Stroke Level Prediction through Pacman Game Data

Chapatte Mateo, Parchet Guillaume, Wilde Thomas
Department of Computer Science, EPFL, Switzerland
Mentor: Arzu Güneysu Özgür, Laboratory: CHILI

***Abstract*—Predicting the stroke level of a patient with Pacman game data could greatly improve the field of stroke rehabilitation as it would highly reduce the time required to estimate the patient's FMA. Nevertheless, this task needs to be accurate and explainable as it takes place in the medical domain.**

During our research, we found out that smoothness and velocity were important factors to predict the stroke level. Although the prediction of our models was not accurate enough to be used in medical area, this research shows that, even with a small number of patients, we can still find out a nice estimator.

I. Introduction

The Cellulo for rehabilitation project aims to provide practical and intuitive gamified rehabilitation using tangible robots as game agents and objects. Pacman is the first game that was designed with this approach and it is used to perform iterative upper arm exercises on patients recovering from strokes (Figure 1). Upper arm rehabilitation mainly focuses on relearning lost or weakened functional movements that are crucial for daily life activities.

The Fugl-Meyer Assessment (FMA) scale is an index to assess the sensorimotor impairment of individuals who experienced a stroke. It is widely used for clinical assessment of motor function. The therapists in Sion care center (Switzerland) started to use the Pacman gamified rehabilitation and collect data on their patient's plays. This displays valuable information that might help to evaluate a FMA subscore regrouping the ones related to upper extremities. This observation leads to two major questionings.

How well can we predict the FMA sub-score of a new patient by comparing his plays with the ones of previous patients? Can we detect if, on a new game, a patient's evaluated abilities regress or improve? This second point could be used as a first warning for the therapists to better spot if the patient's rehabilitation is in progress.

The main challenge to answer these questions comes from the fact that we only have a low number of patients (10 patients in total, with 9 distinct FMA scores). Also, an important recurrent aspect of medical fields is that we need to be able to explain the results in simple terms.

Effective recovery process includes large volumes of repetitive exercises. Therefore, we might expect more data to be collected along with the new patients and they might be used to further refine the models we present.

Figure 1. Gamified upper arm rehabilitation with tangible robots

II. Models and methods

A. Data Processing

The work is based on three data-sets. The first two are very small and contain the patient's FMA sub-scores as well as additional information such as the date they were interned, their age, weight, etc... The main data-set contains raw measurements taken at regular time interval during each of the plays performed by the patients. It contains as well information on the game configurations and some precomputed features from previous analysis done in CHILI lab (those are detailed in the feature section). This data-set had been previously been cleaned of the data points further than three standard deviations away from the mean.

Still, the minDist feature[1] contained 2.4% of missing values so we interpolated the missing ones with their neighbouring values (as minDist is fairly continuous).

It is important to note that even though the raw data contains 229'473 samples, this only represents a total of 145 games realized by the 10 studied patients during their rehabilitation.

The kind of data points we will train on is an important choice for the future results. The four main ways we found and explored were the followings:

- Use complete games as data points. This captures full motions and the overall game performance but this only yields 145 data points in total. Still, it can prove useful to better visualise the data.
- Use sub-games. Each game can be split at each time the patient collects an apple (which is seen through the field $game\_id$). This allows to capture long motions

[1] minDist value relates the difficulty to follow a linear trajectory

and performances as well as multiplying by five the number of data points.
- Use time windows of 10 seconds. As each game lasts exactly 3min the split is easy to perform and we can capture relevant motion features in this time interval. It yields a total of 2'458 data points.
- Use sub-motions windows. By looking at changes in axis directions, we can split the data into short sub-motions (where a sub-motion is the collection of data points going in the same axis directions). This yields 16'700 data points which is way more than the other data-sets but these data points capture less information and might not be as relevant.

B. Features

An important part of the project was to decide which features we should compute and use to train our models. Indeed, this choice might completely change the results as some features can greatly explain the state of the patient. For example, the jerk [2] is often used as a measure of the non-smoothness of a movement.

To find the best features, we decided to start by accessing the results of two related papers: "Quantitative assessment based on kinematic measures of functional impairments during upper extremity movements: A review" from Ana de los Reyes-Guzmán et al. and "Systematic Review on Kinematic Assessments of Upper Limb Movements After Stroke" from Anne Schwarz et al.. Those two studies performed a search to find all metrics used in kinematic assessment and classified them.

This assessment led to the identification of 24 features we could compute. Twenty basic ones: for each data-point window, its corresponding mean, maximum, minimum, median and standard deviation of its velocity, acceleration, jerk and minDist. Four additional features were computed from the velocity over the given data-point window: the rest ratio, number of velocity peaks, velocity mean maximum ratio and Fourier Transform level[3]. They aim to measure the smoothness of a movement. Indeed, a higher Fourier Transform level means higher frequencies and implies non-smoothness. This is complemented with the number of velocity peaks, the rest ratio[4] and the velocity mean maximum ratio.

The elbow maximum angular velocity and the trunk displacement were additional relevant features we cannot calculate with the current measurement taken during the patient's plays.

C. Data Exploration

1) PCA: We used principal component analysis (PCA), in order to get a sense of the data and see whether the data points with different FMA were separable or not. This also aimed to determine the features responsible for most of the variance on our four different data-sets.

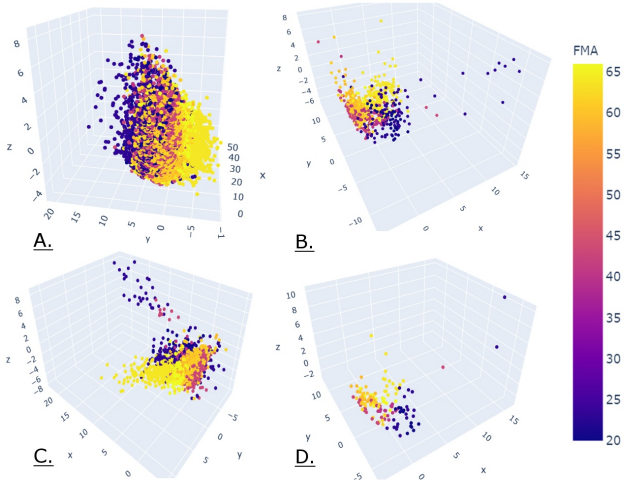

Figure 2. PCA visualization on the 4 different data-sets
A. Aggregated by sub-motion windows
B. Aggregated by $game\_id$ windows.
C. Aggregated by 10 seconds time windows.
D. Aggregated by full game windows.

The explanation of the axis of the four PCA instance give homologous and very interpretative results. If we analyse the 3 axis of the PCA based on the data aggregated by full game session, we get that :
The first component explains 32.8% of the variance which is around half of the total explained variance, the second component explains 22.8% of the variance whereas the third component explains 12.7% of the variance.

4 most important features coefficient in the feature space for the first principal component				
features	mean acc	std jerk	mean jerk	median jerk
coefficient	0.316	0.295	0.288	0.286

Here we can see that this component really represents the smoothness of the patient, because all those features are related to the second or third derivative of the position.

4 most important features coefficient in the feature space for the second principal component				
features	median v	mean v	mean max ratio v	rest ratio
coefficient	0.392	0.381	0.361	-0.286

The second component represents the speed. We can see it is mainly explained by the mean and the median of the speed which is an indicator of an overall high speed. A high mean-max ratio is equivalent to an overall quite constant speed. Finally, we can notice that it is inversely proportional to the rest ratio, therefore this axis represents a great control over the speed.

[2]jerk: rate of change of the acceleration, third derivative of the position
[3]threshold below which 80% of the energy of the Fourier Transform reside
[4]the rest ratio is the ratio between the time the patient is moving and the time the velocity is below a 20% threshold

4 most important features coefficient in the feature space for the third principal component				
features	velocity ft level	mean minDist	std minDist	median minDist
coefficient	0.386	-0.381	-0.379	-0.340

Lastly, the third component represents the minimal distance to the linear trajectory and the $velocity\ ft\ level$. This is interesting and also quite intuitive as the $velocity\ ft\ level$ is related to smoothness and the minDist value relates to a difficulty to follow a linear trajectory.

2) OLS: We also ran an ordinary least square model (OLS) on all our data-sets and they gave homologous results. Let's diagnose the OLS model run on the data aggregated by full game session:
The R-squared is 0.888,it means that our features explain about 89% of the variance which is an important amount of the total variance. The adjusted R-squared is 0.866. Its closeness with the R-squared is an indicator that, overall, the features are relevant.
However, by looking at the t-statistic of each features independently, we can notice that the majority of them do not have a sufficient statistical significance. Furthermore, we noted that features related to extremes such as minimal or maximal were the least significant. Therefore linear models might perform poorly and we did not further explore such models.

D. Models

To answer our two questions, we decided to use two different techniques to learn our models: the Random Forest and the Decision Tree algorithm. We decide to keep the first one as it performs better than the Decision tree and as it can predict a FMA that is not in our data-set. This is useful because when we want to predict the FMA score of a new patient, we might not have his score in our data-set (this is almost always the case for now as we only have 9 different scores) so we need to be able to predict new scores. The Decision Tree yields a lower general accuracy but it is way more explainable than the Random Forest. Indeed, using the decision tree, we can explain step by step why the model chooses a score rather than another one based on the evaluated features.

Figure 3 is an example of a decision tree trained at predicting one of three FMA categories (low [20,34], medium [35,49] or high [50,64] FMA). It shows each of its decisions and it is easy to interpret. In this example, to evaluate a new data-point, the model will first look if the 12^{th} feature (the median velocity) is below -0.391 or not. If it is the case, it will go to the left and check weather the 5^{th} feature (the mean jerk) is below -0.039 or not. If not, it will go to the right and, as a large majority of the previous data-points with these characteristics were classified category 0 (low FMA), it will also classify this new data-point in category 0. From this example we learn that someone with a low median velocity and a high mean jerk will be classified with a low FMA score which is intuitive as a low velocity and non-smooth movement can imply a bad quality of movements.

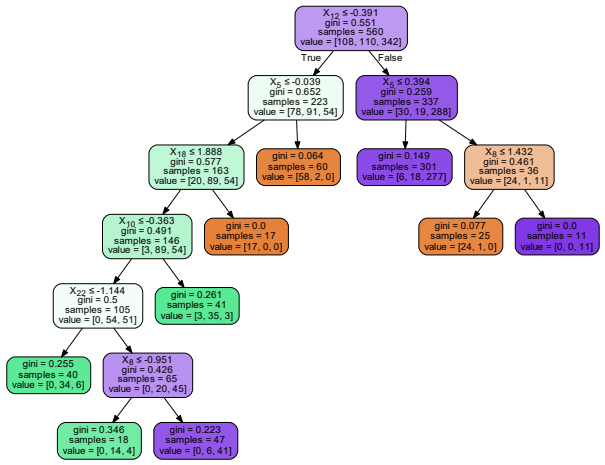

Figure 3. Decision Tree with three FMA categories

The two main metrics we used to analyse the results on the test set were the mean absolute error (MAE) and the error rate. The first one, gives us the average difference between the true label and the prediction. The second one gives the proportion of predictions that were further than a threshold from their label. We found those two metrics interesting for the project as they give a good interpretation of the results.

For both techniques cross validation was used to determine the best hyper-parameters. For the Random Forest, we saw that the $n\_estimator$ did not have a big impact on the result once above 75. Indeed, beyond this threshold, the mean square error does not change much more (maximum variation is 0.1 on validation). We decided to choose 100 as the default value of $n\_estimator$ as it had the best score for a relatively small number of trees, lowering the chance of over-fitting and the training time. The research of good hyper-parameters for the Decision Tree algorithm focused on the ability for the tree not to over-fit too much the training data. To achieve this we tuned the maximum depth parameter as well as the minimum impurity decrease. In Figure 4, we can see that the mean squared error on the validation test decreases until we reach 6 as maximum depth. With this information, the best trade-off seemed to use a maximum depth of 8 that would ensure enough power to the model as well as limiting over-fitting.

III. Results

Our leave one out strategy proceeds as follows: one participant is drawn out and the model is trained with the remaining participants, then we take the mean of the predictions of the data points belonging to the participant that was left aside. We repeat this for each participant

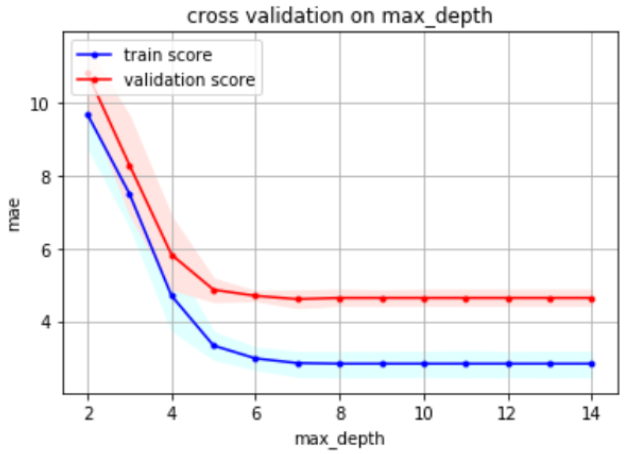

Figure 4. Cross validation on max depth for Decision Tree (with 90% bootstrapped confidence intervals with 1000 draws)

and finally we take the mean absolute error between each estimation and its real FMA label.
In the random split, we split the data points into training (80%) and testing (20%), then aggregate the data points by participants and take their mean as prediction. Finally, we take the mean absolute error of this estimation and the real FMA label, for each participant.

MAE loss on the random forest regressor[5].

	different data aggregation			
splitting strategy	game id	game session (3min)	time window (10s)	sub-motion window
Random split	5.32	3.79	5.30	9.41
Leave one out	11.6	12.0	12.3	12.9

MAE loss on the decision tree classifier[6].

	different data aggregation			
splitting strategy	game id	game session (3min)	time window (10s)	sub-motion window
Random split	5.24	4.69	5.48	11.38
Leave one out	11.9	12.2	12.1	14.6

To have a comparison point, it is interesting to compute the MAE that would be achieved by a statistical model that would not have access to the features of the data points but would instead give a prediction that maximise the likelihood. In the leave one out strategy, the model would output as a prediction: the mean of the label of the participants given for training. This model would output an MAE of 18.1. In the random splitting strategy, the model would output as a prediction: the mean of the label all the participants. This model would output an MAE of 16.3.

In every setup, the data aggregated by sub-motion gives the largest loss, this could come from the fact that by aggregating in this way, we lose the information during the change of direction which is probably relevant information. We perform much better in random split; this is because points coming from the same participant (and therefore same FMA due to the small number of participants) tends to cluster together. Overall, the loss is only a bit lower with the decision tree. The scenario that will be the closest to a real-life application would be to classify a new participant, therefore we would have a loss comparable to the "leave one out" strategy. Because we want our model to be explainable and that the decision tree does only perform bit worse than the random forest, we would choose the former and aggregate our data by sub-games as it performs the best. This score, 11.9, must be taken into perspective with 18.1 which is the upper bound above which our model is useless. Thus, we can say our model is not great but perform reasonably considering the amount of available data.

IV. Summary

Predicting the FMA sub-scores related to upper extremities using the Pacman data seems to be a promising technique to help therapists to estimate their patient's abilities.

So far, the prediction of an unseen patient is still imprecise, but it already gives a fair estimation and can underline the aspects of his plays that led to the predicted FMA score.

Additionally, the PCA visualisation yielded a better understanding of the important characteristics a patient displays while playing the Pacman games.

Overall, the results are encouraging if we take into account the low number of patients and the model's explainability that is required.

To further improve the current models, we could try to prune some of the least useful features (for example with a leave one out technique) and further prevent the trees from over-fitting. One could also try to include features specific to the patient's characteristics such as their age (for example, jerky movements might be more worrying for a young patient).

Acknowledgements

The authors thank Arzu Güneysu Özgür and Chloé de Giacometti for their careful reading and helpful suggestions.

[5]with $n\_estimator$=100
[6]with $min\_impurity\_decrease$=0.01 and max_depth=8

REFERENCES

[1] A. de los Reyes-Guzmán, I. Dimbwadyo-Terrer, F. Trincado-Alonso, F. Monasterio-Huelin, D. Torricelli and A. Gil-Agudo, "Quantitative assessment based on kinematic measures of functional impairments during upper extremity movements: A review," 2014, *Clin Biomech (Bristol, Avon)*.

[2] A. Schwarz, CM. Kanzler, O. Lambercy, AR. Luftand and JM. Veerbeek, "Systematic Review on Kinematic Assessments of Upper Limb Movements After Stroke," 2019.