https://tinyurl.com
/ntx-gbm-ML

University of California, San Diego
NeuroTech

ML Methods and Kaggle

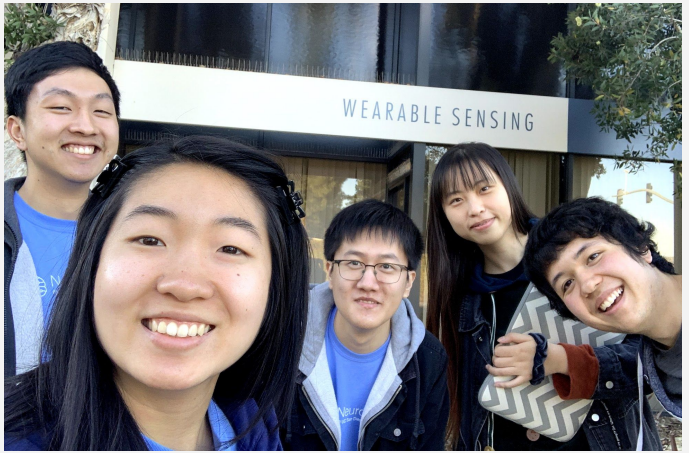
May 21, 2020

Slides by Zeyun Wu and Yundong Wang

About the club

- Affiliated with NeuroTechX
- Annual Student NTX Competition
- Faculty Advisor
  - Professor Vikash Gilja (TNEL Research Group)
- Connections with Wearable Sensing
- Multiple headsets for project groups next year!
  - OpenBCI
  - Wearable Sensing DSI-7

NEUROTECHX



Poll

- How much do you know about Machine Learning?
  - Checkmark: I am the master of Machine Learning algorithms
  - Hand: I have some experience with Machine Learning
  - X: Just heard of; not at all
- Have you attended a Kaggle Competition before?

Presentation Outline

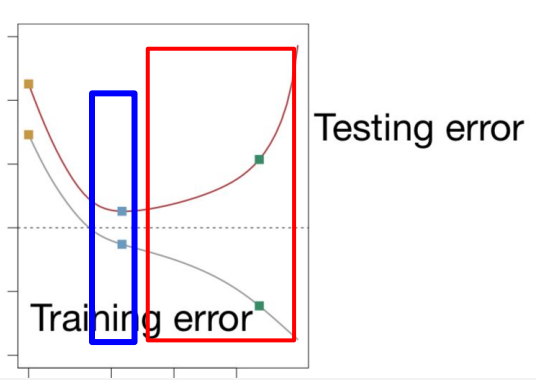
- EEG signals as time series data
- Machine Learning algorithms for EEG signal
- Deep Learning algorithms (essentially CNN) for EEG signal
- A Kaggle Project of BCI Challenge
  - ERP potentials (P300 speller)
  - EEG data epoching and feature extraction
  - Grid search and cross validation
  - Applying Machine Learning algorithms
  - Applying Deep Learning algorithms

EEG Signal as Time-series data

- Generally people use Machine Learning algorithms as a method to learn the feature of the data, make prediction or decision on the data
  - Clustering, prediction, classification, AND feature extraction!
- EEG signals are n by k vectors, where n small is channel numbers and k large is the time points: high dimensional
  - high dependency between data points: spatial and temporal
  - Need more feature engineering

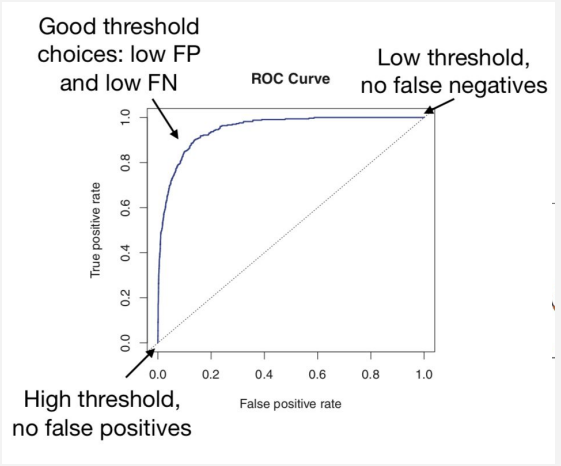
K-fold Cross-Validation

- Procedure
  - Randomly assign engineered features into k groups
  - Choose one “fold” as the validation set and use remaining (k-1) folds as training set
  - Repeat this for k times, using each fold for testing for once
- Always cross validate to avoid overfitting
- Use cross-validation for model selection



ROC and AUC

- ROC (Receiver Operating Characteristics Curve)
  - Measure a given model performance on false positive rate and false negative rate
- AUC (Area Under Curve)
  - Area under the ROC curve
  - Between 0 and 1
  - Represent the shape of ROC



SVM(Support Vector Machine):

Is a supervised machine learning models with associated learning that analyze data used for classification and regression analysis. A SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. (Wikipedia)

Penalty Parameter C: [0.001,0.1,1,10]

Kernel: linear; rbf; sigmoid

SVM

```
7]: ##SVM import packages
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import LeavePGroupsOut
from sklearn.model_selection import KFold
from sklearn import datasets, svm
####SVm with linear
###C list as [0.001,0.1,1,10]
###leave 4 groups out
clSvm = SVC(kernel = 'linear',probability = True)
p = {'kernel':('linear',), 'C':[0.001,0.1,1,10]}
clSvm= GridSearchCV(clSvm, p,cv = KFold(4),n_jobs =-1)
clSvm.fit(X_train, y_train )
y_hat_Svm = clSvm.predict_proba(X_test)
np.save('linear.npy',y_hat_Svm )
# print('ACC:'+ "{0:.3f}".format(accuracy_score(y_test, y_hat_Svm)))
```

Logistic Regression

is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

Binary logistic regression major assumptions:

At the center of the logistic regression analysis is the task estimating the log odds of an event. Mathematically, logistic regression estimates a multiple linear regression function defined as:

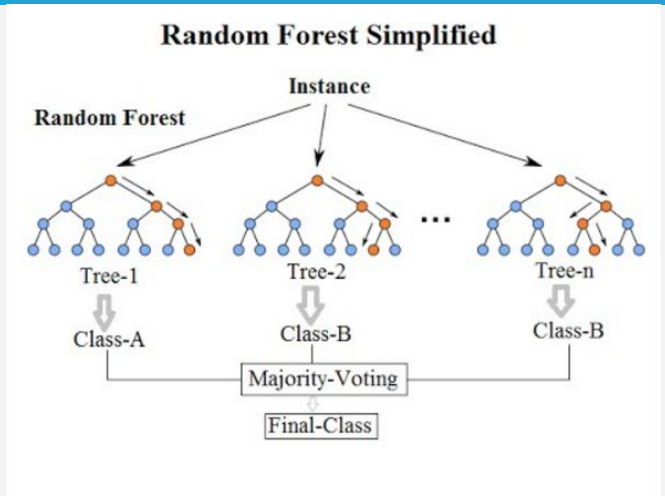
logit(p) $= \log\left(\frac{p(y=1)}{1-(p=1)}\right) = \beta_0 + \beta_1 \cdot x_{i2} + \beta_2 \cdot x_{i2} + \_ + \beta_p \cdot x_{in}$ for i = 1...n.

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X. \quad \longleftrightarrow \quad p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}.$$

https://www.statisticssolutions.com/what-is-logistic-regression/

Random Forest

Random Forest is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees. (Wikipedia)



Why prefer Random Forest rather than Decision Tree?

→ Reduction in overfitting: by averaging several trees, there is a significantly lower risk of overfitting.

→ Less variance: By using multiple trees, you can reduce the chance of stumbling across a classifier that does not perform well because of the relationship between the training and testing data.

Kaggle Competition on BCI

- Our code: https://github.com/YundongWang/COGS189_project_BCI_Challenge

Introduction & Motivation

- As humans think, we produce brain waves and these brain waves can be mapped to actual intentions. We only pay attention to visual stimuli from the brain wave data of people with goal of spelling a word.
- The “P300-Speller” is a well-known brain-computer interface (BCI) paradigm which uses EEG and the so-called P300 response evoked by rate and attended stimuli to select items displayed on a computer screen.
- The goal of this challenge is to evaluate whether the item selection was correct or not through the feedback from EEG cap.
- This decision could then be used to improve the BCI performance by implementing some error correction strategy.

Data Introduction

- Our data is collected at 200 Hz across 26 subjects (16 for training, 10 for testing). Each subject participated in 5 different sessions. The full data set is available at https://www.kaggle.com/c/inria-bci-challenge/data
- Each session include 60 target stimulus, however, the last session of each subject contains 100 target stimulus. Which makes 340 target stimulus for each subject
- 0 or 1 for bad or good feedback, respectively. Bad feedback is when the selected item is different from the expected item. Good feedback is when the selected item is similar to the expected item.

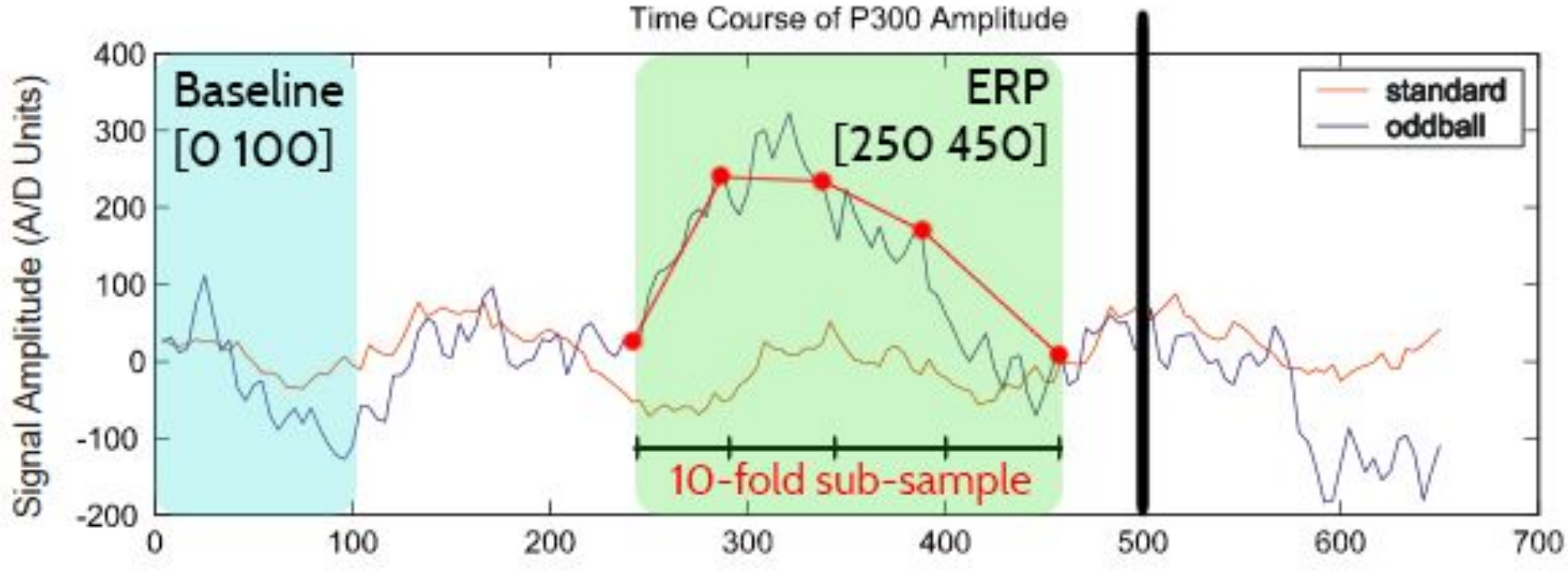
Pre-processing

- Step 1: Epoching + Cleaning
- Step 2: Feature Extraction

Step 1. Epoching + Cleaning

- What is Epoching?
  - Find stimulus time intervals
  - Apply the filter
- Goal of Epoching
  - Help to clean the data
  - Remove the EEG data area that is not necessary
  - Signals are epoched to take only 0.7 second after the feedback event.
  - Applied baseline correction with the signal 0 to 100 second before the feedback event.

Step 1. Epoching + Cleaning



Step 2. Feature Extraction

XDAWN Covariances Algorithm [1]:

- A dedicated Spatial filtering is applied on the data in order to increase the signal to noise ratio.
- Significantly reduces the dimension of the feature vector.
- Covariance matrices are used as feature.
- Applied 5 spatial filters. Transformed data from [340×N, 56, 140] to [340×N, 20, 20]. (N = 10 or 16).

$$\mathbf{Z}_i = \mathbf{W}^T \mathbf{X}_i \qquad \tilde{\mathbf{Z}}_i = \begin{bmatrix} \mathbf{W}^{(0)T}\mathbf{P}^{(0)} \\ \mathbf{W}^{(1)T}\mathbf{P}^{(1)} \\ \mathbf{Z}_i \end{bmatrix}. \qquad \mathbf{\Sigma}_i = \frac{1}{N}\tilde{\mathbf{Z}}_i\tilde{\mathbf{Z}}_i^T$$

Step 2. Feature Extraction

Tangent Space Algorithm [2], [3]:

- Reduced Covariances matrices are then projected in the tangent space.
- The Tangent space projection can be seen as a kernel operation.
  a. Minimum Distance to Riemannian Mean
  b. Tangent Space Mapping
- Useful to convert covariance matrices in euclidean vectors while conserving the inner structure of the manifold
- After the projection, data is transformed from [340×N, 20, 20] to [340×N, 210]. (N = # subject)
- Then, standard processing and vector-based classification can be applied.
- Apply both data preprocessing algorithms, our models had approximately 10% improvement on AUC score (expect EEGNET, where the original epoched dataset were fed).

Classifications we used:

1. SVM (Support Vector Machine)
2. Logistic Regression
3. Random Forest
4. EEGNet
5. StackNet

Cross Validation

- Leave-4-subject-out cross validation on 16 subjects in the training data.
- Grid search is used for finding the best parameters.
- Prediction is performed on the testing data for the 10 subjects.
- Area Under Receiver Operating Curve (AUC) tricks:
  - 1. Predict probability instead of categorical labels (0, 1)
  - 2. Transform categorical labels into soft labels using Elastic Net.
  - We implemented the first approach.
- The predicted probabilities (in CSV file) were uploaded to Kaggle for scoring.

EEGNET

- Architecture designed specifically for EEG data classification [4].
- First Conv layer to learn temporal filters, second Depthwise Conv layer to learn spatial filters, and third Separable Conv layer to learn a combination between temporal and spatial information.
- Data dimension: X_train.shape = (5440, 1, 56, 140), X_test.shape = (3440, 1, 56, 140).
- Configurations:
  - Channels = 56, Samples = 140
  - dropoutRate = 0.5, kernelLength = 100, F1 = 8, D = 2, F2 = 16.
  - Loss = sparse categorical cross entropy, optimizer = adam, metrics = accuracy.
  - Best parameters were saved using checkpoint for prediction.
  - Weighted loss: 1/7 for label 1, 1/3 for label 0.
  - Batch size = 32, epoch = 100

Related Works:

[1]. Rivet, B.; Souloumiac, A.; Attina, V.; Gibert, G., "xDAWN Algorithm to Enhance Evoked Potentials: Application to Brain–Computer Interface," IEEE Transactions on Biomedical Engineering, vol.56, no.8, pp.2035,2043, Aug. 2009

[2]. A. Barachant, S. Bonnet, M. Congedo and C. Jutten, “Multiclass Brain-Computer Interface Classification by Riemannian Geometry,” in IEEE Transactions on Biomedical Engineering, vol. 59, no. 4, p. 920-928, 2012. Pdf

[3]. A. Barachant, S. Bonnet, M. Congedo and C. Jutten, “Classification of covariance matrices using a Riemannian-based kernel for BCI applications“, in NeuroComputing, vol. 112, p. 172-178, 2013. Pdf

[4]. Lawhern, Vernon J. *EEGNet: A Compact Convolutional Network for EEG-based Brain-Computer Interfaces*, 23 Nov. 2016, arxiv.org/abs/1611.08024v4. Accessed 16 Mar. 2019.

Work Cited:

- Wikipedia Contributors, ‘Support-vector machine’, Wikipedia, The Free Encyclopedia, Wikimedia Foundation, 14 March, 2019. Web. 15 March, 2019, https://en.wikipedia.org/wiki/Support-vector_machine
- Wikipedia Contributors, ‘Decision tree’, Wikipedia, The Free Encyclopedia, Wikimedia Foundation, 15 February 2019. Web. 15 March, 2019, https://en.wikipedia.org/wiki/Wikipedia:Citing_Wikipedia
- Wikipedia Contributors, ‘Random forest’, Wikipedia, The Free Encyclopedia, Wikimedia Foundation, 25 February 2019. Web. 15 March, 2019, https://en.wikipedia.org/wiki/Random_forest