Kaggle Melbourne University AES/MathWorks/NIH Seizure Prediction challenge - Winning Solution

Authors: * Alexandre Barachant * Andriy Temko * Feng Li * Gilberto Titericz Junior

Contents :

- Overview of the winning solution
- Models
  - Alex / Gilberto Models
  - Feng Models
  - Andriy Models
  - Whole team ensemble
- Cross-validation
- Reproduce the solution

Overview of the winning solution

The winning solution is a blend of 11 models created by the team members before they teamed up. All models were created subject-specific. No usage of test data and no cross-subject probabilistic tuning was performed. The blend is done using an average of ranked predictions of each individual models. The blend has been designed to reduce overfitting and improve robustness of the solution. To this end, we limited ourselves to the minimum of weight tuning, choosing a weight of 1 for all models.

Each model will be described in details below

Models

Alex and Gilberto Models

A total of 4 models were selected for the final ensemble (see table 1).

For all models, preprocessing consisted in segmentation of the 10 minutes segment into 30 non-overlapping 20 seconds segment. No filtering or artifact rejection was applied. After modeling, the maximum of the prediction of this 30 smaller time window was affected to the 10 minute segment.

Total training time (including feature extraction) is estimated to less than half a day for these 4 models. We used python, scikit-learn, pyRiemann, xgboost, mne-python and pandas.

Alex_Gilberto_relative_log_power_XGB.csv

Features : This dataset consist in the normalized log power in 6 different frequency band (0.1 - 4 ; 4- 8 ; 8 - 15 ; 15 - 30 ; 30 - 90 ; 90 - 170 Hz) and for each channel. Power spectral density was estimated using Welch's method (window of 512 sample, 25% overlap). PSD was averaged in each band, normalized by the total power before applying a logarithm. Total size of this dataset is 6 x 16 = 96 features.

Model : XGB, 10 bags

Alex_Gilberto_all_flat_dataset_XGB.csv

Features : This dataset include the relative log power dataset mentioned before with the addition of various measures including signal statistics (mean, min, max, variance, 90th and 10th percentiles), AR error coefficient (order 5), Petrosian and Higuchi fractal dimension and Hurst exponent. Total size of this dataset is 21 x 16 = 336 features

Model : XGB, 5 bags

Alex_Gilberto_autocorrmat_TS_XGB.csv

Features : For each channel, an auto-correlation matrix was estimated by concatenating time-delayed single channel signal before estimation of correlation matrix. Signal was downsample by 2 and 8 logarithmically spaced delays were used (0, 1, 2, 4, 8, 16, 32, 64). Each of the autocorrelation matrices were projected into their respective riemannian tangent space (see [1], this operation can be seen as a kernel operation that unfold the natural structure of symmetric and positive define matrices) and vectorized to produce a single feature vector of 36 item. Total size of this dataset was 36 x 16 = 576.

Model : XGB, 4 bags.

Alex_Gilberto_coherences_transposed_TS_XGB.csv

Features : This feature set is composed by cross-frequency coherence (in the same 6 sub-band as in the relative log power features) of each channels, i.e. the estimation of coherence is achieved between pairs of frequency of the same channel instead to be between pairs of channels for each frequency band. This produce set of 6x6 coherence matrices, that are then projected in their tangent space and vectorized. Total size of this dataset is 21 x 16 = 336.

Model : XGB, 10 bags.

Table 1.

Model name	Public Score	Private Score
Alex_Gilberto_all_flat_dataset_XGB.csv	0.77081	0.74481
Alex_Gilberto_relative_log_power_XGB.csv	0.80720	0.76908
Alex_Gilberto_autocorrmat_TS_XGB.csv	0.62951	0.69535

Model name	Public Score	Private Score
Alex_Gilberto_coherences_transposed_TS_XGB.csv	0.75004	0.76703
Ensemble	**0.77276**	**0.77439**

Models that did not make it into the ensemble

The difficulty (or impossibility) to build a reliable cross-validation procedure was making very difficult to select best performing feature set properly. Other features developed during this challenge include coherence, correlation, spectral edge frequency, cumulative power and peak frequency covariance. In addition to XGB modeling, logistic regression was used but led to slightly lower performance (depending on the feature set).

Remark

When slicing larger 10 min segment with a 20s time window, comes the question of how to optimally combine those predictions. We chose here to use the maximum probability to represent the probability of the 10 min segment. The rationale was that patterns predictive of seizures were likely not to be stationary during the whole 10 minutes. Post-competition analysis revealed that using mean of probability led to a significant decrease in performances (0.674 Public / 0.722 Private). Very interestingly, using standard deviation of probability led to an increase in performances (0.805 Public / 0.776 Private). This finding seems to validate our initial hypothesis. However, this observation only holds true for the 4 models described above.

Feng models

A total of 4 models were selected for the final ensemble (see table 2)

Preprocessing : The Butterworth filter (5th order with 0.1-180 HZ cutoff) was firstly applied to the raw data and then Ipartitioned the raw data into non-overlapping 30s windows(for GLM model, I used non-overlapping 50s windows). A combination of arithmetic mean of individual analysis windows was used to aggregate them into a single probability score for each 10-minute segment.

Total training time (including feature extraction) is estimated to less than 5 hours for these 4 models on my 8 GB RAM MacBook Pro .

Features :

- The standard deviation and average spectral power in 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 (70–180Hz)
- The correlation in time domain and frequency domain (upper triangle values of correlation matrices) with their eigenvalues.

Models :

1. Standard deviation and average spectral power were used in the XGB classifier and KNN classifier.
2. All features were used in Logistic Regression with L2 penalty(Ridge).

3. All features were used in the second KNN classifier.

Remark

I read relevant papers at the beginning of this competition and found one could generate thousands of features from the raw EEG. Because I don't have background of signal digital processing, I generated those features based on common features used in some important papers and my intuition. Too many noise features and correlated features would damage the performance of most classifiers.

Table 2.

Model name	Public Score	Private Score
Feng_xgb.csv	0.75909	0.77598
Feng_knn.csv	0.79631	0.74006
Feng_knnmorefeature.csv	0.75947	0.72670
Feng_glmmorefeature.csv	0.67852	0.73308
Ensemble	**0.80165**	**0.79044**

Andriy models

A total of 3 models were selected for the final ensemble (see table 3).

Preprocessing: for all models preprocessing consisted in a) demeaning the EEG signal, b) filtering of the EEG signal between 0.5 and 128 Hz with a notch filter set at 60Hz, c) downsampling to 256 Hz, d) segmentation of the 10 minutes segment into non-overlapping 30 seconds segment. After modeling, the maximum probability was taken to represent the probability of preictal for the whole 10m window.

Feature extraction: the features can be divided into two groups, per-channel feature (sometimes called univariate) and cross-channel features (multivariate). From each EEG channel, 111 feature were extracted from both time, frequency and information theory domain to capture energy, frequency, temporal and structural information and to form a generic description of the EEG signal. These features have been previously used in several EEG applications, including seizure detection in newborns and adults [2-3]. These include: peak frequency of spectrum, spectral edge frequency (80%, 90%, 95%), fine spectral log-filterbank energies in 2Hz width sub-bands (0-2Hz, 1-3Hz, ...30-32Hz), coarse log filterbank energies in delta, theta, alpha, beta, gamma frequency bands, normalised FBE in those sub-bands, wavelet energy, curve length, Number of maxima and minima, RMS amplitude, Hjorth parameters, Zero crossings (raw epoch, Δ, ΔΔ), Skewness, Kurtosis, Nonlinear energy, Variance (Δ, ΔΔ), Mean frequency, band-width, Shannon entropy, Singular value decomposition entropy, Fisher information, Spectral entropy, Autoregressive modelling error (model order 1-9). These led to 111*16=1776 features in a concatenated feature vector.

Apart from univariate measures, autoregressive modelling error (model order 1-9) was extracted from a single channel following common spatial filtering. The cross-channel features consisted of the following characteristics: lag of maximum cross correlation, correlation, brain asymmetry, brain synchrony index, coherence, and frequency of maximum coherence. These 6 features were extracted for the five conventional EEG subbands (delta, theta, alpha, beta, gamma) for 6 different montages (horizontal, vertical, diagonal, etc) leading to 180 features.

Both univariate and multivariate features form a pool of 1965 features.

Feature selection: for feature selection a pool of features was subjected to an XGB classifier from which an importance was computed and top N features were used for some models.

Modelling:

1) All features were used in a bagged XGB classifier (XGB).

2) Linear SVM was trained with top 300 features (SVM)

3) GLM was trained with top 200 features (glmnet)

Table 3.

Model name	Public Score	Private Score
Andriy_submission5_7_SVM.csv	0.73580	0.75369
Andriy_submissionLR5_3_glmnet.csv	0.71313	0.75856
Andriy_submissionXGB7_5mean.csv	0.77310	0.77237
Ensemble	**0.75774**	**0.78247**

Whole team Ensemble

Table 4.

Model name	Public Score	Private Score
Ensemble	**0.80630**	**0.80856**

comment: Due to some reproducibility issues discovered during code release, the score of the ensemble obtained from this code is different from the one submitted durring the competition. New score shows a slight increase in private score and slight decrease in public score.

CV method

Building a representative CV was one of the most challenging tasks in this competition, especially after the leak. We tried many CV approaches and most of them gave us too optimistic AUC scores.

Finally, our CV score is mainly composed of observing the score of the 2 approaches. Both of them were not perfect and served as an indicator whether to trust the LB or not. Both CV approaches are based on the integrity of 1-hour segments and only "safe" data were used.

2-Fold CV: The first fold is the first half interictal and the train preictal. The second fold is the second half of the interictal and the old test preictal. The problem of this approach is that we have time-based split of preictal data and no such split for interictal data. Also, as only 2 folds are used the results possess natural measurement noise.

26-Fold CV: Because we have 25 1-hour preictal segments for train per patient, we put each 1 hour to form first 25 folds, whereas as no sequence was provided for old test data, all of it was placed to the 26th fold. Interictal are split into 26 folds based similarly preserving the 1-hour sequence integrity and keeping the proportion of the preictal to interictal roughly equal in all folds.

References

> [1] Barachant, A., et al., Classification of covariance matrices using a Riemannian-based kernel for BCI applications, Neurocomputing 112 (2013): 172-178.
>
> [2] Temko, A., et al., EEG-based neonatal seizure detection with support vector machines, Clin. Neurophysiol. 122 (2011): 464-473.
>
> [3] Temko, A., et al., Performance assessment for EEG-based neonatal seizure detectors, Clin. Neurophysiol. 122 (2011): 474-82.

Reproduce the solution

The code corresponding to each group of model is available in separate folders. The solution can be reproduced in three steps :

- 1 : place the data in the `data` folder
- 2 : Go in each of the 3 following folders and follows instruction given in the `README` of each folder:
  - `Alex_Gilberto`
  - `Andriy`
  - `Feng`
- 3 : run `python make_blend.py` to blend the 11 different models.

Instruction for Hold-out evaluation

The best solution is to replace new_test data with hold out data using the same file s naming convention, and re-generate the solution from scratch following the three steps above.