# Prepared by:

Julian Ramos

# General conclusions

* The classification performance for the On vs Off tests was reasonably good and stable. This means the models obtained from our methodology are robust as can be observed on the standard deviation results in figure 1.
* We can improve our classification. Here we only tried very basic features but there are many ideas from different domains that we can use to obtain even better features.
* Our data was incredibly noisy. In most cases at least 50% of the data in a task corresponded to the task itself. In some cases multiple tasks are in the same segment. Last we tried our best at labeling but there are probably mistakes.
* Regression is a much more difficult problem than classification and validating results against it is a bigger challenge. For instance, we cannot rely on a single physician to asses or score the tasks. By having a single Physician scoring, supposing there are instances in which our regression model is better than the physician, we cannot really know if our model is really doing better. In order to assess the regression model we need multiple physicians assessing the score. Not knowing what this entails of course means that I may be proposing something absurd however is worth considering since this constrain limits the prediction power of our regression model to the knowledge of a single physician. There are multiple papers in the literature on automated decision systems for diagnosis in which the system can be as good as 50 different physicians and being even able to get it right on very difficult cases. However are able to show that because they have multiple physicians, otherwise this could have been easily overlooked.

# General information about the tests performed

* All of the results reported either in graphs or verbally come from a 5 fold cross-validation leaving one of our participant’s data out for testing for each fold.
* The classifier used was a SVM using an rbf kernel
* Three different regression models were used, however the results were equally bad: elasticNet, linear regression and decisitionTree regressor.

# Classifying On vs Off Medication

We built a classifier from On and Off medication patients executing all the tasks in the test. From the data, we used a sliding window from which we computed the Power Spectrum Density from which we computed 7 different statistical features: mean, standard deviation, median, max, min, quartiles and interquartile range. Since we didn’t know the best size for the sliding window, a set of tests using different sizes and overlapping times was performed. The results summarized in figure 1 show that using a window higher than 5 seconds does not increase the accuracy or f-scores for the On and Off conditions. The results come from a cross-validation test where the training data used was from all but one of the patients. This left out patient-data was used later for testing and those results are the ones displayed in Figure 1. Since we have 5 patients in our data set we performed a 5 fold cross-validation.

|  |
| --- |
|  |
| Figure 1. Summary of Results experiments varying window length |

Similarly, experiments with the overlap value between sliding window where performed. The best overlap value was 1 second.

The next step was fine tuning the output bands of the power spectrum density. The results are summarized in Figure 2. We can see that having more than 100 bins in the spectrum of 0 to 20 Hz does not improve the results.

|  |
| --- |
|  |
| Figure 2. Fine tuning the power spectrum density. Performance metrics vs the number of bands in the spectrum of 0 to 20Hz. |

# Classifying On vs Off Medication per task

For this set of experiments we wanted to observe how well a classifier can predict On or Off medication using only data from specific tasks. Based on Vinod’s recommendation we only looked at the next tasks:

3. Finger tapping, 5. Pronation-supination (Right hand), 6. Postural Tremor, 11. Finger tapping, 13. Pronation-supination (Left hand)

14. Postural tremor of hands, 22. Gait right, 27. Gait left

Then, we used cross-validation and obtained the results summarized in figure 3.

|  |
| --- |
|  |
| Figure 3. Classification performance for each task |

All of these results however are to be taken only for illustrative purposes. For these evaluations the number of parameters exceeds the number of data points for training our models. These means that mathematically our models are underdetermined so this is just one among many solutions. In other words, if we were to run this same procedure again, using another different 5 participants the results could be very different.

# Regression : General comments

For these experiments all of the data from different participants and tasks was put together to estimate a regression model using the features previously obtained. In general the results obtained from this approach were not good. R^2 values were negative while the average mean square error was 2. The range is 3 so an error of 2 is not good at all. Here we have at least three problems:

1. Features are correlated: This is due to the very nature of our analysis in the frequency domain. We would expect to have a signal from a given frequency not on a precise value but on a spectrum.
2. Multitude of tasks: Putting together all of the tasks increases immensely the variance of the data making the regression function harder or maybe impossible to estimate.
3. Violations of many of the assumptions : Errors may not be normally distributed, Errors may have different normal distributions (reasonable to suspect on this one due to the different tasks).

Possible ways to fix most of this problems are building regression models per task and running feature selection to decrease the number of parameters.

# Regression: per task

These experiments were not performed again due to the under-constrained nature of our problem under this setup (less data than parameters), least squares is specially sensitive to this. We have here the same problems than we have on the general regression tests with exception of number 3.

# Regression after feature selection

The results are better ( R^2 score still negative but close to zero) but still not very good. The feature selection algorithm used was a uni-variate Chi-squared test using the top 10 and top 20 features. Using 10 features produced better results than 20. The mean square error was still very high ( around 2). While the results improve over our previous test but are not good either. They show that by performing a more sophisticated feature selection method we could improve the results even further.