Stroke Level Estimation through Pac-Man Game Data Played by Acute Stroke Patients

Victoria Perez Cortes, Simon Dürr, Valentine Perrin
CHILI Lab, EPFL, Switzerland

***Abstract*—The aim of this project was to predict the stroke level of patients based on a game of Pac-Man. Using hand movement recording during games and feature engineering with metrics derived from literature, we aim at classifying motor function of stroke patients by their score in the Fugl-Meyer Assessment test. We develop a model with meaningful features and easy interpretation, opening new perspectives for rehabilitation processes, and allowing health professionals to obtain fast stroke level assessment via the Pac-man game. It is a random forest model, leading to an accuracy of 0.56.**

I. Introduction

A stroke happens when poor blood flow to the brain causes cell death. Patients stroke level can be assessed by the Fugl-Meyer scoring (FMA). It is a stroke-specific, performance-based impairment index, with 5 aspects that can be affected by a stroke: motor functioning, sensory functioning, balance, joint range of motion, and joint pain. Here, it is the impairment of the upper body motor function that is explored. In FMA, the upper body motor function is graded out of 66, on 9 criteria such as reflex, movement, and flexibility. For each, a grade of 0 is given if the patient can't do the task: a level of 66 certifies no impairment in the motor function.

Pac-Man is a maze chase game; the player controls Pac-Man character through an enclosed maze. The goal of the game is to eat all of the dots placed in the maze while avoiding four ghosts that pursue the player. Here, the gameplay is strongly inspired by Pac-Man, but there are few differences, as shown in Figure 1 : there are 2 ghosts instead of 4, and dots are replaced by apples to be 'eaten'; the ghosts can catch them, making the player lose the apples. The game ends when all apples are collected.

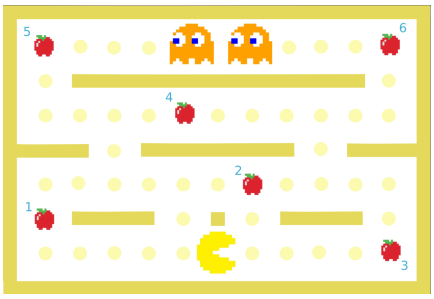

Fig. 1. Map of the game played by the acute stroke patients

Recording of the hand movements during the games leads to recording of numerous features, such as ones describing the hand position or movement velocity.

We aim at using feature engineering to develop meaningful features from game data to achieve successful estimation of motor function impairment level in the 19 acute stroke patients.

II. Data Exploration

The data consists of three data sets. One consists of the patients' stroke levels for the motor function and the 9 categories of the FMA; there are 19 patients, whom final scores range from 28 (most affected) to 66 (not affected); the distribution of the stroke level of the patients is shown in Figure 2. The target set is very small and the distribution is highly skewed towards less impaired patients, with 7 out of 19 patients showing only light impairment (FMA score >60).

The second data set consists of data collected through the games played. Each game is defined by the patient ID, game ID, day, time and settings. There are 32 variables measured for each timestamp, such as velocity, jerk, and acceleration. Each patient played between 2 and 10 games (15 of them played at least 8 games), for a total of 136 games.

The third data set contains the metrics relating to the apples during the games.

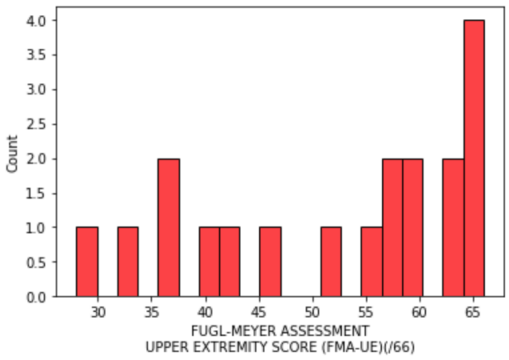

Fig. 2. Distribution of the 19 patients stroke levels (FMA)

III. Data Pre-processing

A. General pre-processing

We developed a data pre-processing pipeline. Initial data exploration showed that there is only one feature with missing values, the feature 'minDist', a measure of deviance. Out of 96'597 time points, 10'916 miss the minDist feature. They account for small portions of each game, in a equally distributed manner. We thus decided that missing minDist values will be

replaced by the last non-missing value (chronological order) after having split the data by games and by patients. As the minDist is related to position through the game, it explains why we use last known minDist value as a filling for missing values.

B. Dealing with the data set

1) Small data set: The highly skewed distribution of our data might raise some issues, or could affect the quality of our prediction. We can consider that patients with e.g. 64-65-66 scores are very close to each other and their score difference might be only a human error (from doctors evaluating stroke levels). Also, it could be a way to improve accuracy without losing the model interpretability. We thus split the outcome variable into bins and defined score levels, presented in part V-F. We tested two possibilities: we didn't consider the highly skewed distribution as an issue, or we also measured accuracy within 5 units of FMA.

2) Human data: Our data set is composed of human data. Patient differences are not negligible, particularly as we consider data from patients with strong differences in motor function impairment. This calls for careful considerations during data processing. The data should be split (to train and to test) by patients and/or then by the games they played, to obtain patient-level or game-level data, from which we later build the features. Patient-level data in that case consists of the data from all games of a patient. The generalization of a feature at levels above patient-level is not meaningful and is wrong considering human data.

IV. Feature Engineering

We need some literature metrics to pursue the development of a stroke level prediction model through feature engineering.

A. Literature metrics

It is known that quantitative measures of human movement quality are significant in rehabilitation for expressing the outcomes during rehabilitation treatments and discriminating between healthy and pathological conditions [1]; for that reason, it is essential to develop meaningful features so that we can accurately interpret the results. We found several features from literature [1], [2]. Considering the types of gameplay and task, we further select some metrics. Since we have position-based data collected through a robot held by the patient, the selected metrics should take into account the gameplay and the available data.

B. Computed features

We selected the following features from literature and engineered them.

- Mean velocity
- Mean jerk
- Trajectory error (deviance, using minDist)
- Number of velocity peaks
- Peak velocity
- Task/movement time
- Max velocity
- Mean velocity:Max velocity ratio
- Total motion (assessed by position variability)
- Mean time to get one apple
- Number of apples caught during a game (a number higher than 6 means that the patient lost some apples during the game - by being caught by a ghost or hitting a wall)

All these features will then be further selected through model selection for result optimization.

V. Models and Methods

A. Models

We aim at developing a interpretable model, that is meaningful for health applications. We needed a model that would not work as a 'black box', which can be explained to health professionals to justify further medical investigation and rehabilitation, and in which each feature contribution can be justified. Furthermore, with a very scarce target set, it is known that simple models should be preferred to prevent overfitting [3].

We used decision trees as the center piece of our models. Decision tree learning is a visual way to make predictions which corresponds to our criteria. There are two types of decision trees, and we implemented both, as explained in parts V-E1,V-E2: classification and regression tree analysis.

We implemented a model based on random forests, where the outcome is either the average of all decision trees outcome (regression) or a majority vote (classification).

B. Bootstrapping

To assess the standard error of the decision trees in the random forest we used bootstrapping as implemented in scikit-learn. In Bootstrapping we draw samples with replacement from the training set.

C. Out-of bag score (OOB)

As the dataset is small we used the out-of-bag score that one obtains with no reduction in training set size when implementing the random forest model with bootstrapping. This allows us to obtain an unbiased estimate of the generalization error [4]. The OOB score replaces a cross-validation, which would have made the dataset even smaller.

D. Feature selection

1) Feature importance: The feature importance describes which features are relevant for our model. Feature importance is easily implemented for random forests, as we can measure how each feature decreases the impurity of the split, and collect the average impurity reduction, across all trees. We use feature importance to guide feature selection.

2) Backward selection: We implemented a recursive feature elimination algorithm (RFE), which allows to obtain the best number of features as well as the best combination of features.

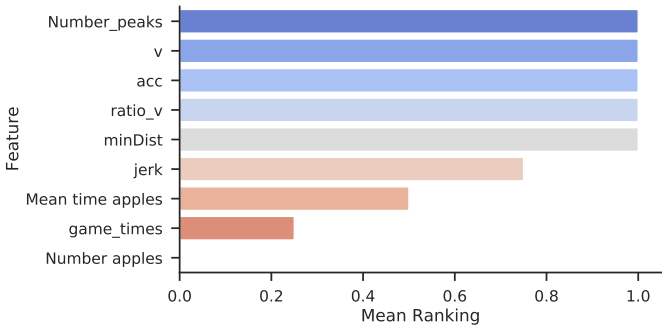

Fig. 3. Feature selection using Recursive Feature Elimination

E. Trees and random forests

1) Regression trees (by rounding): Our target set can be seen as a continuous, discrete set, thus we used regression tree analysis. Regression predictions were however round up to obtain discrete numbers, as the scores we predict are integers. This can be artificially interpreted as a classification where each level is a class.

2) Classification trees: Our target set can be seen as classes, where each class is a level of impairment. Thus we were able to use classification tree analysis.

3) Tree depth and number of trees: The depth of a decision tree is an important parameter to regulate the model fit. The tree depth leading to the best accuracy was investigated. Furthermore, as we work with random forest models, we need to define the best number of trees for model optimization.

F. PCA & NCA

Principal components analysis is an unsupervised learning method to derive a low-dimensional set of features from a large set of variables. Combining this method with K means clustering, one can define clusters to group the data in classes of similar elements, making the assumption that similar data belongs to the same class. It was used to determine whether score groups could be formed to train game. For the PCA, 6 labels were defined: very high (50-66), high (40-50), good (30-40), medium (20-30), low (10-20), very low (0-10).

Neighborhood component analysis (NCA) is a supervised method that learns a low-dimensional embedding of the data by finding a distance metric that maximizes the leave one out error in a stochastic nearest neighbor search. We used Linear Discriminant Analysis to initialize the NCA search. The data was pre-segmented into 4 classes before performing the NCA based on the FMA score to provide the labels. For NCA, four labels were defined: very high (>60), high (52-60), medium (45-52), and low (0-45).

VI. Results

The investigation for best pipeline was made in a sequential manner. We used the R2 score to assess the best hyper-parameters. The best model was then selected and further optimized based on the accuracy and the OOB, even if we used regressions (we predict discrete numbers so it becomes a classification problem): as a mean of interpretation, accuracy is valid. All accuracy measures for the FMA score are presented in Table 1, with the best hyperparameters.

Table 1. R2, accuracy and best parameters.

Model	R2	Accuracy	Depth	N-trees	OOB
Preliminary results (linear model)	0.40	///	///	///	///
Preliminary results (non-linear model)	- 0.40	///	///	///	///
Regression tree, patient-level, no apples	0.85	0.42	8	500	0.47
Regression tree, patient-level, with apples	0.93	0.79	8	50	0.46
Classification tree, patient-level	1.0	1.0	8	10	0.21
Classification tree, game-level	1.0	1.0	12	10	0.19
Regression tree, game-level, no apples	0.90	0.56	12	1000	0.23
Regression tree, game-level, with apples	0.90	0.53	12	1000	0.24

A. Preliminary results (from the lab)

The CHILI lab provided us with their initial research as a baseline for our project. The improvements brought by our models are compared to these using the R2 measure.

B. Regression using patient-level data

The regression using patient-level data achieved relatively low accuracy of predicted scores. We then added to that model the two features from the apples data (as described in feature engineering), at a patient-level. This resulted in an improved accuracy, as shown in Table 1. These good results are sustained by relatively high OOB scores, which is encouraging for model generalization at patient-level. But the patient-level data does not quite answer to our needs of a procedure to assess motor impairment from unique games, as it would mean that a patient will need to play several times before we will be able to assess its motor impairment. We reproduce that regression with game-level data.

C. Classification with game-level and patient-level data

While the classification models perform well when looking at the accuracy and R2 metrics, the OOB score is low, showing that the generalization error for the classification models are high. A likely reason for this is that the target set is skewed, containing many patients with only light impairment of motor functions.

D. Regression using game-level data

The regression using game-level data achieved relatively high accuracy of predicted scores. We then added to that model the data from the game apples, at a game-level. This resulted in a slightly lower accuracy, but a slightly higher OOB score, which is why we further selected that model as our best one for further optimization.

E. Pipeline and feature selection

All models implemented show strong improvement compared to the preliminary results from the lab. We then implemented a pipeline computing feature selection and used OOB as an estimate for the test error. The number of features to keep was set to 5 (the best number of features from RFE, Figure 3) and were the trajectory error (minDist), the mean velocity, the mean:max velocity ratio, the mean acceleration, and the number of velocity peaks. That final model gave us the following results for the total FMA scores estimation (using regression tree at game levels and with apple data):

- R2 score : 0.90
- OOB score : 0.27
- Accuracy : 0.56
- Depth : 12
- N-trees : 1000

As we predict a single number between 1 and 66, an accuracy of 0.56 is very encouraging as it is already way better than by random chance.

F. Principal Component Analysis (PCA)

The variance in our data is most explained by the two first components. We then determine the number of clusters. In order to do this, we fed these principal components to the k-means algorithm. After k = 5, the change in the value of inertia (sum of the squared distances to the nearest cluster) is no longer significant and most likely, neither is the variance of the rest of the data after the elbow point.

Given these results, we have performed PCA with 2 principal components, using 5 clusters (very low, low, medium, high, very high) and we obtained the following results:

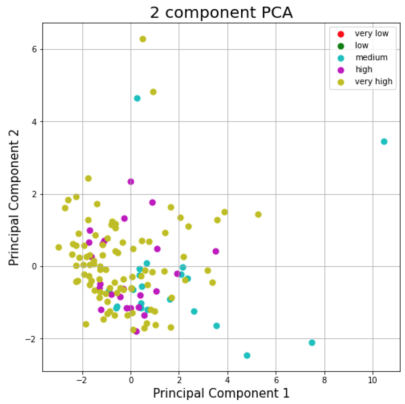

Fig. 4. Principal component analysis with 5 clusters.

This graph shows clusters for the *very high* and *high* categories, but it does not separate well these results from the *medium* or *low* categories. This is probably due to the lack of data for these low levels.

G. Neighborhood component analysis (NCA)

NCA is a supervised technique which cannot be used to create features as this would constitute leakage of the labels into the training data. The NCA analysis is nevertheless informative as it clearly shows that there exists a learnable embedding for the data that separates patients of different stroke levels in a reduced 2-dimensional feature space. In addition, new patients can be added to the learned embedding based on the previous data to find out their stroke level in the assumption that the existing data already well describes the different stroke levels. Unlabeled stroke patients after transformation should appear in the respective corner of Fig. 5.

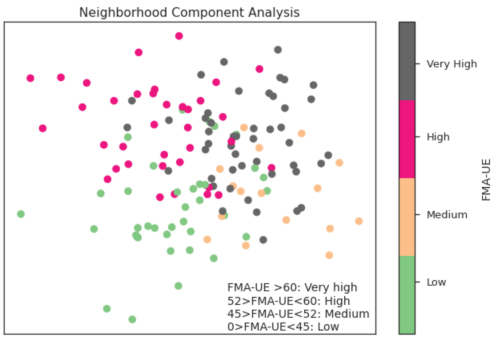

Fig. 5. Neighborhood component analysis showing the embedding using the categorical labels

VII. Discussion & Conclusion

We developed an interpretable model with meaningful features to estimate the stroke level of acute stroke patients. We constructed a random forest model using game-level data, and features defined by feature selection algorithms, and were able to predict the FMA score with an accuracy of 0.56.

However, even if the R2 score is close 1, we observed that the out-of-bag score was relatively low. This indicates that our model has issues with generalization. This is due to the scarcity of our data; it means that prediction for a new patient FMA score is hard. The data size should be increased to improve the generalization of our model.

Interpretability and meaningfulness are the most essential aspects of our model: it is made such that helpful conclusions can be obtained, allowing direct applications in health domains. We developed a way to assess stroke level impairment via a fast procedure, allowing health professionals to orient the rehabilitation and reeducation in consequence. This model could be further improved by pursuing different leads:

- Using bin-classification (accuracy within a given range) is definitely a way to improve the accuracy of the results, without losing interpretability of the model. Indeed, accuracy within 5 allows to obtain a 0.82 accuracy with our best model.
- For patient level data, it could be useful to subdivide the data to have more data and improve the results further.
- Also, the relatively good OOB scores obtained with patient-level data suggest that patient-level prediction could be made from game-level data to increase model stability, stroke level estimation being given as a majority vote/mean/median of all game results.

REFERENCES

[1] Ana de los Reyes-Guzmán, Iris Dimbwadyo-Terrer, Fernando Trincado-Alonso, Félix Monasterio-Huelin, Diego Torricelli, and Angel Gil-Agudo. Quantitative assessment based on kinematic measures of functional impairments during upper extremity movements: A review. *Clinical Biomechanics*, 29(7):719–727, 2014.

[2] Anne Schwarz, Christoph M Kanzler, Olivier Lambercy, Andreas R Luft, and Janne M Veerbeek. Systematic review on kinematic assessments of upper limb movements after stroke. *Stroke*, 50(3):718–727, 2019.

[3] Dealing with very small datasets. [Online]. Available: https://www.kaggle.com/rafjaa/dealing-with-very-small-datasets.

[4] Out-of-bag error estimate. [Online]. Available: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#ooberr.