QUALITY ASSESSMENT OF EEG RECORDINGS

Jules Belveze, David Mirabet, Marina Pons

Final report of the course *02460 - Advanced Machine Learning*
at DTU - Technical University of Denmark

ABSTRACT

EEG recordings from 31 different patients are analysed throughout this work. By labelling the data as good or bad we are able to perform supervised machine learning to classify the recordings. Different techniques are used to process the data and achieve the best possible accuracy of the classifier. Among these methods we use feature extraction, PCA, Outlier removal, etc. Finally we use SVM and we try to increase the accuracy by balancing the data. The accuracy reached using SVM, down-sampling and the whole set of features is: 80.4 ± 0.05 % (with a 50 % baseline).

1. INTRODUCTION

The electroencephalogram (EEG) is the recording of electrical activity occurring inside the brain. Diagnosis of neural diseases based on EEG signals is quite a new field that has gained much attention in the past years. Nevertheless, this type of data has always been difficult to analyze and interpret. Thus, more and more scientists and data analysts are starting to work with different libraries and tools in order to manage it and to be able to convert it into something actually understandable and useful. Python is starting to gather some libraries that help computer scientists to analyse those records. The main objective of this project is to given EEG data, try to find some patterns while applying some classification tools, that can help in some manner to make a faster and easier diagnosis of brain diseases in under developed countries. In order to approach the available data, our starting point is to define what is a satisfying recording and from that apply some methods to achieve a good classifier of good and bad recordings.

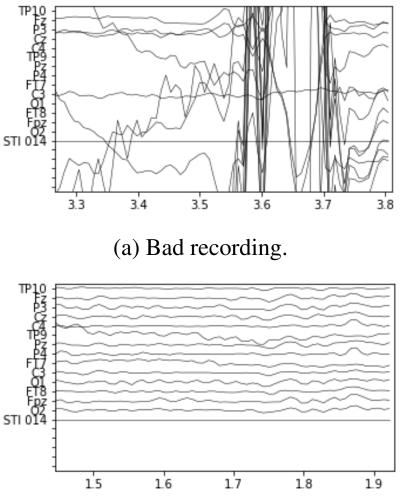
This paper is organized as follows: Section 2 stands for methodology including labelling, feature extraction, correlation analysis, PCA and classification. In Section 3 we present our results from the PCA and the SVM classification. Afterwards, in Section 4 we discuss our results and in Section 5 we extract our final conclusions.

2. METHODOLOGY

In this section we explain the different steps that we followed to achieve the main objective of our research. First labelling, feature extraction, study of correlated features and outlier removal using Local Outlier Factor (LOF), Principal Component Analysis of the whole data set and for each class (good or bad recordings), data balancing and finally classification using support vector machine.

2.1. Data characteristics and labelling

The data set consists of EEG recordings from 31 patients. They were recorded in Bhutan with a portable real-time neuroimaging system [1, 2]. Our aim was to determine in a fast way if a recording is good enough to be analyzed by a doctor or not. In order to approach this problem we divided the data for each patient in chunks of three minutes long, using a Python library called *MNE* [3]. Hence we moved away from real-time and focused on giving a solution to this issue. Once we did that, we ended up with a data set of 410 observations.



(a) Bad recording.

(b) Good recording.

Fig. 1: Example of the criterion that we used to label the different recordings.

We labelled the chunks based on the criterion in Fig.1. The results were: 240 recordings labeled as bad and 170 as good. Those labels are our baseline to build and test our classification model.

2.2. Feature extraction

We used the *pyeeg* library from Python to extract the most relevant features of the recordings. Basing ourselves on [4, 5] we extracted different type of features in the frequency domain (e.g. the Wavelet transform). Moreover, we obtained some features related to the characteristics of the time series (e.g. mobility, complexity, skweness, kurtosis). Therefore, we ended up with many features computed per channel, and others that were computed as the mean of the 14 channels. However, we will find out in 3.2 that computing all the features per channel gives a better accuracy.

2.3. Correlation matrix

There are some features that represent similar characteristics about the recordings, thus, we compute the correlation matrix to reduce the dimension of the features set. Two features with an absolute value of Pearsons correlation coefficient higher than 0,7 are considered highly correlated. Consequently, the feature in the pair which is more correlated with other features is removed from the set. Once we do that, we keep 86 features over the initial 182. This procedure leads to faster model computing and enhances classification reliability and pattern recognition within PCA.

2.4. Outlier detection with k-nearest neighbours

We also observed the presence of extreme outliers that hindered the PCA. Therefore, we performed unsupervised outlier detection using Local Outlier Factor (LOF) [6] in the features data set. LOF computes the local density deviation of a given data point with respect to its neighbors. This local density deviation is estimated by the k-nearest neighbors algorithm. In our case, the algorithm considers as outlier the data point that has substantially lower density than their 5 nearest neighbors. These detected outliers have been taken into account in the ongoing study. In PCA, they have been removed to avoid their significant effect in the variance explained by principal components.

2.5. Principal Component Analysis (PCA)

Two different approaches for PCA were tackled. The first one consisted of computing the singular-value decomposition of the complete feature data set while in the second one the data set was grouped by recording label, which is good or bad label. The idea behind this second approach arises from the hypothesis that good and bad recordings could be explained by distinct features. In both cases, the principal components are constructed as linear combination of the initial features. These combinations are linearly independent and most of the information within the initial variables is squeezed or compressed into the first principal components [7]. Thus, the percentage of variance that every component accounts for and the contribution of the initial features to these components are calculated and plotted for interpretation and possible dimension reduction of the problem.

2.6. Classification using SVM

First of all, before carrying classification out, due to the small amount of observations and the fact that our data set is imbalanced we decided to apply both *Synthetic Minority Oversampling Technique* (SMOTE) to generate new points and down-sampling in order to equipoise the classes. To classify our recordings we decided to try out a powerful and flexible model: support vector machine, which can be applied to both linear and non linear problems. In order to validate our model and to come up with the highest possible accuracy we have used nested cross-validation to tune the hyperparameters like: the kernel function, the penalty of the error term, etc... The model was in each case tried with the whole set of features and a set of features excluding the correlated ones. To assess the performance of our classifier we have used the classification report provided by *Scikit-learn* which gathers the following metrics: precision, recall and $f1$-score.

3. RESULTS

3.1. PCA results

Applying the correlation matrix to the data set led to reduce its dimension from 182 to 86 in the features axis. We have 410 observations (as we said before), so PCA is applied to a 410x86 matrix.

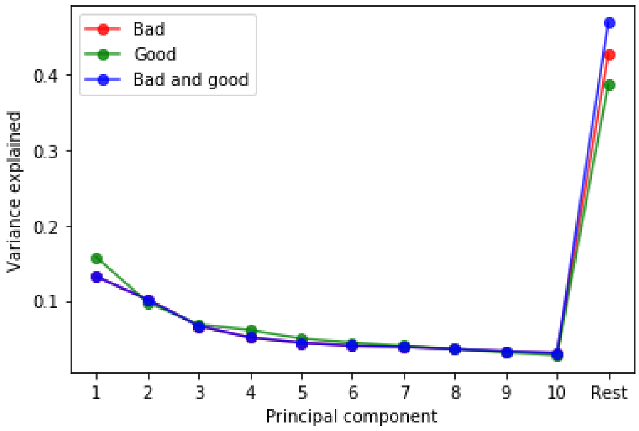


Fig. 2: Scree plot of the variance explained by principal components, for the bad class, the good and for both. We present at the end the cumulative variance for the last 76 PC.

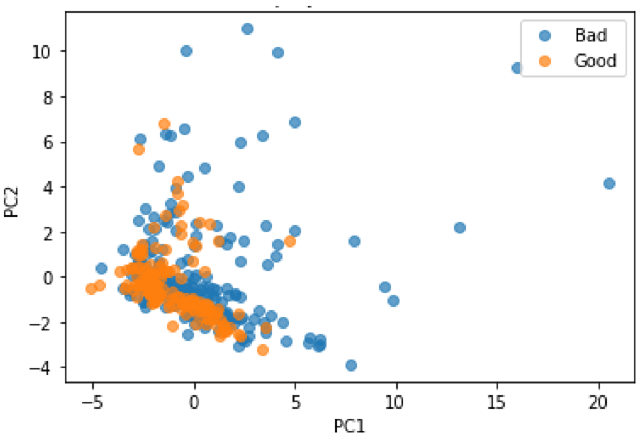


Fig. 3: EEG features data set projected onto the first two principal components and grouped by class.

In Fig.2 we show the variance explained by the 10 principal components of this new feature set. The first component only accounts for the 13,15% variance and the second for the 9,31% while components from 11 to 86 account for a 46,84%. In the case of the ones grouped by class data set, the results show a similar distribution of the variance explained. Moreover, in Fig.3 the data has been projected into the first and second principal component and grouped by class. There is not a clear pattern in the two components that makes the two classes distinguishable.

3.2. SVM classification

It turned out that for each case using the whole set of features, there was an improvement in the accuracy of between 1.1% and 7%. Thus, from now on, we will only work with the complete set of features. Combining both SMOTE upsampling and SVM classifier leads us to 70.7% of accuracy when training the classifier on 70% of the resulting ”new” data set. When using the imbalanced data set (that is constituted of 58% of bad recordings) the SVM classifier is 71.5% accurate but with a baseline of 58%. The highest accuracy we obtained, 80.4% was reached by using down-sampling before applying the model.

	precision	recall	f1-score	support
bad	0.77	0.80	0.79	46
good	0.83	0.80	0.82	56

Table 1: Metrics for the SVM classifier using downsampling

	precision	recall	f1-score	support
bad	0.69	0.83	0.76	65
good	0.76	0.59	0.66	58

Table 2: Metrics for the SVM classifier with imbalanced classes

	precision	recall	f1-score	support
bad	0.75	0.66	0.70	65
good	0.67	0.76	0.71	58

Table 3: Metrics for the SVM classifier using SMOTE

In Tab.1, Tab.2 and Tab.3 we present the different classification metrics for the three different combinations of SVM and balancing data technique that we used.

4. DISCUSSION

4.1. PCA

First, PCA revealed that the variance of the new feature set is scattered among the 86 components as shown in Fig.1. The pattern is mainly flat, the first two principal components don't even reach to explain at least 25% of the variance and the first 10 principal components accounts for a similar amount of variance as the rest of components. This often occurs in high dimensional data sets where the information is scattered among many features. Since this occurs, we cannot rely on a set of principal components to use in further analysis. To reach a 90% of variance explained we would have to take the fist 72 principal components which is comparable to the 86 original features but loosing some information. Hence, we decided not to remove any principal components as we would be loosing information.

On the other hand, we performed PC grouping the data by class to see if we can obtain any relevant information. Intuitively and looking at the recording plots, it seemed that recordings that belong to the good class would have more characteristics in common inside their group while bad class recordings would have subclasses of bad recordings. Contrary to our hypothesis, good and bad recordings behave similarly in terms of the variance decomposition. However, in Fig.3, it can be observed that the bad class is scattered to a certain extent in the first two principal component plane while the good class is much more concentrated in a particular region, which partly supports the previous hypothesis.

Focusing again on Fig.3, we can identify a pattern for both classes where the major density of samples seems to follow a negative slope line. However, it is difficult to interpret since the data is high dimensional. Good and bad class are partly overlapped which can be one of the reasons for miss classification. As previously commented, bad class samples are more prone to escape from their density cluster than good ones which makes their classification easier.

4.2. Classification

As shown in section 3.2 the most accurate model is the one using the complete set of features and down-sampling. The aim of SMOTE is not to affect the classifier's accuracy or f1-score [8] (which might be not relevant in the presence of unbalanced classes) but it is mostly about focusing on the trade-off between precision and recall. Since the main goal here is to build an accurate model we will not keep SMOTE for our final model. Down-sampling is more effective than SMOTE for most of the classifiers [9]. With our work, even though we do not have a big data set we were able to reach the same conclusion. After that, we tuned the hyper-parameters of the SVM. By using 5-fold cross-validation it turned out that the radial basis function kernel was the most accurate one. Then, by performing a nested cross-validation on the penalty of the error term C and the kernel coefficient γ we found out, according to Fig.4, that combining $\gamma = 10^{-2}$ and $C = 10^{3}$ we reached the highest accuracy.

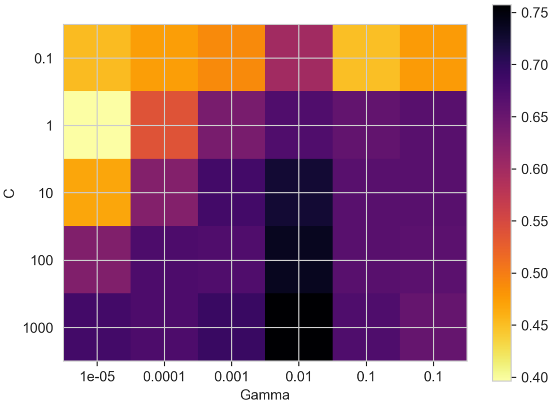


Fig. 4: SVM parameters tuning

5. CONCLUSIONS

The best accuracy we reached for the SVM using downsampling (80.4 %) is satisfying, regarding the fact that our data set is composed of recordings from different people, but not optimal. Thus, one of the conclusions that we can make is that is really difficult to analyse EEG data, and even more when the data set is not big enough. Moreover, one of the reasons why we did not achieve a higher accuracy might be the inaccurate labelling of the recordings. That is, we do not precisely know what are a doctor's expectations to define a recording as usable from further analysis.

On the other hand, the differences between brains are noticeable which means that is really difficult to build a classifier capable of being homogeneous independently of the differences between different person's brains. Taking this into consideration, we can conclude that out model is promising.

Regarding the different methods used, we can conclude that balancing data significantly increases the accuracy in the case of down-sampling, but leads to a lower one when using SMOTE.

An interesting result that is provided by our analysis is that using the whole set of features, without removing the correlated ones nor using the principal components, leads to a higher accuracy with imbalanced data, down-sampling and SMOTE. This might be due to the fact, as stated in 3.1, that most of the principal components contributes equally to the variance explanation. Nevertheless, by removing the correlated features it looks like we are loosing information. Thus, we observe that, with this data set, performing feature selection is unnecessary, as the best accuracy (80.4%) is reached using the whole set of features.

In conclusion, EEG data is difficult to treat and analyse due to the impossibility of making an homogeneous criterion between different patients. However, applying different balancing techniques and tuning the parameters when using the classification tools can give good accuracy and allow doctors to make a faster diagnosis in underdeveloped countries, that is, avoiding the loss of time that implies analysing the recording to know if it can be useful for the diagnosis.

6. REFERENCES

[1] Stahlhut C. Larsen J.E. Petersen M.K. Stopczynski, A. and L.K. Hansen, "The smartphone brain scanner: a portable real-time neuroimaging system," 2014., vol. PloS one, 9(2), p. p.e86733.

[2] A. Burton, "Brainwaves from bhutan.," 2015., vol. The Lancet Neurology, 14(12), pp. pp.1154–1155.

[3] MNE, "Reading and writing raw files," 2019.

[4] Xin Liu Forrest Sheng Bao and Christina Zhang, "Pyeeg: An open source python module for eeg/meg feature extraction," 2010.

[5] Amjed S. Al-Fahoum and Ausilah A. Al-Fraihat, "Methods of eeg signal features extraction using linear analysis in frequency and time-frequency domains," 2014.

[6] Scikit Learn, "Localoutlierfactor," 2019.

[7] Mikkel N. Schmidt Tue Herlau and Morten Mørup, ," in *Introduction to Machine Learning and Data Mining*. DTU, 2018, vol. I, pp. 34–43.

[8] Lawrence O. Hall W. Philip Kegelmeyer Nitesh V. Chawla, Kevin W. Bowyer, "Smote: Synthetic minority over-sampling technique," 2002.

[9] Kamran Raza Nadeem Qazi, "Effect of feature selection, smote and under sampling on class imbalance classification," 2012.