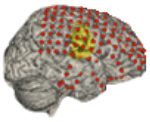


Completed • $8,000 • 202 teams

UPenn and Mayo Clinic's Seizure Detection Challenge

Mon 19 May 2014 – Tue 19 Aug 2014 (15 days ago)

Features for seizure detection

Yan Xu

Hello, everyone,

I'm wondering which features you have extracted for this dataset. I mainly have used the following features for each channel to achieve an accuracy of 0.9423:

fractal dimension, mobility, complexity, skewness, kurotsis, variance,

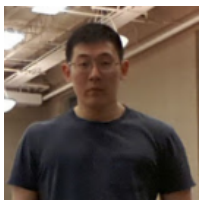
and frequency energy at following bands:

delta(0.5-4Hz), theta(4-7Hz), alpha(7-14Hz), beta(14-30Hz), gamma(30-100Hz)

Have fun~

Thanked by rbroberg , Damian Mingle and Triskelion

#1 / Posted 15 days ago Reply / Quote / Thank / Flag / Email User



Mike Kim

The following text probably won't be quite precise, but the intuition is I just threw a kitchen sink of feature transformations at the problem in the limited time I had (about 2-3? weeks) without thinking about it too much.

I tried various things at various windows and preprocessed with various filters not all of which was done within the box of conventional signal processing:

R's summary, variance, sd, mad, IQR, range, R (moment's package) skewness, kurtosis, geary, fft(Arg()) (half of it), mean, median filter, gaussian filter, correlation (upper tri, pearson, of the channels), lagged differences (diff) at various lags, various combinations of the above in different orders.

Then I ensembled gbm, brnn, glmnet, and randomForest. While brnn and glmnet did fairly poorly, these still helped overall. Everything was really slow since I did this all in R although multiple instances on AWS helped. Maybe some people had better ideas on handling the size of the data.

Thanked by Yan Xu , Damian Mingle and rbroberg

#2 / Posted 15 days ago

Reply / Quote / Thank / Flag / Email User

Serhii

I have used the next features:

maximal and minimal values for each channel;

standard deviation by channel;

I had an accuracy about 0.92 with Random Forest, KNN and Logistic Regression.

Then I've added:

logarithm of the sum of absolute values for each channel;

logarithm from the first 250 values of FFT for two channels

and improve accuracy to 0.94 .

Thanked by Yan Xu , Damian Mingle , rbroberg and Ruben Rybnik

#3 / Posted 14 days ago

Reply / Quote / Thank / Flag / Email User

Ruben Rybnik

Hi, I thought about hardware constrains and so I decided for 'cheap' time domain features, so I went with zero crossing rate (zcr) and average short time energy (aste). Extracting these features for all of the available channels and running them against a RF I've achieved a 0.89 score.

This was a fun competition.

PS: I like Serhii (min, Max, sd) approach :)

Thanked by Yan Xu and rbroberg

#4 / Posted 14 days ago

Reply / Quote / Thank / Flag / Email User

Senecaur

I hand-crafted features based on the following reference: D F Wulsin, J R Gupta, R Mani, J A Blanco and B Litt, Modeling electroencephalography waveforms with semi-supervised deep belief nets: fast classification and anomaly measurement, J. Neural Eng. 8 (2011) 036015

From Appendix B I extracted the following features for each EEG channel seperately:

-Normalized positive area under the curve

-Normalized decay: the chance-corrected fraction of data that is decreasing or increasing

-Frequency band power: the mean power spectral density in each of the frequency bands 1.5Hz-7Hz, 8Hz-15Hz and 16Hz-29Hz computed using Welch's method

-Line length: sum of the absolute differences between successive data samples

-Mean energy: mean of the square of samples across each channel

-Average peak amplitude: base-10 log of the mean-squared amplitude of each peak in the channel

-Average valley amplitude: as above but for the valleys

-Normalized peak number: number of peaks in the channel normalized by the mean difference between adjacent data points

-Zero crossings: subtracts the mean value (rather than the line of best fit as in the paper) from the channel and then counts how many times the zero-mean data crosses zero

Looking at the scores achieved above with smaller/simpler feature sets it seems that it was wasted effort to extract some of these. I plateaued very early on in this competition and couldn't figure out how to make any significant improvements to my score. I trained using Scikit ExtraTreesClassifier for both feature selection and classification.

Thanked by Yan Xu and rbroberg

#5 / Posted 14 days ago

Maineiac

This competition was definitely all about feature engineering. Using an ensemble of models (random forest, glmnet, bagged MARS, gbm) I got to 0.95+ with the following:

Measures of variability: Variance, max-min, 95th-5th percentile

Average Spectral Power (using wavelets): Delta (0-4 Hz); Theta (4-8); Alpha (8-14); Beta (15-30); lowGamma (30-100); highGamma (100-200)

Ratio of Spectral Powers to each other (e.g., theta / alpha; all combinations).

Right at the end, I added the mean and variance of cross-correlation values (i.e. lagged correlation ranging from 1-20 samples), which improved the accuracy of some of my individual models, but in the end, didn't improve the final ensemble.

No single model of mine had better prediction accuracy of 0.94, and some were as low as 0.87, so combining the predictions definitely helped here.

Thanked by Yan Xu , Damian Mingle and rbroberg

#6 / Posted 14 days ago

Joe Schmo

I mainly extracted frequency bins of about 3 Hz range and used a hamming window to improve the frequency resolution. Further marginal improvements came by splitting the time series into 'early' and 'late' pieces (crossover hamming windows) and extracting frequency content, as well as ratios between frequency bins.

I used a random forest model to get scores above .95. If I had used my best public leaderboard score for submission, that would have been good enough for ~ 6th place. I was concerned about overfitting so relied on a 5 fold CV for choosing the best model (in terms of reducing unnecessary features), but this yielded ~ .94 in both. Perhaps a leave-on-out CV would have been better?

I meant to spend more time at the end developing more features (both time and spectral) but didn't get around to it. Congrats to the winners! Fun competition!

Thanked by rbroberg

#7 / Posted 14 days ago

Yan Xu

I have evaluated the features individually with the label with AUC score described in the first post. I found that std, mobility, fractal dimensions and gamma are the most important ones. I have also included the 20%, 50%, 80% percentiles of cross correlation values, The attached is what I have got (the number in front indicates the channel):

Dog_1
Seizure top 10: ['4_std' '3_std' '7_std' 'percentile0.2' '2_std' '11_std' '15_std' '12_std''13_std' '9_std']
Early seizure top 10: ['15_beta' '11_mobility' '15_gamma' '7_gamma' '15_fractals' '3_gamma''7_fractals' '11_gamma' '3_fractals' '11_fractals']

Dog_2
Seizure top 10: ['14_mobility' '9_mobility' '14_gamma' '10_gamma' '7_mobility''11_mobility' '7_gamma' '15_mobility' '15_gamma' '10_mobility']
Early seizure top 10: ['12_fractals' '7_fractals' '11_fractals' '15_gamma' '14_gamma' '10_gamma''7_gamma' '10_fractals' '14_fractals' '15_fractals']

Dog_3
Seizure top 10: ['9_std' '2_std' '3_std' '5_std' '7_std' '14_std' '10_std' '6_std' '12_std''13_std']
Early seizure top 10:['11_std' '2_std' '5_std' '3_std' '10_std' '14_std' '12_std' '7_std''13_std' '6_std']

Dog_4
Seizure top 10: ['12_mobility' '7_std' '14_fractals' '12_complexity' '5_gamma' '5_fractals''6_fractals' '5_complexity' '5_mobility' '5_std']
Early seizure top 10: ['3_std' 'median' '15_std' '11_std' '13_std' '7_std' '4_std' '12_std''5_std' 'percentile0.2']

Patient_1
Seizure top 10: ['10_alpha' '9_fractals' '62_alpha' '19_fractals' '18_alpha' '13_alpha''13_fractals' '12_alpha' '20_alpha' '29_alpha']

Early seizure top 10: ['18_complexity' '19_mobility' '29_fractals' '20_complexity' '18_fractals''18_mobility' '29_gamma' '29_mobility' '19_complexity' '20_mobility']

Patient_2
Seizure top 10: ['2_std' '3_complexity' '1_complexity' '1_std' '2_beta' '1_beta' '3_beta''0_complexity' '0_std' '0_beta']
Early seizure top 10: ['0_std' '2_mobility' '3_complexity' '3_beta' '0_complexity' '0_beta''1_complexity' '2_complexity' '1_beta' '2_beta']

Patient_3
Seizure top 10: ['8_theta' '6_std' '5_delta' '8_beta' '5_beta' '4_alpha' '5_complexity''54_fractals' '4_delta' '4_complexity']
Early seizure top 10: ['5_skew' '5_alpha' '54_fractals' '5_complexity' '5_beta' '4_mobility''4_beta' '4_alpha' '4_delta' '4_complexity']

Patient_4
Seizure top 10: ['48_std' '33_gamma' '42_std' '34_std' '8_alpha' '56_fractals' '56_std''33_fractals' '33_std' '49_std']
Early seizure top 10: ['48_std' '33_gamma' '42_std' '34_std' '8_alpha' '56_fractals' '56_std''33_fractals' '33_std' '49_std']

Patient_5
Seizure top 10: ['7_mobility' '0_gamma' '0_complexity' '31_fractals' '15_mobility''1_fractals' '7_fractals' '0_mobility' '15_fractals' '0_fractals']
Early seizure top 10: ['1_mobility' '7_mobility' '15_fractals' '7_fractals' '0_gamma''15_mobility' '1_fractals' '0_complexity' '0_mobility' '0_fractals']

Patient_6
Seizure top 10: ['14_skew' '23_beta' '22_complexity' '15_alpha' '22_std' '14_delta''23_std' '23_complexity' '14_complexity' '14_alpha']
Early seizure top 10: ['22_complexity' '14_skew' '22_fractals' '14_complexity' '23_beta''23_complexity' '23_alpha' '14_alpha' '23_std' '22_std']

Patient_7
Seizure top 10: ['35_theta' '27_theta' '27_std' '31_fractals' '34_fractals' '26_fractals''30_fractals' '27_fractals' '7_std' '27_alpha']
Early seizure top 10: ['25_fractals' '12_alpha' '2_beta' '35_theta' 'percentile0.5' '12_fractals''23_gamma' '12_gamma' '1_gamma' 'percentile0.2']

Patient_8
Seizure top 10: ['9_std' '6_mobility' '5_mobility' '9_fractals' '14_mobility' '12_mobility''13_mobility' '11_fractals' '12_fractals' '10_fractals']
Early seizure top 10: ['9_beta' '13_delta' '13_complexity' '8_fractals' '14_mobility''15_mobility' '12_delta' '13_mobility' '12_complexity' '12_mobility']

Thanked by rbroberg

A quick summary of my model:

Resample to 500 sps. Extract 0.5 second windows from the beginning, middle, and end of each segment. Apply Hanning windows and compute DFTs. Sum the power in bands 4-8, 8-13, 13-30, and 30-100 Hz and convert to log scale. Discard all but 16 channels (I did this because I was short on time and didn't want to search for a better way to incorporate additional

Matthew Roos

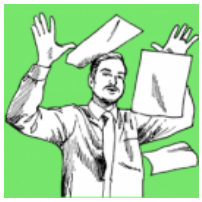
channels). The channels that provided the greatest d-prime discrimination of ictal vs. interictal were retained and ordered by their d-prime values. This resulted in a feature vector of length 3(times)x4(bands)x16(channels) = 192 for each segment. I used SVMs with RBFs and gamma = 1.58. One SVM for each of the predictions we needed to make.

Matt

Thanked by Serhii and rbroberg

#9 / Posted 14 days ago

Reply / Quote / Thank / Flag / Email User



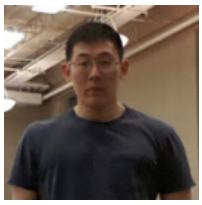
d00

We simply used 5 frequency bands per channel, and used pwelch() for power-spectral-density estimation, using tons of overlap within each segment and a small window size. Using those features alone, the tricky part, from our point of view, seemed to be get rid of all those "spikes" which occurred randomly, and which we interpreted as measurement error. We didn't have enough time to write a robust enough spike detector to improve our score significantly, w.r.t. not removing the spikes.

Thanked by rbroberg

#10 / Posted 14 days ago

Reply / Quote / Thank / Flag / Email User



Mike Kim

Can people elaborate on computational speed and methods they used in terms of implementation? The data set was fairly large compared to the median Kaggle contest.

#11 / Posted 14 days ago

Reply / Quote / Thank / Flag / Email User

rbroberg

For feature extraction, I used Julia on an Amazon c3.xlarge. Runtimes generally on the order of an hour or less, but I was only looking at fairly simple features.

#12 / Posted 14 days ago

Reply / Quote / Thank / Flag / Email User

Jonathan Street

For features I used median, variance, and extracted 6 frequency bands from an FFT. Classification was done using a Random Forest.

The way I handled the data import was quite inefficient. Reading in the data took the majority of the run time for building the model and generating predictions. I rented a EC2 r3.xlarge instance overnight to avoid spending time optimizing my implementation.

Thanked by rbroberg

#13 / Posted 14 days ago

Reply / Quote / Thank / Flag / Email User

The dataset was only a few gb after downsampling the clips to O(100) time units and storing

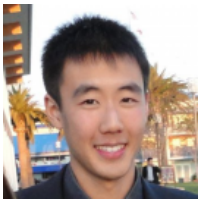
George Mohler

data as single precision (and it didn't seem like much information was lost doing this).

Thanked by rbroberg and Senecaur

#14 / Posted 14 days ago

Reply / Quote / Thank / Flag / Email User



DryRun

Mike Kim wrote:

> Can people elaborate on computational speed and methods they used in terms of implementation? The data set was fairly large compared to the median Kaggle contest.

Using joblib's Parallel function was a big help - I was able to run four subjects at once on my quad-core processor. I'm not sure what the exact speedup was, but it was certainly much faster, and pretty simple to implement.

Thanked by rbroberg

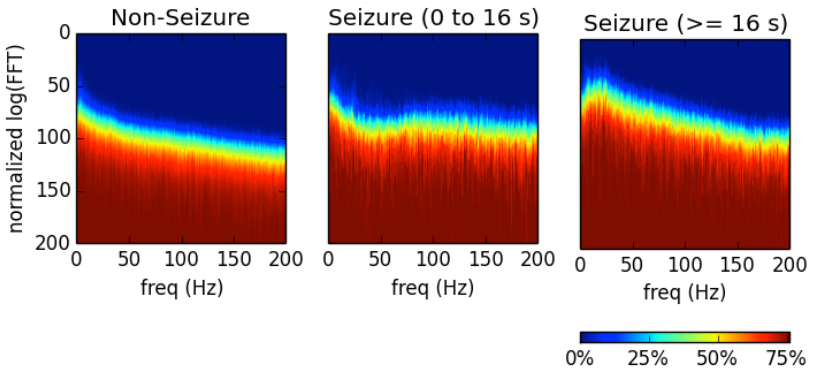
#15 / Posted 13 days ago

Reply / Quote / Thank / Flag / Email User

small yellow duck

I decided to avoid the whole problem of feature selection by turning the signals into images and feeding them to nolearn - the image classification neural net that comes pre-trained on a huge image database and which performed well on the Cats vs Dogs challenge. This strategy worked pretty well (92.1 on the final leaderboard), but not as well as the feature selection done by all you other clever folks.

A description of my strategy is here: **http://small-yellow-duck.github.io/seizure_detection.html**



Thanked by rbroberg , Senecaur and Jeong-Yoon Lee

#16 / Posted 13 days ago

Reply / Quote / Thank / Flag / Email User

This Challenge was a great learning experience for me, and the forum provided a great platform for interactive education.

I had very little time through and for the Challenge and very little computer resources. However, I had a desire to understand and contribute to this important topic (of seizures).

I derived mean, median, ..., temporal correlation (of one file with the next file), and spatial

Lalit

correlation (of one channel with the next channel) for each file. Then I used Random Forest in R. The score of about 0.58 was not impressive.

To improve the score, I started recomputing temporal correlation of a file's beginning part with the file's ending part. However, I could not finish this "experiment".

Thanks a lot to all of you for sharing your wisdom.

#17 / Posted 12 days ago

Reply / Quote / Thank / Flag / Email User

Maineiac

Mike Kim wrote:

> Can people elaborate on computational speed and methods they used in terms of implementation? The data set was fairly large compared to the median Kaggle contest.

I did all of my processing and modeling on my laptop: 16g ram, 4 core i7. Initial reading and pre-processing of the data took quite some time (i.e., 4+ hours), but once it was downsampled to 400 hz or so, things became more manageable. Once all the processing was done, my actual input files to my models were never larger than 5 mb or so. Some of the models I ran (e.g., bagged MARS) took 10+ hours to complete over all subjects.

I used R for just about everything. All of my feature creation was done within **data.table**. Modeling through **caret**, but I started to use **h2o** at the end, which has a more limited number of algorithms (for now), but is SUPER fast.

#18 / Posted 12 days ago

Reply / Quote / Thank / Flag / Email User

bbrinkm
Competition Admin

Thanks to everyone for participating in the competition and for your innovative work. The contest has been a tremendous success, and our congratulations to the winners.

First, we would like to announce our next competition on kaggle will be on seizure forecasting - identifying features in EEG that may indicate a seizure-permissive brain state, making it more likely a patient will have a seizure in the near future. This contest will begin soon, and will have a larger prize pool. It will use data clips in the same format as this contest, so many of the tools you developed for this contest could be directly applicable.

Second, we would invite all participants to log in to ieeg.org to continue the seizure detection effort on continuous data. Creating an account takes only a few minutes and allows you to save analyses and upload tools. We will also set up a forum and link area for github repositories for seizure detection. This is a great opportunity for experts in machine learning like yourselves to team up with epilepsy researchers to help advance our understanding of epilepsy.

Thanked by Matthew Roos , blaine and Lawrence Chernin

#19 / Posted 12 days ago

Reply / Quote / Thank / Flag / Email User

Yan Xu

Mike Kim wrote:

> Can people elaborate on computational speed and methods they used in terms of implementation? The data set was fairly large compared to the median Kaggle contest.

I processed the data on a workstation with two 3.47G 4-core CPUs with Python sklearn. Downsampling the data to 400Hz finishes in minutes and calculating features takes the most time. But it can be paralleled for different subjects and takes about two hours. Then the classification model only needs to load the feature tables every time and runs in parallel with ensemble methods. The classification takes about between 1 to 1.5 hours.

Thanked by Triskelion

#20 / Posted 12 days ago

Reply / Quote / Thank / Flag / Email User