Stage 1. Machine Learning - when and why?

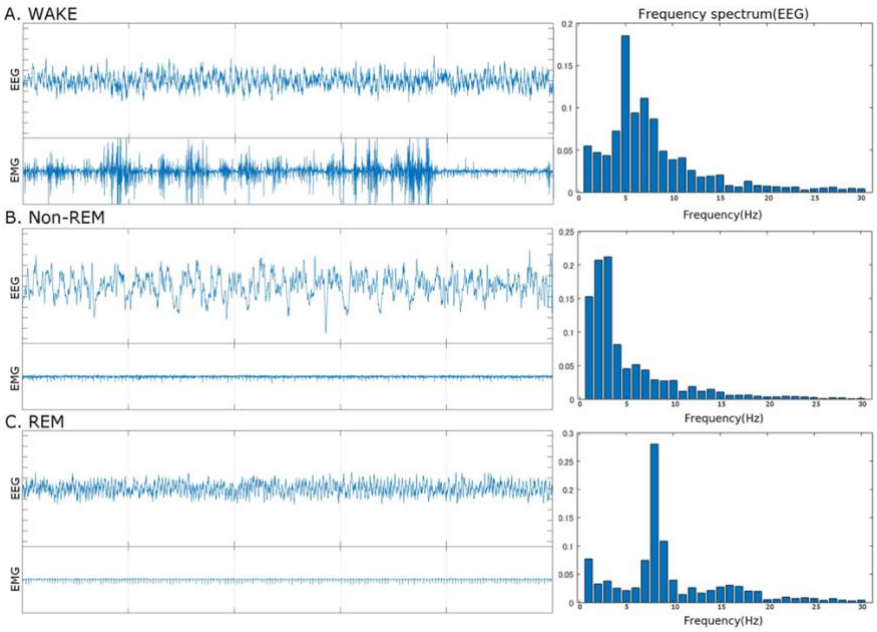
Explain the background (real-life scenario) of your ML application.

Project "Sleep stages classification of mouse ECoG signal"

Electrocorticography (ECoG), or intracranial electroencephalography (iEEG), is a type of electrophysiological monitoring that uses electrodes placed directly on the exposed surface of the brain to record electrical activity from the cerebral cortex. In contrast, conventional electroencephalography (EEG) electrodes monitor this activity from outside the skull [1]. ECoG measures voltage fluctuations resulting from ionic current within the neurons of the brain. ECoG recordings measure differences in electrical potentials between two points and are usually expressed in units of microvolts (μV).

In the current task we build a classifier for automatically assigning the brain activity state based on the EEG recording of the mice. We distinguish three brain states: non-rapid eye movement (NREM) sleep, rapid eye movement (REM) sleep and wake state.

Below you can see example ECoG traces corresponding to three brain states (stages), NREM, REM and wake. Wake stage is characterized by high frequency low amplitude ECoG activity, while NREM sleep is distinguished by low frequency high amplitude brain activity. REM sleep looks similar to WAKE stage from the first sight, but the ECoG trace has much more regular oscillations with low muscular activity (see EMG trace).
On the right side we see corresponding to signal (ECoG trace) spectrograms obtained by applying the Fourier transform (FT). FT maps the signal into a two-dimensional function of frequency and time [2]. We can see that each stage has distinct spectrogram: NREM sleep has highest power in the range of 1-4 Hz, REM sleep - in the range of 7-8 Hz and wakefulness is characterized by dominant 5-7 Hz activity.



Classifier is specifically meant for use by researchers in laboratory settings, where ECoG recording is routinely used in sleep research and to study the effect of pharmacological treatments on the brain activity.

Justify the need for ML approach for your problem.

Manual labelling ECoG traces is time consuming repetitive task, which requires sustained attention for prolonged time during labeling process. In addition, labeler required to have domain expertise as labeling based on visual inspection of ECoG traces. We also should mention possibility of discrepancy between labeling provided by different individuals.

Automating labeling process with ML based methods seems like an ideal solution. Even from the visual inspection of the ECoG traces and its' Fourier transform we can see differences between different states. Thus, we can hypothesize, that even simple ML classifier will be successful in separating these three brain states. Applying ML approach most probably will not require a lot of computational resources.

Describe requirement of your ML system. For example, what is most important for your application: very fast ML system with a small latency or maybe high precision?

This ML solution is meant for small-medium size research laboratories, which has its unique requirements. First, the data analyses usually done offline, meaning both, that analyses done after data is already collected and that ML system does not need internet connection. That means that robustness of the network infrastructure and fast performance is not of the main concern. Similarly, scalability is not something of importance as this type of ML solution is used by very niche user base (academic researchers), although in the case of large pharmacological company scaling the ML solution should be considered.

The main requirements for this type of ML system are reliability, good generalization property and interpretability. Reliability means that ML system should deliver consistent and correct (with high values of the chosen metrics - accuracy, F1 score, precision etc.) results and be robust to perturbations and outliers as biological data is often very noisy.
At the same time, we expect the system generalize well to other datasets of similar type. For example, if the model is trained on data collected from C57BL mouse line, it still should perform well on the ECoG recording of different mouse lines.
Interpretability is also important as understanding how model works can give unexpected insights of biological phenomena itself. For example, we can hypothesize that ML classifier will classify ECoG traces based on their time-frequency profiles, but it might be that ML model instead will discover some other hidden or latent features that we are unaware of.

Explain who are stakeholders* (company owners, investors, ML engineers, developers, users) of your project/company/application and what are their goals.

The main stakeholders are users (researchers in this case) and ML engineer. From the prospective of ML engineer, it is important to have accurate model with good generalization abilities, the goals shared between ML engineer and users. In addition, ML engineer would be happy to have fast training models, while for user the inference time is more important. Interpretability may not be the most important for ML engineer, but it is definitely useful for researcher who doesn't want to use "black box" algorithms.

References

[1] https://en.wikipedia.org/wiki/Electrocorticography
[2] https://en.wikipedia.org/wiki/Fourier_transform

Stage 2. ML problem formulation – DATA

Explain the source of the dataset. How did you collect (sample) data (you can make up some possible scenario for your type of data)? Is your sampling biased (e.g. due to non-probability sampling)? Discuss class (im)balance if applicable.

The dataset consists of approximately 4 hours of ECoG recordings of 34 animals, collected at Well Known University X. The ECoG signal (voltage value) is obtained as the difference between recording and reference electrode. The sampling rate of the ECoG signal is 200 Hz. ECoG signals were manually analyzed at 4 seconds time intervals and labels were assigned to each of these 4s time intervals (so called epochs).
Labels are codes (0,1,2,3,8) for different classes, where 0 is Wake state, 1 - NREM sleep, 2 - REM sleep, 3 - Artefact, 8 - not analyzed.
The ECoG data is stored in ' ECoG.npy' file, where we find recordings of 34 animals. ECoG recording is divided into 4000 x 4s intervals (4s = 800 data points at frequency 200 Hz), therefore the shape of the whole numpy array is (34,4000,800). The file ' ECoG_codes.npy' contains lables of the 4s epoches (shape of the numpy array is (34, 4000)).

Clearly explain the data points, features and labels of this ML problem. Indicate type of data (continuous variable, categorical or ordinal values etc.) and units of measurement when applicable.

The datapoint is one epoch (4 second) of ECoG trace, collected at 200Hz sampling rate. The features of the datapoints are voltage values in microvolts (μV). We have 4s*200Hz=800 samples per epoch, meaning that feature vector of the datapoint consist of 800 floating-point numbers.
Each epoch has corresponding categorical label: 0 is Wake state, 1 - NREM sleep, 2 - REM sleep, 3 - Artefact, 8 - not analyzed.

Explain your feature selection process (no theoretical justification needed). Do you need to continuously collect (update) data, or do you use static dataset?

We choose to divide ECoG trace into 4s epochs and thus generated 800-dimentional feature vectors, based on domain expertise and practical considerations. Shorter epochs might be more difficult to label (e.g. contain transition states, e.g. Wake → NREM or artifacts) and will require more time and labor (need to label more epochs). Too long epochs may contain several brain states, making it impossible to assign one label per epoch. Thus, current epoch duration was chosen empirically after visual inspection of the recorded signal.

Stage 3. ML problem formulation – MODEL and LOSS

State the number of datapoints, briefly describe the dataset and/or any data preprocessing or cleaning needed. If using categorical or ordinal variables, explain how you encode them.

The dataset consists of 34*4000 = 136k datapoints. The feature vector is 800-dimension vector of real numbers. Labels are categorical and encoded as integer numbers: 0 is Wake state, 1 - NREM sleep, 2 - REM sleep, 3 - Artefact, 8 - not analyzed. Feature matrix and labels are stored in a form of numpy arrays of shapes (34,4000,800) and (34,4000) respectively.

As we are only interested in three stages NREM, REM and Wake, we need to handle two other cases: 3 - Artefact, 8 - not analyzed. We will filter out all unlabeled epochs (8 - not analyzed) but will include epochs labeled as artefacts as a separate class in addition to NREM, REM and Wake.

ECoG signal collected from each animal is unique and the quality of the recording is dependent on many factors. Therefore, even the values range of ECoG signal may differ from animal to animal and we will use feature scaling to bring all feature vectors to the same scale. We will try out different scaling methods (e.g. min-max normalization, mean normalization) to find which pre-processing step is the best one. The sklearn scaling function will be first applied (fit) to training set and then this scaler is used for scaling validation set. This is done to avoid data leakage [1].

Before using classification models, we will apply fourier transform to get power-frequency information from ECoG signal. This feature engineering technique is commonly used for EEG and ECoG data analyses. We will use Python scipy.signal to perform fast fourier transform with Hanning window of size 256. Thus, the size of the transformed feature vector is 256. Low frequencies (below 1.5-2 Hz) will be filtered out with high-pass filter as they usually represent moving artefacts.

Describe and explain (why?) your choice of ML model(s)/hypothesis space(s), e.g., linear predictors, etc.

Transformed features will be used to fit following models:

- support vector machine (SVM)
- Decision tree (DT)

The hypothesis space of the SVM is a linear hypothesis space, same as linear and logistic regression. What makes SVM different is the use of kernel map to construct (typically high-dimensional) features and then using these "new" features to separate datapoints [2]. We chose SVM as it handles high-dimensional data well and we have feature space dimension in order of hundreds. In addition, it is very flexible and allows the use of different kernels, which in turn allows to create linear and non-linear decision boundaries.

DT is a non-parametric supervised learning method used for classification and regression. The hypothesis space of the DT is piecewise-constant over regions of the feature space [3]. DT are easier to interpret and visualize then some other powerful classification models (e.g. artificial neural networks), can handle numeric and categorical data, and the cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree.

Describe and explain (why?) your choice of loss function(s).

SVM classifier optimizes following problem [4]. This allows to maximize the margin, while incurring a penalty when a sample is misclassified or within the margin boundary. Linear SVM can be formulated as [5], which is augmented hinge loss. Sklearn implementation is based on libsvm library and provides set of kernels such as 'linear', 'poly', 'rbf', 'sigmoid' [6].

DT use the impurity (Gini coefficient) of individual decision region for categorical labels (classification) as a loss function. Gini coefficient allows to assess the quality of a candidate split of node. By minimizing impurity of the node DT model can create a tree (i.e. split feature space) where datapoints with a same label are grouped together ("pure" node). Other loss functions provided by sklrean implementation of DT are "entropy" and "log_loss" [7].

Explain the process of model validation - how did you split the data into training, validation and test sets. What are the sizes of each set and why did you make such design choice.

The size of the dataset is 136k samples. We will divide it into training-validation and test sets. We will use `train_test_split()` sklearn function with shuffling and stratified sampling enabled [8]. Stratified sampling is used to account for class imbalance. There are no strict rules on the data split ratio, but we choose 80/20, which is a common choice used in practice.
Training-validation set will be used for model selection and hyperparameter tuning, while test set is preserved for final model evaluation.

If applicable, describe and explain your use of metrics (e.g. accuracy) in addition to loss function (e.g. logistic loss).

In addition to loss values, we use metrics commonly used for classification problems - accuracy, precision, recall and F1 score [9]. Accuracy is a proportion of correctly classified

datapoints and is the most common metric for classification problems. Precision and recall indicate at what extent model assigns wrong label to the class (false positive) and how many sample belonging to the class are missed (false negative). These measures are important to evaluate especially in the cases of imbalanced datasets, as the class with smaller sample size tend to have low recall value, thus sample weights may need to be adjusted. F1 score is the harmonic mean of precision and recall, it symmetrically represents both precision and recall in one metric [10].

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$

Precision = $\frac{TP}{TP+FP}$

Recall = $\frac{TP}{TP+FN}$

F1 = $\frac{2TP}{2TP+FP+FN}$

References

[1] https://scikit-learn.org/stable/common_pitfalls.html#common-pitfalls-and-recommended-practices
[2]https://github.com/alexjungaalto/MachineLearningTheBasics/blob/master/MLBasicsBook.pdf, chapter 3.7
[3]https://github.com/alexjungaalto/MachineLearningTheBasics/blob/master/MLBasicsBook.pdf, chapter 3.10
[4] https://scikit-learn.org/stable/modules/svm.html#svc
[5] https://scikit-learn.org/stable/modules/svm.html#linearsvc
[6] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC
[7] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
[8] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
[9] https://scikit-learn.org/stable/modules/model_evaluation.html#the-scoring-parameter-defining-model-evaluation-rules
[10] https://en.wikipedia.org/wiki/F-score