Epilepsy Seizure Prediction – Performance of uni-variate and bivariate features using Logistic Regression and SVMs

Kushank Raghav
University of Michigan, Ann Arbor, MI

Abstract

Epilepsy affects 1% of world's population and is marked by spontaneous occurrence of seizures. There is no available reliable seizure prediction methods. A seizure prediction algorithm is proposed that will classify an Intracranial EEG segment as interictal (between seizures) or preictal (prior to seizure). I focused on computing linear bivariate features such as correlation coefficient between all pairs of channels and Eigen values of correlation coefficient matrix between all pairs of channels and linear uni-variate features such as power in eight different frequency bands. I employed L2-regularized Logistic regression and Support vector machines with non-linear kernels to make predictions. In certain cases, a 10 minute window was divided into 3 equal segments, each totaling 200 seconds. My best machine learning model, when applied to EEG recordings of 7 subjects in the Kaggle's American Epilepsy Seizure Prediction Challenge database, produced an area under the ROC curve (AUC) score of 0.72690. This algorithm can be run on an implantable device and it can predict a seizure with moderate predictive accuracy.

Introduction

Epilepsy is a neurological disorder. Epilepsy patients suffer recurrent seizures (convulsions) over time. Epilepsy medications often cause side-effects. Medications are not effective for 20-40% of epilepsy patients. Epilepsy surgery removes the area of the brain which causes epilepsy. However, despite the surgical intervention, many patients continue to suffer from spontaneous seizures. If an epilepsy patient suffers a seizure during activities such as swimming and driving, it can be potentially dangerous. If a computational algorithm can reliably predict a seizure few seconds or minutes before it is about to happen, it can be used to warn an epilepsy patient of an impending seizure. **(1)**This warning can help an epilepsy patient avoid embarrassing or life-threatening situations. However, there is no algorithm that can reliably do seizure prediction.

The specific problem, in seizure prediction, is to classify an EEG segment as interictal or preictal. An EEG data segment is a time series signal and each value represents electrical activity in the brain at a particular time. An Intracranial EEG segment can have multiple channels. This is because in a given subject, electrodes to record EEG can be placed in different areas of brain. Each of these single electrodes record EEG data. When an EEG segment from a single subject is considered, each electrode recording is a separate row and the EEG values at the same

time instance are columns. **(1)** This is shown in Fig. 1.

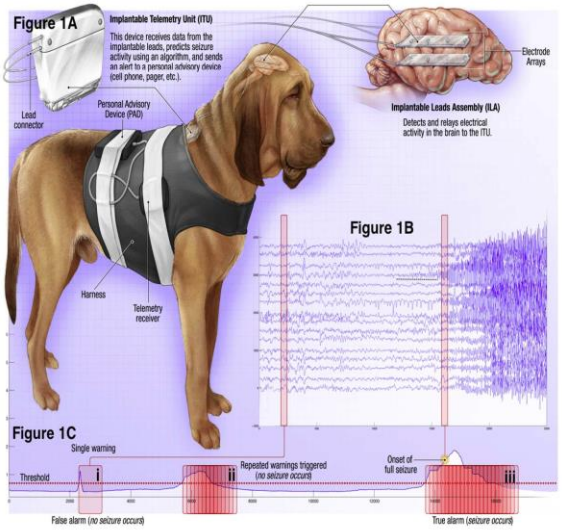

Fig. 1. A dog is pictured here, with implantable electrodes recording Brain EEG. EEG from this single dog will have multiple channels. Segments of equal time lengths are taken from this EEG. **(1)**

I have used machine learning to train an algorithm by providing labelled EEG data and the trained model is used to make predictions. This is done by first extracting features from a given EEG data segment. These EEG data segments will be labelled as interictal or preictal. The machine learning algorithm can learn that a label corresponds to a particular feature data set. When an unlabeled EEG segment is fed to the algorithm, the algorithm extracts the features and based on what it learned during the training phase, it will predict the label of this data segment. Features can be computed for each channel separately (uni-variate) or can be computed as relationship between pairs of channels (bivariate).

Research by Williamson, Bliss, Browne, and Narayanan (2012) supports EEG seizure prediction using Eigen spectra of pairwise space-delay correlation and covariance matrices between EEG channels and support vector machines. **(2)**Research by Netoff, Park, and Karhi (2009) demonstrated good seizure prediction using power in different frequency bands and cost-sensitive support vector machines. **(3)** My seizure prediction approaches are a modification of the approaches outlined in these papers. I am going to discuss the modifications briefly. In the first method using bivariate features, I am using Logistic regression in addition to a SVM. In the first paper, only SVMs are used. Also, I am using correlation coefficient and Eigen values of correlation coefficient matrix. This is a different feature set and a different feature combination. In the second method, I am using power in different frequency bands. In the second paper, total power is used in addition to power in each frequency bin. I have not used total power in my feature set. The second paper assigns a bigger weight to preictal than interictal. I am weighting both classes equally. In the second paper, 3-of-5 window analysis is done. I am using 2-of-3 window analysis. Also, I am standardizing the signal before computing features. This approach is not used in both papers. I have also applied 2-of-3 window analysis in first method. These are some major differences, in addition to a lot of smaller ones that include the different database and result metrics.

Methods

A. Kaggle Database

The dataset was sourced from a competition hosted by Kaggle. The title of the competition was "American Epilepsy Seizure prediction Challenge". The dataset consists of Intracranial EEG recordings from 7 different subjects. Five of these subjects are dogs and two of them are humans. The dataset from each subject has multiple interictal and preictal EEG segments. The EEG data in each segment is a matrix of EEG sample values

arranged row x column as electrode x time. Each segment also contains information about sampling frequency. The duration of each segment is 10 minutes. (1) Each subject dataset also contains unlabeled test segments. The probability of each of these test segments being preictal was calculated. After predictions have been made for each subject, all predictions are written into a single csv file. This csv file is submitted at Kaggle competition website and it returns an AUC score. I have used the same AUC scores in this paper.

B. Preprocessing

For each channel in the signal, we subtract the mean from each of its values and divide the result by standard deviation.

C. Feature Extraction

The first method used bivariate features. Two feature sets were extracted: a) correlation coefficient between all pairs of channels b) Eigen values of correlation coefficient matrix between all pairs of channels. Both these features were calculated using functions from "numpy" module. (4)

Correlation coefficient between X and Y is a measure of linear dependence between X and Y.

Correlation coefficient matrix, P, in terms of covariance matrix C, is defined as:

$$P_{ij} = \frac{C_{ij}}{\sqrt{C_{ii} * C_{jj}}} \quad (5)$$

For a square matrix, A, there is a non-zero vector that when multiplied with A, results in a scalar multiple of itself.

$$\mathbf{A}\mathbf{X} = \lambda \mathbf{X}$$

λ = eigenvalue of A. (6)

These features were tested with a Logistic Regression model and a support vector machine with 'rbf' kernel. In case of a Logistic Regression model, these features were calculated on the entire 10 minute window. However, In case of the support vector machine, each 10 minute window was divided into 3 equal chunks, each of 200 seconds duration, and these features were calculated for each 200 second window.

The second method used a uni-variate feature. Power in the following frequency bands was calculated for each channel: delta (0.5-4Hz), theta (4-8Hz), alpha (8- 13Hz), beta (13-30Hz), four gamma (30-50Hz, 50-70Hz, 70- 90Hz, 90Hz-). The upper limit of the last frequency bin was half of the sampling frequency. The power in different frequency bins was calculated using a function "bin_power" from an open-source python module named "pyeeg". (7)Power in a frequency bin is calculated by computing sum of absolute values of discrete Fourier transform in frequency ranges of a signal.

DFT is defined by formula:

$x_0, x_1, \ldots, x_{N-1}$ Is a sequence of N complex numbers and is transformed into a periodic sequence of complex numbers.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi k \frac{n}{N}} \qquad k = 0, \ldots, N -$$

Absolute value is square root of sum of square of real part and square of

imaginary part of each value of DFT series.

$$|X_k|/N = \sqrt{\mathrm{Re}(X_k)^2 + \mathrm{Im}(X_k)}$$

(8)

This feature was calculated for each 200 seconds window (after dividing original 10 min window into 3 chunks).

D. Feature Scaling

In case of methods using Support Vector Machines, feature array (for each feature) for each file was standardized using "preprocessing.scale" method from "scikit-learn" module. This means that feature arrays for each file had a zero mean and unit variance. (9)

E. Machine Learning Models

Machine Learning models were used from package "scikit-learn". All Machine Learning models used "class_weight=auto" because there is a difference in the number of interictal and preictal classes in training set and this parameter ensures that the classifier adjusts the class weights inversely proportional to class frequencies. (9)

In the case of first method (bivariate features), two machine learning models were used. The first was L2-regularized Logistic regression. The class label for 'preictal' is 1. The class label for 'interictal' is 0.

Logistic Regression computes the probability of target variable 'Y' (label) being 1 given an X (dataset) by the following formula:

The series 'w' represents the weights, which have to be learned by minimizing the loss function. In case of L2 regularization, we have to find vector W that minimizes the loss function and regularization term (sum of square of weights multiplied by regularization coefficient). This is done to prevent over fitting.

$$P(Y=1|X) = \frac{1}{1+\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

(10)

The second was a support vector machine with 'rbf' kernel. The cost was set to be 50. The class label for 'preictal' is 1. The class label for 'interictal' is -1. The cost was decided using cross validation and the 'C' values that were explored were multiples of 50 till 1000. Grid Search could have been used as an effective method to ascertain the optimal values of 'C' and 'gamma'.

In support vector machines, our aim is to maximize the margin between positive and negative training data sets and to find an optimal decision boundary.

The optimization problem in SVM is:

"W" represents the weights. The function below penalizes non-zero ξ_i. 'C' is the cost term and it represents how much is misclassification penalized.

$$\arg\min_{\mathbf{w},\xi,b} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \right\}$$

Subject to

$$y_i(\mathbf{w}\cdot\mathbf{x_i} - b) \geq 1 - \xi_i, \qquad \xi_i \geq 0$$

(11)

In the case of second method (uni-variate feature), a support vector machine with a 4 degree polynomial kernel was used. The cost was set to be 50. These values were received from a research paper written by Netoff, Park, and Karhi (2009). (3) The class label for 'preictal' is 1. The class label for 'interictal' is -1.The same machine learning model was used twice, once without preprocessing of signal and one with preprocessing of signal.

F. Post Processing

For methods in which a 10 minute window is divided into three 200 second windows, I have implemented a post processing method. The machine learning models are trained on three windows for each labelled file. When a machine learning model evaluates a file, the model returns predictions for three segments in a given file. If two of these three segments are predicted as 1.0, the model labels the whole segment as 1.0. Otherwise, the model labels the file as 0.0. The value of 1.0 for a test file means that there is a 100% probability that the file is preictal. The value of 0.0 for a test file means that there is a 0% probability that the file is preictal.

Results

The output from Logistic regression for each test segment was probability of the test segment being preictal.

The output from a support vector machine for each test file was either 1.0 or 0.0. This can be interpreted in terms of probabilities too. 1.0 represents a probability of a file being preictal as 100% and 0.0 represents 0%.

i) Using Bivariate features

The L2-regularized Logistic regression model returned an AUC of 0.60638.

The 'RBF' kernel SVM returned an AUC of 0.61904.

ii) Using Uni-variate feature

The 4 degree polynomial kernel SVM returned an AUC of 0.68881. This implementation was without preprocessing of signal before computing features.

After each channel of signal was preprocessed, the same model returned an AUC of 0.72690.

Conclusion

A patient-specific algorithm that uses a linear uni-variate feature such as power in different frequency bins with non-linear kernel support vector machine provides an AUC of approximately 0.73. Also, in addition to feature scaling, signal scaling should be applied. Dividing a 10 minute window into smaller chunks and performing post processing on these chunks is a useful methodology. The top score in Kaggle competition was 0.84. In the future, powerful uni-variate feature such as power in different frequency bins can be coupled with bivariate features to increase the AUC and predictive capacity of the model.

References

1. American Epilepsy Society Seizure Prediction Challenge." *Description - American Epilepsy Society Seizure Prediction Challenge*. Kaggle Inc, 25 Aug. 2014. Web. 12 Dec. 2014. <https://www.kaggle.com/c/seizure-prediction>.
2. Williamson, James R., Daniel W. Bliss, David W. Browne, and Jaishree T. Narayanan. "Seizure Prediction Using EEG Spatiotemporal

Correlation Structure." Epilepsy & Behavior 25.2 (2012): 230-38. Web.
3. Netoff, Theoden, Yun Park, and Keshab Parhi. "Seizure Prediction Using Cost-sensitive Support Vector Machine." IEEE Xplore. Seizure Prediction Using Cost-Sensitive Support Vector Machine, Sept. 2009. Web. 12 Dec. 2014. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5333711>.
4. Walt, Stefan Van Der, S. Chris Colbert, and Gael Varoquaux. "The NumPy Array: A Structure for Efficient Numerical Computation." Computing in Science & Engineering 13.2 (2011): 22-30. Web.
5. "Numpy.corrcoef¶." Numpy.corrcoef — NumPy V1.9 Manual. N.p., n.d. Web. 12 Dec. 2014. <http://docs.scipy.org/doc/numpy/reference/generated/numpy.corrcoef.html>.
6. "Eigenvalues and Eigenvectors." Wikipedia. Wikimedia Foundation, n.d. Web. 12 Dec. 2014. <http://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors>.
7. Bao, Forrest Sheng, Xin Liu, and Christina Zhang. "PyEEG: An Open Source Python Module for EEG/MEG Feature Extraction." Computational Intelligence and Neuroscience 2011 (2011): 1-7. Web.
8. "Discrete Fourier Transform." Wikipedia. Wikimedia Foundation, n.d. Web. 12 Dec. 2014. <http://en.wikipedia.org/wiki/Discrete_Fourier_transform>.
9. Pedregosa, Et Al. "Scikit-learn: Machine Learning in Python." Journal of Machine Learning Research 12 (2011): 2825-830. Web
10. "Logistic Regression." Wikipedia. Wikimedia Foundation, n.d. Web. 12 Dec. 2014. <http://en.wikipedia.org/wiki/Logistic_regression>.
11. "Support Vector Machine." Wikipedia. Wikimedia Foundation, n.d. Web. 12 Dec. 2014. <http://en.wikipedia.org/wiki/Support_vector_machine>.