Report 1: Regression on Parkinson's Disease Data

Seyed Mohammad Mehdi Hosseini, s301769,
ICT for Health attended in A.Y. 2022/23

November 17th 2022

1 Introduction

Parkinson's disease is a neurodegenerative disorder that affects mainly dopamine-producing neurons in the substantia nigra, a region of the brain that produces dopamine. Symptoms generally develop slowly over the years and it affects about 2–3% of the population $\geq$65 years of age.[1] As a result of the diversity of the disease, the progression of symptoms varies from person to person. Tremors, limb rigidity, paucity of movement and even postural instability are some of symptoms. Moreover, it may lead to difficulties with correctly speaking due to the inability to control the vocal cords and vocal tracts.

The first-line drug for managing Parkinson's motor symptoms is levodopa. Levodopa is absorbed in the blood from the small intestine and travels through the blood to the brain, where it is converted into dopamine. Patients receive levodopa, but the dosage should increase as the illness progresses as well as be given at the right time during the day to prevent freezing. In order to tailor treatment to an individual patient, it would be helpful to measure the total UPDRS (Unified Parkinson's Disease Rating Scale) frequently during the day. Patient voice recordings is a simple technique which can be used to generate vocal features that can be further be used to regress total UPDRS scores on a daily basis through smartphones.

Several linear regression models based on different optimization approaches were used on the public dataset available at [2] in this study to estimate total UPDRS.

2 Data Analysis

The dataset at [2] has 22 features, listed in Table 1. Upon this list, "subject ID", "Jitter:DDP', "Shimmer:DDA" were removed. It's to be noted that the data has been manipulated using "test time" feature in order to have only average voice parameter values in one day. Finally, "Total UPDRS" is the regressand, with the other 18 features acting as regressors.

Figure 1 shows the measured covariance matrix for the entire normalized dataset: correlation between total and motor UPDRS is evident, and strong correlation also exists among

1	subject	2	age	3	sex
4	test time	5	motor UPDRS	6	total UPDRS
7	Jitter(%)	8	Jitter(Abs)	9	Jitter:RAP
10	Jitter:PPQ5	11	Jitter:DDP	12	Shimmer
13	Shimmer(dB)	14	Shimmer:APQ3	15	Shimmer:APQ5
16	Shimmer:APQ11	17	Shimmer:DDA	18	NHR
19	HNR	20	RPDE	21	DFA
22	PPE				

Table 1: Dataset features

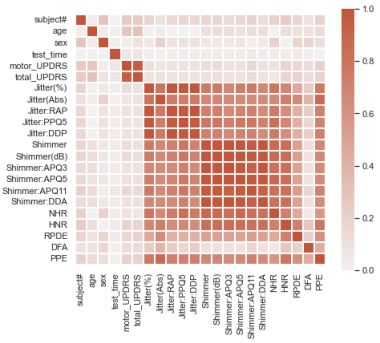


Figure 1: Covariance Matrix of the Features

shimmer parameters and among jitter parameters (possible collinearity); on the other hand only a weak correlation exists between total UPDRS and voice parameters.

Considering the above-mentioned manipulation, the total number of instances in the manipulated data is 990; data are shuffled and the first 50% of the points are used to train the linear model, and the rest are used for the test the model performance. All the data (both regressors and regressand) are normalized using mean and standard deviation measured on the data:

$$X_{m,N} = \frac{X_m - \mu_m}{\sigma_m}, Y_N = \frac{Y - \mu_Y}{\sigma_Y} \tag{1}$$

3 Linear Regression

The model assumed in linear regression is:

$$Y = w_1 X_1 + \ldots + w_F X_F = \mathrm{X}^T \mathbf{w} \tag{2}$$

where Y is the regressand (total UPDRS), $\mathrm{X}^T = [X_1, \ldots, X_F]$ is the row vector that stores the F regressors (random variables) and $\mathbf{w}^T = [w_1, \ldots, w_F]$ is the weight vector to be optimized. The optimum vector $\mathbf{w}$ is the one that minimizes the mean square error:

$$e(\mathbf{w}) = \mathbb{E}\left\{ \left[Y - \mathrm{X}^T \mathbf{w} \right]^2 \right\}. \tag{3}$$

3.1 Linear Least Squares (LLS)

Linear Least Squares (LLS) directly finds $\mathbf{w}$ by setting to zero the gradient of $e(\mathbf{w})$:

$$\nabla e(w) = \; -2\mathbf{X}_N^T \; \mathbf{y}_N + 2\mathbf{X}_N^T \; \mathbf{X}\mathbf{w} = 0 \tag{4}$$

$$\hat{\mathbf{w}} = \left(\mathbf{X}_N^T \mathbf{X}_N \right)^{-1} \mathbf{X}_N^T \mathbf{y}_N \tag{5}$$

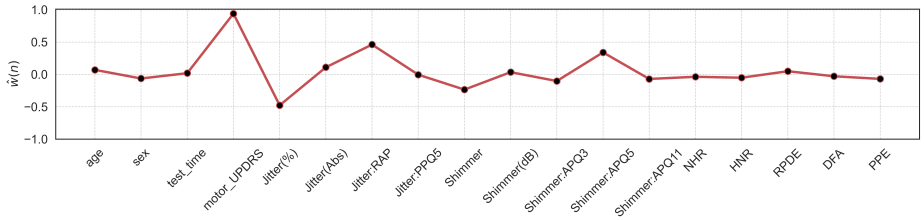
Given $\hat{\mathbf{w}}$, the normalized regressand for the test dataset is estimated as:

$$\hat{y}_{N,te} = \mathbf{X}_{N,te}^T \mathbf{w} \tag{6}$$

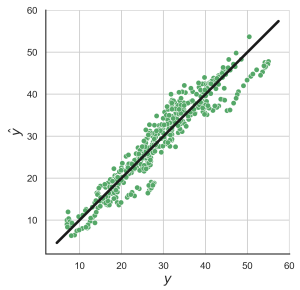
The denormalized regressand is instead:

$$\hat{y}_{te} = \sigma_Y \hat{y}_{N,te} + \mu_Y. \tag{7}$$

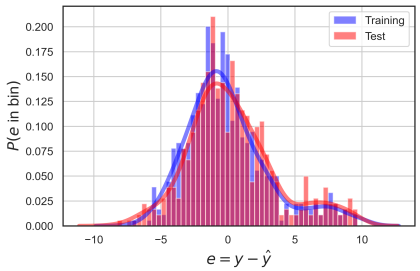
Figure 2 shows the results obtained with LLS, using denormalized data.



(a) $\hat{\mathbf{w}}$ for Linear Least Squares (LLS).



(b) $\hat{y}$ versus y for test dataset.



(c) Histogram of $y - \hat{y}$ for training and test datasets.

Figure 2: Results for Linear Least Squares (LLS).

3.2 Normal Steepest Descent (NSD) Algorithm

Normal Steepest Descent (NSD) algorithm finds the "optimum" value of γ at each step. Note that the Hessian matrix for the square error is:

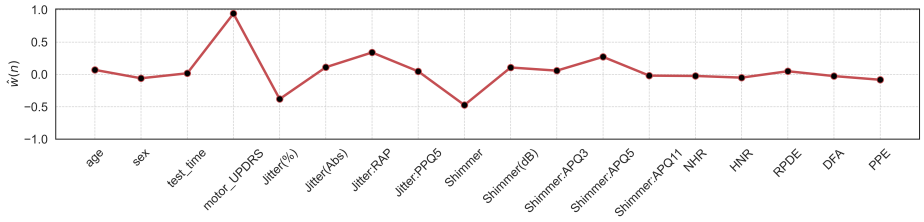
$$\mathbf{H}(\mathbf{w}(\mathbf{i})) = \ 2\mathbf{X}^T \ \mathbf{X} = \mathbf{H} \tag{8}$$

As it doesn't depend on the $\mathbf{w}$ so that it can evaluated just once. Algorithm start with an initial guess $\hat{\mathbf{w}}(\mathbf{0})$ and it evaluates the gradient and the Hessian matrix at point $\hat{\mathbf{w}}(\mathbf{i})$ and on the next step it finds the new point in an iterative mode unless a stopping condition is met:

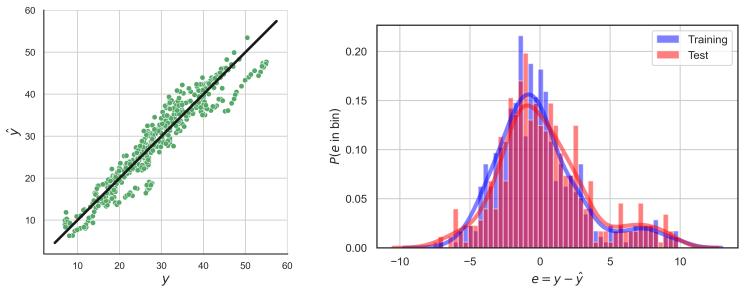
$$\nabla e(\hat{w}(i)) = \ -2\mathbf{X}^T \ \mathbf{y} + 2\mathbf{X}^T \ \mathbf{X}\hat{\mathbf{w}}(\mathbf{i}) = 0 \tag{9}$$

$$\hat{\mathbf{w}}(i+1) = \hat{\mathbf{w}}(i) - \frac{||\nabla e(\hat{w}(i))||^2}{\nabla e(\hat{w}(i))^T \mathbf{H} \nabla e(\hat{w}(i))} \nabla e(\hat{w}(i)) \tag{10}$$

Figure 3 shows the results obtained with Normal Steepest Descent (NSD) algorithm, using denormalized data.



(a) $\hat{\mathbf{w}}$ for Normal Steepest Descent (NSD) Algorithm.



(b) $\hat{y}$ versus y for test dataset.

(c) Histogram of $y - \hat{y}$ for training and test datasets.

Figure 3: Results for Normal Steepest Descent (NSD) Algorithm.

3.3 Local Steepest Descent (LSD) Algorithm

Given a test point, in Local Steepest Descent (LSD) algorithm, instead of finding $\hat{\mathbf{w}}$ from all the points in the training dataset, only the **N**-th closest point are used. For each of the normalized test points $\mathbf{x}$, the algorithm finds the nearest $\mathbf{N}$ (batch size) points in the normalized training dataset $\mathbf{X}_{tr}$ that are closer to $\mathbf{x}$. There will be a new training dataset with matrix $\mathbf{X}_{tr}(x)$ and vector $\mathbf{y}_{tr}(x)$, both with $\mathbf{N}$ rows only for each test point. Then we can find the $\hat{\mathbf{w}}(x)$ that minimizes:

$$||\mathbf{X}_{tr}(x)\mathbf{w} - \mathbf{y}_{tr}(x)||^2 \tag{11}$$

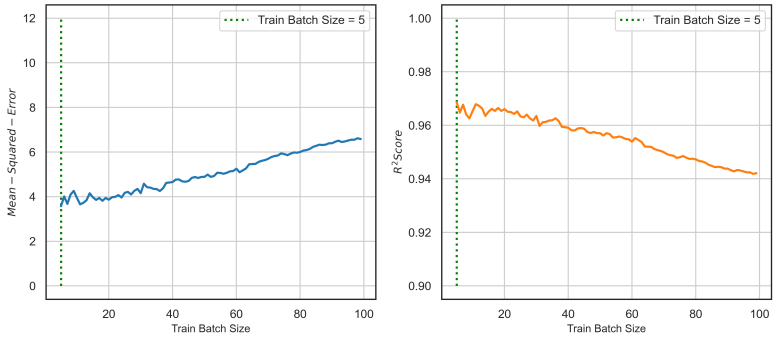
Then it's possible to find the normalized regressed value:

$$\hat{\mathbf{y}} = \mathbf{x}^T \hat{\mathbf{w}}(x) \tag{12}$$

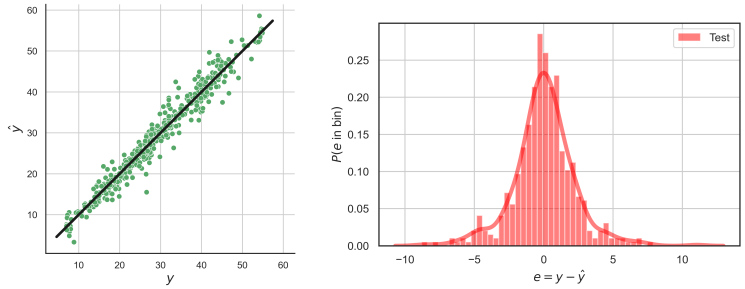
Figure 4 shows the results obtained with Local Steepest Descent (LSD) algorithm, using denormalized data.

3.4 Comparison

The regression error $e = y - \hat{y}$ for the training and test datasets can be seen as a random variable, which can be statistically described. Table 2 lists its main statistical parameters: mean μ_e, standard deviation σ_e, mean square value MSE, and coefficient of determination



(a) Results for different **N** (Batch Size) for Local Steepest Descent (LSD) Algorithm, $\mathbf{N = 5}$ results in the minimum **MSE** and maximum $\mathbf{R}^2$.



(b) $\hat{y}$ versus y for test dataset with $\mathbf{N = 5}$.

(c) Histogram of $y - \hat{y}$ for test dataset with $\mathbf{N = 5}$.

Figure 4: Results for Local Steepest Descent (LSD) Algorithm.

R^2, for the three analyzed methods. All these methods have been set to run 1000 epochs but the algorithm includes an early stopping criteria. The algorithm stops training unless the MSE has been reduced at least one-thousandth of the last ten training MSE.

According to the Figures 2 and 3, the results obtained by both the LLS and the NSD algorithms are significantly similar. Both histograms show that they have got a combination of two Gaussian PDFs. Moreover, It is unlikely that overfitting occurs between training and test data since there isn't much difference between them. There can be a possible co-linearity between Schimmer and Schimmer:APQ5 features accroding to Figures 2a and 3a. Motor UPDRS, Jitter(%), Schimmer and Schimmer:APQ5 are the most relevant features. On the other hand, LSD has a clean Gaussian PDF and it also outperforms LLS and NSD. All the standard deviations recorded in the research are much lower than the total UPDRS standard deviation which is 10.805 and, LSD's standard deviation has been recorded as the smallest value. In general, all of the algorithms have recorded R^2 and R scores greater than or equal to 0.9. Even though there are some points where the algorithms are wrong, the algorithms are pretty good overall but LSD is the best among the algorithms.

	Dataset	Min	Max	μ_e	σ_e	MSE	R^2	R
LLS	Training	-7.133	10.038	7.034×10^{-15}	3.131	9.799	0.915	0.956
	Test	-8.253	9.805	3.124×10^{-1}	3.285	10.885	0.904	0.951
NSD	Training	-6.985	10.219	6.689×10^{-15}	3.134	9.824	0.915	0.956
	Test	-7.623	9.824	3.201×10^{-1}	3.285	10.896	0.904	0.952
LSD	Test	-8.795	11.055	3.386×10^{-2}	2.199	4.836	0.957	0.979

(a) Results for Seed equal to PoliTO ID (301769), Batch Size for LSD is = 5.

	Dataset	Min	Max	μ_e	σ_e	MSE	R^2	R
LLS	Training	-7.328	10.193	-1.256×10^{-15}	3.126	9.784	0.915	0.957
	Test	-8.223	20.174	4.319×10^{-2}	3.450	12.033	0.893	0.946
NSD	Training	-7.387	10.197	-1.297×10^{-15}	3.133	9.827	0.915	0.956
	Test	-8.059	20.213	4.352×10^{-2}	3.444	11.983	0.894	0.946
LSD	Test	-11.500	17.622	-7.918×10^{-2}	2.302	5.395	0.952	0.976

(b) Average of Results from 20 Random Seeds Between (1,100), Batch Size for LSD is = 5.

Table 2: Comparison among the three methods: LLS, NSD and LSD.

4 Conclusions

Understanding Parkinson's disease and its progression is the first step to living well with it. To predict the progression of the Parkinson's Disease, we've evaluated several methods. The results are pretty good overall but Local Steepest Descent is the most reasonable and reliable among the others in comparison with the other algorithms which led to more errors in predicting high total UPDRS values. This is critically important to identify the most severe stage of Parkinson's Disease. Although the motor and total UPDRS acquisition process takes a long time and effort by neurologists, there is a big correlation between motor UPDRS, and total UPDRS. It's completely obvious that without this feature, predictions will face much more errors than before. There is a solution to ease the process of total UPDRS acquisition for neurologists by using some image processing and mobile application-based solutions on a trial basis through online surveys. For example, scoring based on the movement of four fingers with respect to the thumb finger or analysing the posture of patient when trying to rise from a chair using smartphone cameras. There are also other solutions to overcome this and it can help neurologists to save more time.

References

[1] Poewe, W., Seppi, K., Tanner, C. et al. Parkinson disease. Nat Rev Dis Primers 3, 17013 (2017). https://doi.org/10.1038/nrdp.2017.13

[2] https://archive.ics.uci.edu/ml/datasets/Parkinsons+Telemonitoring