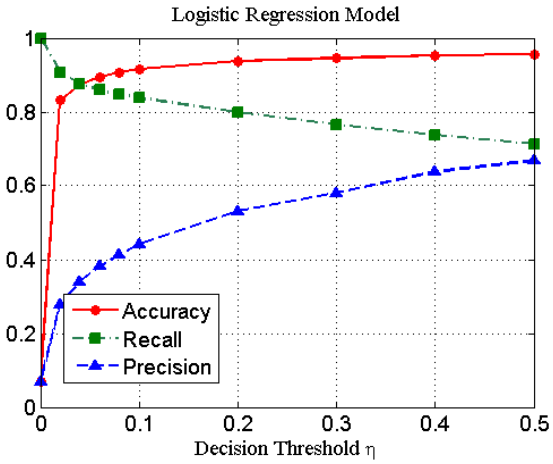

Fig. 2 The relation between the decision threshold η and accuracy, recall and precision of the logistic regression model

It can be seen in Fig. 2 that when $\eta = 0.5$, both accuracy and precision are high, but the recall is not satisfactory. When $\eta = 0$, we achieve the maximum recall but the accuracy and precision are very low. So a trade-off has to be made between recall and precision/accuracy by choosing an appropriate value of η. For example, if $\eta = 0.04$, both accuracy and recall are high and close to each other, meaning the accuracy for classifying all test examples and the one for classifying positive test examples are close. The low precision when $\eta = 0.04$ is caused by the increased number of false positives, which is acceptable to some extent since false positives are less important than true positives in seizure prediction.

5.2 Naïve Bayes

The multinomial distribution is used to model the features of each iEEG clip. Since the value of each feature is continuous, we first discretize the values into N groups where $N = \frac{V_{max}}{G}$, V_{max} is the maximum value of the features of all clips and G is the group size. Similar to the modification made to logistic regression, a test example is labeled "1" if $QP(\hat{Y} = 1|\hat{X}) \geq P(\hat{Y} = 0|\hat{X})$ where Q is a positive constant specified to overcome the imbalanced data when making predictions. Fig. 3 indicates that the naïve Bayesian model cannot make satisfactory predictions no matter what the value of Q is.

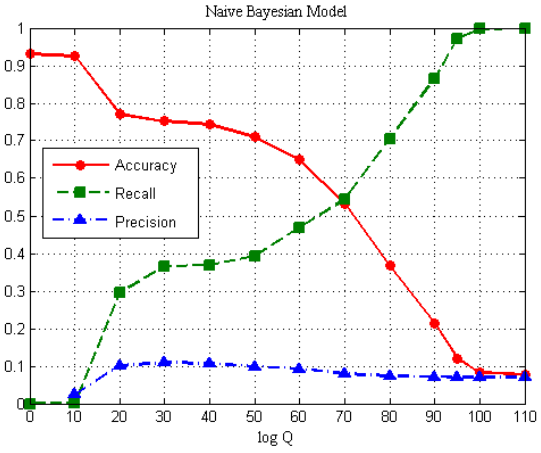

Fig. 3 The relation between Q and accuracy, recall and precision of the naïve Bayesian model

5.3 Support Vector Machine (SVM)

In the l_1-regularized SVM, the hypothesis is $h_{w,b}(x) = w^T x + b$ where parameters w and b are obtained by solving the primal optimization problem whose objective is $\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$, where $C\sum_{i=1}^{m}\xi_i$ is the cost term. Since it is more important to correctly classify more positive test examples than to have a few false positives, the 2-cost-sensitive SVM (2C-SVM) [**3**] is adopted in which two different costs are assigned to negative and positive examples, respectively. In 2C-SVM, the objective of the primal optimization problem is $\min_{w,b} \frac{1}{2}\|w\|^2 + C^+\sum_{i\in I^+}\xi_i + C^-\sum_{i\in I^-}\xi_i$ where $C = C^+ + C^-$ is a trade off between the classification margin and misclassified or non-separable examples and the cost factor $R = \frac{C^+}{C^-}$ is the ratio of costs between positive and negative examples. We employ the LIBSVM library [**4**] as our 2C-SVM implementation.

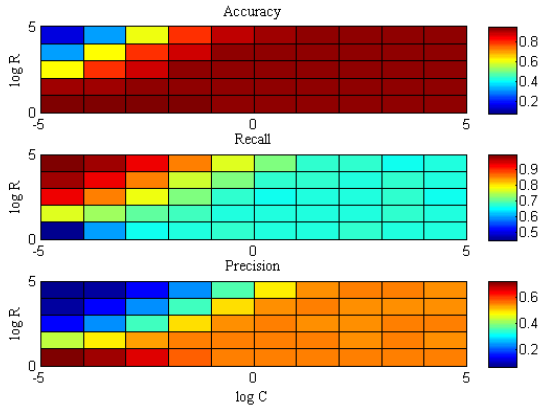

Fig. 4 The relation between C, R and accuracy, recall and precision of the 2C-SVM model

Different values of $C > 0$ and $R \geq 1$ are chosen and the corresponding results of accuracy, recall and precision are shown in Fig. 4. Satisfactory accuracy and recall can be achieved when (C, R) is chosen as $(10^{-4}, 10^2)$, $(10^{-3}, 10^3)$ or $(10^{-2}, 10^4)$, etc.

5.4 Correlation Decision

We derived the correlation decision model from a few concepts in communication systems. The previous three models discard the training set after a hypothesis is built. However, in correlation decision, we do not use training examples to build a hypothesis and we keep the entire training set.

Consider a given query point $\hat{X}$. Calculate its correlation with each training example and assign a score to each of them: $Score(i) = corr(\hat{X}, X^{(i)}), \forall i = 1,2 \dots m$. Find the training example with the maximum score: $i^* = argmax\ Score\ (i)$. Since $X^{(i^*)}$ is the training example that the query point is most similar to, we can make a prediction that $\hat{Y} = Y^{(i^*)}$.

In order to classify more positive examples correctly, in other words, to output more positives, we increase the scores for positive training examples by a factor $\gamma \geq 0$, namely, $Score := Score \times (1 + \gamma)$ for all positive training examples.

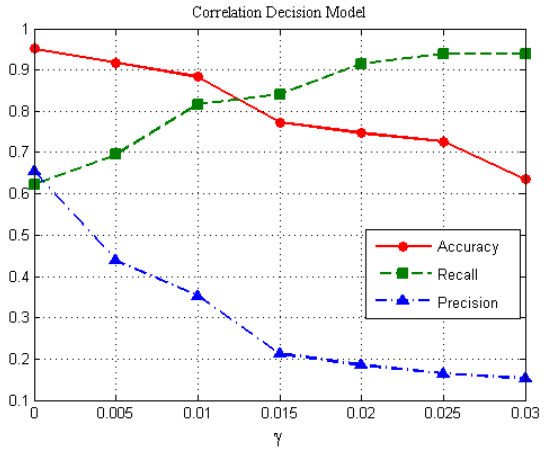

Fig. 5 The relation between γ and accuracy, recall and precision of the correlation decision model

5.5 Others

We also tried extracting features using PCA and ICA, using different kernels (such as the Gaussian kernel and the sigmoid kernel) in SVMs, and capturing non-linear behaviors of features (such as $\log x$, $\sqrt{x}$ and x^2). But we didn't achieve improvement in performance.

6 Results and Discussion

Table 2 shows the results of the models discussed above, where for each model, the results before and after our modification to these models are compared. For each modified model except naïve Bayes, the decision parameters that achieve an acceptable performance are shown. Among all chosen models, logistic regression with a threshold of 0.04 works best, in whose results both accuracy and recall are close to 90%.

Table 2. Performance results of different models

Model	Decision Parameters	Training Error	Test Error	FNR	FPR	ACC	Precision	Recall
Logistic Regression	$\eta = 0.5$	0.0046	0.0443	0.2871	0.0262	96%	67%	71%
	$\eta = 0.04$	0.0567	0.1255	0.1232	0.1257	88%	34%	88%
Naïve Bayes	$Q = 1$	0.0685	0.0691	1	0	93%	NAN	0
	$Q = 10^{70}$	0.1993	0.4688	0.4565	0.4697	53%	8%	54%
2C-SVM	Linear Kernel; $C = 1$ $R = 1$	0.0018	0.0610	0.3273	0.0412	94%	55%	67%
	Linear Kernel; $C = 10^{-4}$ $R = 10^2$	0.0387	0.1994	0.1453	0.2034	80%	24%	85%
Correlation Decision	$\gamma = 0$	-	0.0489	0.3780	0.0244	95%	66%	62%
	$\gamma = 0.0125$	-	0.1502	0.1341	0.1521	85%	28%	85%

Fig. 6 shows the ROC curve and the precision-recall curve of each model. The area under curve (AUC) is calculated by using the trapezoidal areas created between each point [**5**]. It can be seen that logistic regression outperforms others since its AUCs for ROC and precision-recall curves are the highest. Correlation decision model also provides high AUCs for both kinds of curves, which indicates its performance is close to that of logistic regression.

Seizure prediction is usually performed in real time. Table 3 gives the comparison of average runtime for different models. The test is performed on the same PC with a 2.2GHz CPU. It can be seen that logistic regression consumes the least time. Since it also has the highest prediction performance, it is the most cost-efficient algorithm in this situation. Although the performance of the correlation decision model is as good as that of logistic

regression, it runs much more slowly, since it has to keep track of all training examples during the prediction process. Therefore, we consider logistic regression with threshold modification as the best model in this project.

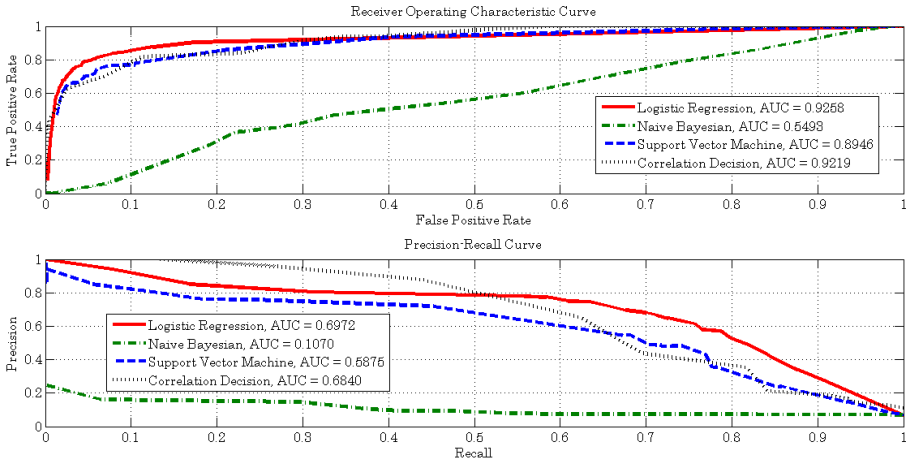

Fig. 6 The ROC curve and precision-recall curve of each model

Table 3. The average runtime of different models (# of training examples : 2758, # of test examples : 1182)

Model	Logistic Regression ($\eta = 0.04$)	Naïve Bayes ($Q = 10^{70}$)	2C-SVM ($C = 10^{-4}$, $R = 10^{2}$)	Correlation Decision ($\gamma = 0.0125$)
Runtime (in seconds)	0.267448	0.488759	0.432028	12.8690

The major challenge of this project is the imbalanced data. What we've done so far is to sacrifice false positive rate to achieve a low false negative rate because a false negative is far more dangerous than a false positive in seizure prediction. The current results are within our expectations but they are not good enough since we believe the information extracted from the limited number of positive examples is not enough to perfectly distinguish between the two classes.

7 Conclusions and Future Work

Three supervised learning models covered in CS229 and one model inherited from communication systems are employed in this project to predict the occurrence of seizures. Modifications are made to these models to deal with the highly imbalanced data. Among the four models, logistic regression outperforms others, which obtains the highest AUCs for ROC and precision-recall curves. Meanwhile, when choosing the decision parameter properly, both accuracy and recall of logistic regression are close to 90%.

An important method to deal with imbalanced bits "0" and "1" in wireless communication is to code "0" into "01" and "1" into "10" so that the numbers of both classes become balanced. We've been trying to apply similar ideas to the project, but have yet got a satisfying result. So the exploration will be continued to seek better solutions.

Also, instead of using a single model to build a classifier, attempts can be made to combine the predictions of different models and develop strategies to make a final decision. Models involved in the combination may differ in their feature extraction process since it is possible to develop for each model the features that best fit the model.

8 References

[1] Kaggle Inc. (2014) Kaggle: The Home of Data Science. [Online]. http://www.kaggle.com/c/seizure-prediction

[2] J. Jeffry Howbert et al., "Forecasting seizures in dogs with naturally occurring epilepsy," *PloS one*, vol. 9, no. 1, p. e81920, 2014.

[3] Yun Park, Lan Luo, Keshab K. Parhi, and Theoden Netoff, "Seizure prediction with spectral power of EEG using cost-sensitive support vector machines.," *Epilepsia 52*, no. 10, pp. 1761-1770, 2011.

[4] Chih-Chung Chang and Chih-Jen Lin. (2014) LIBSVM -- A Library for Support Vector Machines. [Online]. http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[5] Jesse Davis and Mark Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233-240.