Introduction to Machine Learning: Autumn Semester 2015
Instructor: Prof. Joachim M. Buhmann

Course Project 2

Sleeping states classification based on EEG and EMG data

1 Introduction

You have seen in class what classification is and a traditional approach to the problem, bayes classifier. In this task, you are asked to build a classifier in order to predict the sleeping stages of mice given some features extracted from their electroencephalography (EEG) and the electromyography (EMG).

A way to analyze the brain is through the measurements mentioned above. Electroencephalography records the brain's spontaneous electrical activity over a period of time, namely taking the measurements from multiple electrodes placed on the scalp. Electromyography consists on the recording the electrical activity from skeletal muscles, in this context we look at the neck muscles.

In diagnostic applications, such as the task that has be designed for this project, they generally focus on the spectral content of EEG and EMG, that is, the type of neural oscillations (usually called "brain waves") that can be observed in EEG signals.

Sleep staging, i.e. the identification of stages/states of sleep, is an unavoidable step in sleep research and it requires visual inspection of EEG and EMG data. The importance of machine learning techniques in this area is exactly due to this visual inspection, in fact this operation is biased and prone to error by humans and thus it turns out to be the biggest bottleneck for large-scale sleep research.

To be able to properly model the classes you need to experiment with different classification techniques and maybe to transform the feature space in order to achieve better performance.

The project is hosted on Kaggle and you get invited using this link: https://kaggle.com/join/5XuFneF8VrQtn4u3.

2 Data set description

2.1 Input

The raw data we deal with are EEG and EMG data recorded on mice. We split these records in 4 seconds long intervals and your task is to classify those intervals as belonging to one of the following three sleeping stages: wake, REM (Rapid Eyes Movement sleep), NREM (Non Rapid Eyes Movement sleep).

As said in the introduction, for diagnostic purposes, we care about the spectral contents of EEG and EMG data. More specifically what we are interested in the power of the signal in specific windows of frequencies. This frequencies windows caracterize what in the medical field they refer to as brain waves, such as delta, theta, alpha waves. So, considered one temporal sample we compute the its fast fourier transform (FFT) obtaining this way the frequency representation of the sample. Then we split the so obtained signal in different frequency bands and we compute the power of the signal over these bands.

As often done in the literature we consider 7 input features. The first 6 features are nothing but the spectral power of the signal for certain frequency bands retrieved from the EEG, while the 7th feature is the power of a

specific band of frequencies from the EMG. The specific bands are specified in Table 1.

Feature	Frequancy band
$feat_1$	0.49 Hz $< freq_{EEG} \leq$ 5 Hz
$feat_2$	5 Hz $< freq_{EEG} \leq$ 9 Hz
$feat_3$	9 Hz $< freq_{EEG} \leq$ 15 Hz
$feat_4$	15 Hz $< freq_{EEG} \leq$ 23 Hz
$feat_5$	23 Hz $< freq_{EEG} \leq$ 32 Hz
$feat_6$	32 Hz $< freq_{EEG} \leq$ 64 Hz
$feat_7$	4 Hz $< freq_{EMG} \leq$ 40 Hz

Table 1: Frequency windows over which the power is computed in order to obtain the considered seven features.

Sleeping stages are mapped as follows: wake-0, REM-1, NREM-2.

Even though essential and interesting in itself, the features engineering is not the focus of this project, thus all said above has been performed in order to provide you directly the 7 input features. Nonetheless, those groups that feel already confident with ML are encouraged to send an email to David and Julian in order to have all the necessary files. Any successful solution will be considered as a bonus at the moment of the project evaluation.

Note 1: The meaning of each feature is not important for performing well at this task. You can obtain the perfect score without knowing anything about neuroscience.

2.2 Output

You are asked to build a model that predicts the sleeping stage of each provided sample. Again, the sleeping stage s has to be 0, 1 or 2.

2.3 Training Set

This data is formatted as a comma-separated values (CSV) file in which each line corresponds to an observation. Each observation consists of 9 values: an unique id of the data set, 7 features (in the same order as they have been introduced above) followed by the sleeping stage s. Each line has the following format: id, $feat_1$, $feat_2$, $feat_3$, $feat_4$, $feat_5$, $feat_6$, $feat_7$, sleeping stage (the label). The training set is in the file "training.csv".

2.4 Validation and Test Sets

Both validation and test set contain stages that have not been labeled yet. Your task is to predict the delay for a given feature vector. You will be given several samples with their respective feature vectors that describe the sleeping stages and that provide the information neccessary to perform automatic calssification and you are asked to predict the sleeping stage for each of those samples. The data sets are given in the file "validate_and_test.csv". (Often the validation and test set are provided separately. As it was for the first project, since Kaggle has only a single submission, the validation and test set are combined here.)

The formatting of the prediction file is as follows:

- Same line format as the training set except that the sleeping stage is not given (each line has only the unique sample id and the 7 comma-separated features).

- **Required output:** a file that contains the predictions. Make sure to use the same ids as in the "validate_and_test.csv" file and column names as shown in the "example_solution_handin.csv" file. Kaggle won't be able to score your submission if you forget the column headers!

After making the submission Kaggle will compare your solution to the ground truth and compute the error. From the submitted data you will get only feedback on half of your predictions. This feedback corresponds to the validation data set and will be listed in Kaggle's "public leaderboard". The final score and grading will be made from the other half of your prediction in a "private leaderboard" (this corresponds to the test data set). During the competiton, you will see only the results in the "public leaderboard".

To generate the submission file the following code snippet might be helpful. It shows you how to write your prediction with the necessary column headers to a file:

```
% IMPORTANT: Make sure to have the sleeping stage as column vector.
labels = [2;0]; %<<< Compute your prediction here
ids = [1;2];        %<<< Use the ids from the prediction file

% Generate a single array with the Ids and delays.
res = horzcat(ids, labels);

% Convert the array to a table and add column headers.
table = array2table(res, 'VariableNames', {'Id', 'Label'});

% Write the obtained table to a CSV file.
writetable(table, 'your_prediction.csv');
```

3 Evaluation and Grading

Each submission (upload of a prediction file for a given data set) will be ranked according to the Categorisation Accuracy (see "http://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html") of the predictions. Since we have the measured actual stage for each sample in the validation set and test set we can calculate the response.

We compare the cost of the submission to a baseline accuracy which is 0.90. Hitting this baseline corresponds to a grade of 4. The required score to get a 6 will be determined after the competition has finished.

3.1 Report handin

In addition to your predictions on the test set you need to provide a brief report that explains how you obtained your results. One report per team is enough.

We include a template for LATEX in the file "report.tex". If you do not want to use LATEX, please use the same sections as shown in "report.pdf". Upload a zip file with the report (as a PDF file) along with your code or parameters/screenshots of the tools you used. For further instructions refer to the report template.

We might ask you to show us what you did, so please keep the necessary files until the end of the semester.

3.2 Deadline

You will be able to submit predictions starting from **Monday, 26.10.2015** until **Friday, 20.11.2015, 23:59:59 UTC**.

4 Questions

If you have questions regarding the project, please ask during the tutorial next week. Afterwards contact Julian Viereck (jviereck@student.ethz.ch) and David Tedaldi (dtedaldi@student.ethz.ch) via email and use "[ML]" in your email subject. For example "[ML] Question about ...".