Comparative Analysis of Machine Learning Techniques for Seizure Prediction

Evan Hofmeister — evanhof@uw.edu

1 Abstract

I examine the effectiveness of several statistical and machine learning methods for classifying Intracranial Electroencephalography (iEEG) data as pre-ictal or inter-ical in order to predict seizure events. The main performance criterial will be ROC-AUC. For patients who suffer from seizures that cannot be treated with anti-seizure medications or surgery, advances in the prediction of seizure events is critical for the development of successful treatments.

2 Introduction

Epilepsy is a neurological disorder in which brain activity becomes abnormal and unpredictable interruptions of normal brain activity occur (1,2). Epileptic seizures can reliably be predicted from electrocorticographic data (ECoG) which is collected from electrodes monitoring electrical activity produced by the cerebral cortex (3). Epileptic seizure prediction research centers around creating predictive models from preictal and interical data segments. The former being the occurance of a seizure and the later being the absence.

3 Past Research

Past research into epileptic seizure prediction has focused on statistical analysis of ECoG including moments and similarity indices (4). More recent research has focused on the use of deep learning classifiers for prediction (5). A newer approach is to use spectral power approach, as features of the model. In this research this is commonly acomplished with Fast-Fourier Transform or a similar fourier approach. Another point worth mentioning is that most studies have utilized a moving window of time to summerize data and develop features.

Past research utilizing these methods have presented successful models that have included GLM, Logistic, Gradient-Boost, SVM, and Random Forest Models. For this project, I tested several sk-learn dependencies and then improved apon the approach that showed the most promise (4,5).

4 Data and Features

Both Melbourne University's Epilepsy Ecosystem platform and The American Epilepsy Society release and maintain ECoG data consisting of 10 minute electrode channel time series data observations (16 cerebral cortex electrode locations in total) (6). These ECoG data clips were either Pre-ictal or Interictal, the former being pre-seizure data segments and the latter being non-seizure data segments. Each data clip has 16 channels where each band is a 400 Hz subset. Each clip is 600-700 seconds with one sample per second. The data includes five patients, with on average 2000 recordings per patient (6). Only the fifth patient's data was used since all other patients experienced large events of signal drop, where the channel has a values of 0. Often bands did not have signal drop at the same time or for the same duration, so ommiting these values did not produce a successful model. This issue is discussed more in the 'Conclusion and Future Work' section of this text.

As mentioned in the in the past research portion, much of the current research focuses on analysis of the power of each band. A common method is to use Fast Fourier Transform (FFT). This method was used with a few summary steps to normalize the dataset. The data is then broken into testing and training sets with a proportion equal to ratio in the original given data. In later sections, I discuss why using Precision-Recall in addition to AUROC is a good way to evaluate the models (7,8).

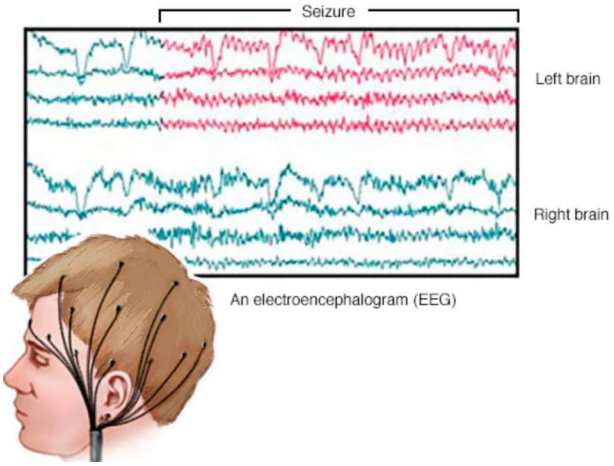

Figure 1: ECoG data channal frequency example (9)

5 Pre-Processing

The data needed to be preprocessed to both prepare it for statistical and machine learning modeling methods. Preprocessing is most necessary because in total each patient's data there are at least 5 million time-point observations for both pre-ictal and interictal data. This is a very large dataset to attempt to implement machine learning algorithms on, so it will be beneficial to implement feature selection.

This included summarizing the electrode band data into smaller subsets to reduce the dimensionality. To do this a Fast-Fourier Transform is perfromed. This transformation is able to express that data as a sum of sinusoids. This is helpful because we are transforming a time domain signal in frequency so that we can identify periodicity. This transformation assists us in creating elements and features based on the frequency of the data (10).

The data is then grouped into bins to further red reduce the size of the dataset. Below you can see a visual representation of the frequency of each channel before and after the FFT transformation and a few other procedures to normalize the data (11).

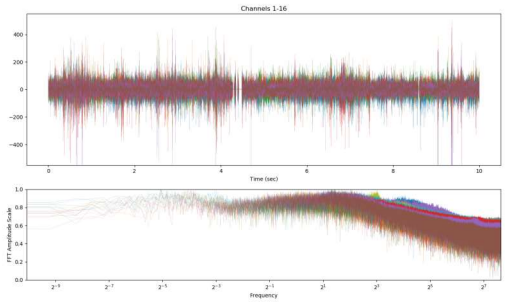

Figure 2: Pre-proccessing

5.1 Generalized Linear Model

Generalized Linear models take the form of:

$$f(x) = w_0x_0 + w_1 + x_1...w_dx_d = w^Tx$$

These models are optimized by minimizing the mean-squared error (12):

$$E_n = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$$

5.2 Logistic Regression

For logistic regress, the hypothesis takes the form of a logistic function:

$$p_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

The cost function for LR (l_2-regularized) minimizes the negative log-liklihood loss (13,14):

$$min_\theta \ \lambda||\theta||^2 + \sum_{i=1}^{n} log(1 + exp(-y_i\theta^T x_i))$$

5.3 Support Vector Machine

SVM uses hinge loss function:

$$l(y) = max(o, 1 - y)$$

This will minimize the cost function:

$$min \ \frac{||\theta||^2}{2} + C\sum_{i=1}^{n} max \ \{0.1 - y_i x_i^\theta \theta\}$$

A support vector machine is an excellent choice for a task like this because it's able to perform binary clasification and create a hyperplane across features to best minimize the above cost function (15).

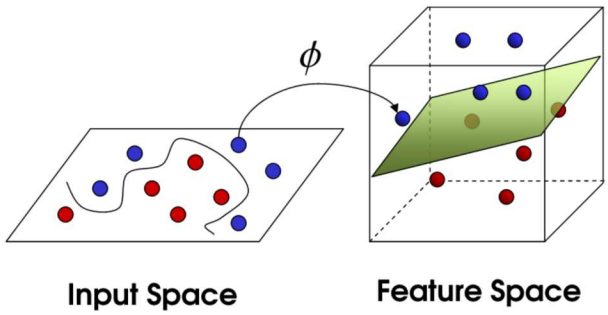

Figure 3: SVM Kernelling

5.4 Random Forest

Random Forest is a supervised machile learning algorithm that is is composed of a series of trained decision trees that apply a technique, bootstrap aggregating which is a form of bagging.

After training, predictions of hidden samples x' can be found by averaging predictions from individual regressions trees (16,17):

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

6 Results

6.1 Mesuring Performance

Typically AUROC curves are used to evelute this type of model when there is an even balance of samples between each class. This application is unique in that inter-ictal samples will always be more prevalent than pre-ictal samples due to the infrequency of seizure events. For this reason, in additional to using AUROC, the models Precision-Recall will also be examined since they are best suited for situations where there is an imbalance between classes.

Also, as mentioned by [citation], it is important correctly classify pre-ictal events, in order to not miss a seizure event, so recall is regarded as an important measure.

Below are the ROC curves plotted on the same indecies. This ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) at varying thresholds. Intuitively we can understand AUROC to be equivalent to the probability that a classifier will rank a randomly selected positive instance higher than a randomly chosen negative instance.

The precision-recall curve is also plotted below for each approach. This metric captures the trade-off between the TPR and positive predictive value by incrementing a threshold. This metric along with recall is important because we would like to maximize the number of correct identifications of a seizure over the entire sample space (18,19,20,21).

6.2 Hyperparameter Optimization

From the top perfroming models in the previous section, we can run simulations to optimize the hyper perameters within each model. For example for SVM, we can optimize the penalty parameter, by iterating through a range of possible values and selecting the score that produces the highest score. For hyperparameter tuning, I have used AUROC as the score criteria. From the results, a penalty parameter of 534 was found. More expansive windows were tested but this was a very computationally heavy proccess so progress was limmited.

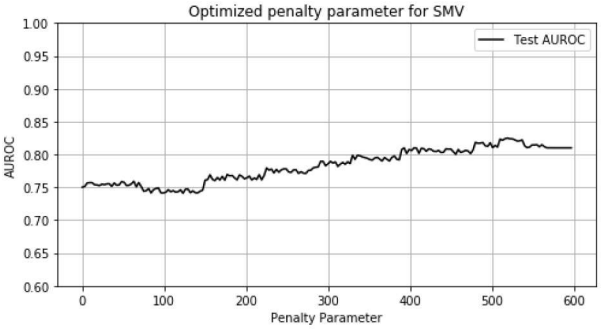

Figure 4: Optimal penalty parameter for SVM

Similarly for the K-Nearest-Neighbor model, we can optimize the the number of neighbors used within the base KNN model. The optimal values was found to be 17. The number of neighbors is significant because it's the number of neighbors considered and used to smooth estimates at a certain point. In the results both 16 and 17 yeilded similar results but often its helpful to choose the number of neighbors to be an odd number, since this would prevent tied votes in the algorithm (22).

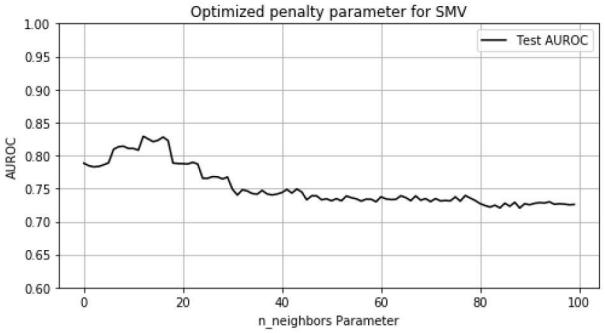

Figure 5: Optimal Nneighbor for KNN

For Logistic Regression we can use a grid search to find the optimal value of parameter C. Parameter C is the inversion of regularization strength From the grid search results below, we can see that the optimal value is one, which is the default. Increasing this parameter only reduces the performance of the regression (23).

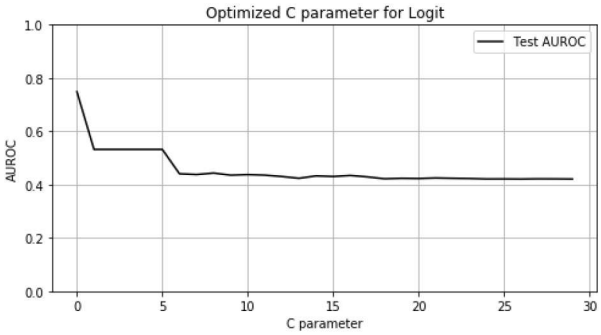

Figure 6: Optimal Logistic Parameter C

6.3 Performance Comparison

After retuning the models, we can see how they compare in terms of ROCAUC and Precision-Recall. The best model model in terms of these two metrics was the optimized SVM model. This model found an AUC score of 0.822 and recall of 0.521. Similarly, the KNN model was also moderately successful with an AUC of 0.801 and Recall of 0.484. From previous research, FFT paired with SVM often seems to be a successful pairing for creating the most predictive model that also maximizes recall.

	GLM	Logit	SVM	KNN	RF
AUC	0.684	0.746	0.822	0.801	0.702
Recall	0.407	0.464	0.521	0.484	0.446

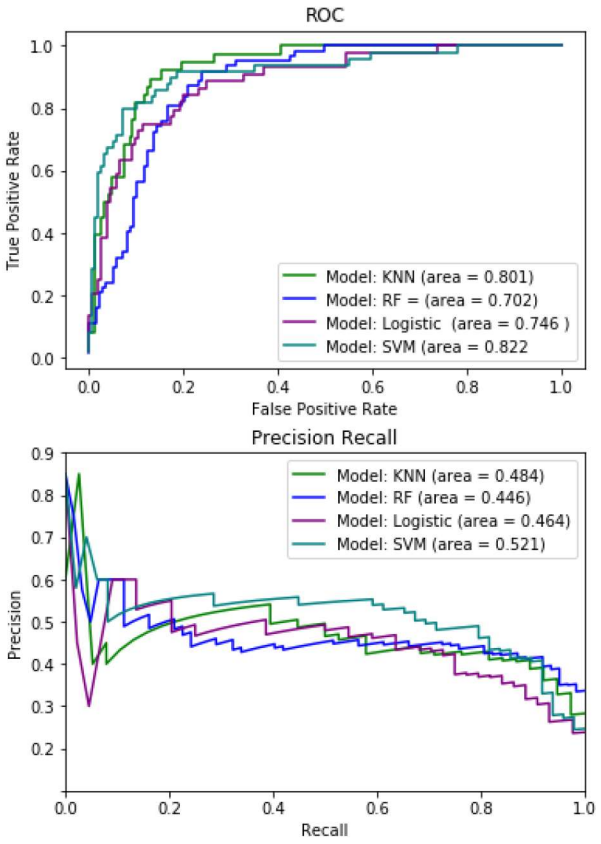

Figure 7: AUROC and Precision-Recall Curves

7 Conclusion and Future Work

From previous research, my current results could be improved by taking into account the correlation between channels from a FFT or similar, transformations based on the frequency of the data. I attempt to implement this, but did not find sound results, so I have abandoned this approach until I have a better theoretical framework on how to include this. I believe, the most gains to be had lie in improving the pre-proccessing procedures, whether that be to include additional features or re-evaluate implemented features. Another area to explore is using multiple patients data to create one uniform model. This was scrapped in the testing phase because other patients data (1-4) experienced very large segments of signal drop where, after preproccessing and removing these segments, no model turned out to be significantly successful. Another approach I would like to test is modifying the window used to summerize data. For example, instead of using a rolling 40 second window, that could be expanded or contracted or possibly be dynamic depending on other factors. This would also most likely produce significantly different results depending on the size of the window because these algorithms are very sensitive ot input data. It's often the case in seizure detection that models trained on one patient will perform poorly on others. Ideally this type of research will yeild more universal models that would be predictive for more than one individual.

Works Cited

(1) "Epilepsy." Mayo Clinic, Mayo Foundation for Medical Education and Research, 13 June 2018, www.mayoclinic.org/diseases-conditions/epilepsy/symptoms-causes/syc-20350093.

(2) "Epileptic Seizure." Wikipedia, Wikimedia Foundation, 18 Oct. 2018, en.wikipedia.org/wiki/Epileptic_seizure.

(3) "Electrocorticography." Wikipedia, Wikimedia Foundation, 6 Oct. 2018, en.wikipedia.org/wiki/Electrocorticography.

(4) Kiral-Kornek, Isabell, et al. "Epileptic Seizure Prediction Using Big Data and Deep Learning: Toward a Mobile System." EBioMedicine, vol. 27, 2018, pp. 103-111., doi:10.1016/j.ebiom.2017.11.032.

(5) Mormann, F., et al. "Seizure Prediction: the Long and Winding Road." Brain, vol. 130, no. 2, 2007, pp. 314-333., doi:10.1093/brain/awl241.

(6) "About." Epilepsy Ecosystem, www.epilepsyecosystem.org/.

(7) Usman, Syed Muhammad, et al. "Epileptic Seizures Prediction Using Machine Learning Methods." Computational and Mathematical Methods in Medicine, vol. 2017, 2017, pp. 1-10., doi:10.1155/2017/9074759.

(8) Parvez, Mohammad Zavid, and Manoranjan Paul. "EEG Signal Classification Using Frequency Band Analysis towards Epileptic Seizure Prediction." 16th Int'l Conf. Computer and Information Technology, 2014, doi:10.1109/iccitechn.2014.6997315.

(9) "EEG (Electroencephalogram)." Mayo Clinic, Mayo Foundation for Medical Education and Research, 7 Dec. 2018, www.mayoclinic.org/tests-procedures/eeg/about/pac-20393875.

(10) "Fourier Transform." Wikipedia, Wikimedia Foundation, 11 Dec. 2018, en.wikipedia.org/wiki/Fourier_transform.

(11) Bogaarts. "Quantitative EEG and Machine Learning Methods for the Detection of Epileptic Seizures and Cerebral Asymmetry." doi:10.26481/dis.20170223jgb.

(12) "GLM." Statmath.wu.ac.at, statmath.wu.ac.at/courses/heather_turner/glmCourse_001.pdf.

(13) "Logistic Regression." University of Wisconsin, pages.cs.wisc.edu/~jerryzhu/cs769/lr.pdf.

(14) "Classification of Patterns of EEG Synchronization for Seizure Prediction." NeuroImage, Academic Press, 17 Oct. 2009, www.sciencedirect.com/science/article/pii/S1388245709005264.

(15) "Machine Learning Linear Regression ." University of Pittsburgh , people.cs.pitt.edu/~milos/courses/cs2750-Spring03/lectures/class6.pdf.

(16) "Random Forest." Wikipedia, Wikimedia Foundation, 4 Dec. 2018, en.wikipedia.org/wiki/Random_forest.

(17) Donges, Niklas. "The Random Forest Algorithm - Towards Data Science." Towards Data Science, Towards Data Science, 22 Feb. 2018, towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd.

(18) "False Positive Rate." Wikipedia, Wikimedia Foundation, 22 Nov. 2017, en.wikipedia.org/wiki/False_positive_rate

(19) "Receiver Operating Characteristic." Wikipedia, Wikimedia Foundation, 12 Dec. 2018, en.wikipedia.org/wiki/Receiver_operating_characteristic.

(20) Hidden Markov Model and Naive Bayes Relationship, www.davidsbatista.net/blog/2018/08/19/NLP_Metrics/.

(21) Mirowski, Piotr, et al. "Classification of Patterns of EEG Synchronization for Seizure Prediction." Clinical Neurophysiology, vol. 120, no. 11, 2009, pp. 1927-1940., doi:10.1016/j.clinph.2009.09.002.

(22) "K-Nearest Neighbors Algorithm." Wikipedia, Wikimedia Foundation, 3 Nov. 2018, en.wikipedia.org/wiki/K-nearest_neighbors_algorithm.

(23) "Sklearn.linear_model.LogisticRegression." 1.4. Support Vector Machines - Scikit-Learn 0.19.2 Documentation, scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.