Report 1: Regression on Parkinson's disease data

Federico Villata, s247586,
ICT for Health attended in A.Y. 2022/23

November 17th 2022

1 Introduction

Parkinson's disease is characterized by resting tremors, rigidity, akinesia, and bradykinesia. Typically (not always) patients affected by this disease have problems speaking, as they cannot control the vocal cords and vocal tract.

The disease is treated by prescribing Levodopa to increase dopamine and restore the balance with acetylcholine. The dose must be regulated depending on the severity of the symptoms and, for this reason, it is of utmost importance to accurately measure the disease progression. This can be achieved through the measure of total UPDRS (Unified Parkinson's Disease Rating Scale).

The measurement should be carried out many times during the day using a simple setup in order to allow the patient to autonomously carry out the tests, for this reason, several studies indagated whether the voice can be a predictor of the health condition of the patient, due to the known relationship between Parkinson's disease and voice quality.

Voice quality can be measured, even with a smartphone, to generate vocal features that can be used to regress total UPDRS.

Linear Regression, performed using Linear Least Square (LLS) and Steepest Descent, and Local Linear Regression, performed using Steepest Descent, were used on the public dataset "Parkinsons Telemonitoring Data Set" [1] to estimate total UPDRS, and the results were compared.

2 Data analysis

The 22 features available in the "Parkinsons Telemonitoring Data Set" [1] are listed in table 1: of these, subject ID, test time, Jitter:DDP, and Shimmer:DDA were removed. Total UPDRS is the regressand. All the remaining features were used as regressors in linear regression.

The number of points in the dataset is 990; data are shuffled and the first half (445) of the points are used to train the model and the remaining 445 are used to test the model performance. Data are normalized using mean and standard deviation measured on the training dataset.

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: List of features

Figures 1a and 1b show the measured covariance matrix for the entire normalized dataset: motor and total UPDRS are highly correlated among themselves, and the same occurs for shimmer parameters and jitter parameters. This might give rise to collinearity or multicollinearity: one feature among the regressors can be linearly derived from other regressors. On the other hand only a weak correlation exists between total UPDRS and voice parameters.

In the following considerations the rows of the original DataFrame are randomly permuted (shuffled) to prepare a new DataFrame. Shuffling is performed setting the seed, to have reproducibility of the script

Note that, to get a meaningful comparison between linear regression based on LLS, linear regression based on steepest descent and local linear regression based on steepest descent, the training dataset and test datasets are the same.

3 Linear Regression

Linear regression performs the task to predict a dependent variable value ($\mathbf{y}$) based on a given independent variable ($\mathbf{X}$). So, this regression technique finds out a linear relationship between $\mathbf{X}$ (input) and $\mathbf{y}$ (output).

The model assumed in linear regression is

$$Y = w_1 X_1 + ... + w_f X_f = \boldsymbol{X^T w} \tag{1}$$

Where Y is the regressand (total UPDRS), $\boldsymbol{X^T} = [X_1, ..., X_f]$ stores the f regressors and $\boldsymbol{w}$= $[w_1, ..., w_f]$ is the weight vector to be optimized. Y, X_1,..,X_f are all random variable.

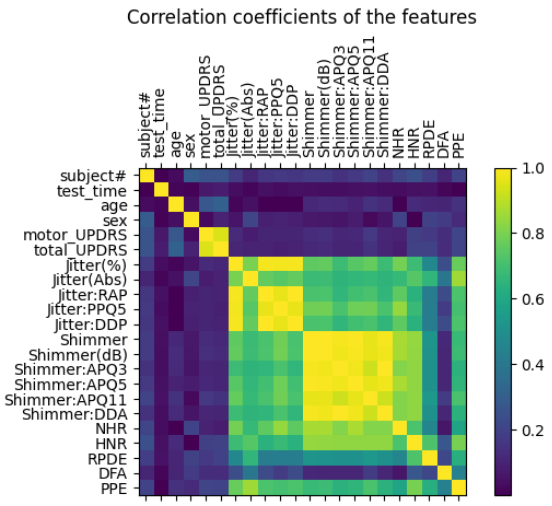
3.1 Linear regression based on Linear Least Square

LLS minimizes the mean square error (MSE). The optimum weight vector $\mathbf{w}$ can be obtained in closed form as:

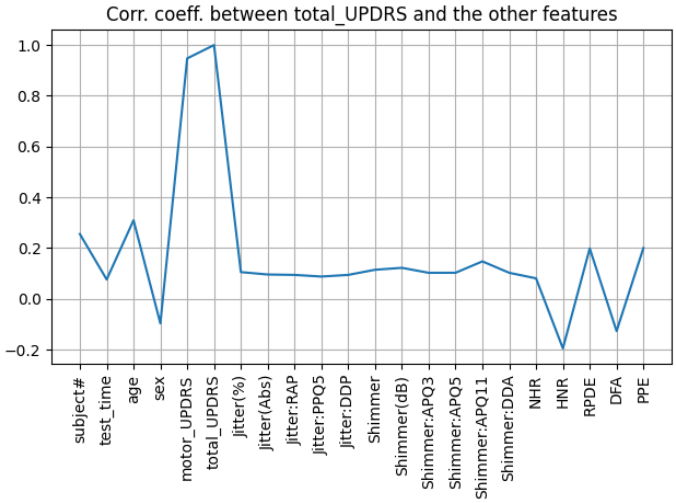
$$\hat{\boldsymbol{w}} = argmin\mathbb{E}(Y - \boldsymbol{X^T w})^2 = \boldsymbol{(X^T X)^{-1} X^T y} \tag{2}$$

Where $\mathbf{X}$ is the matrix that stores the (normalized) training regressor points and $\mathbf{y}$ is the (normalized) training regressand vector.

figure 2 and table 2 show the results obtained with LLS.



(a) Covariance matrix of the features



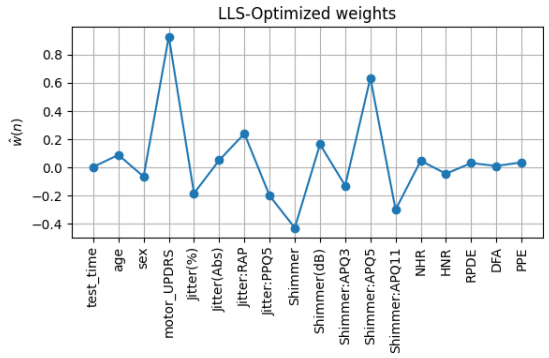
(b) UPDRS

Figure 1: Data analysis

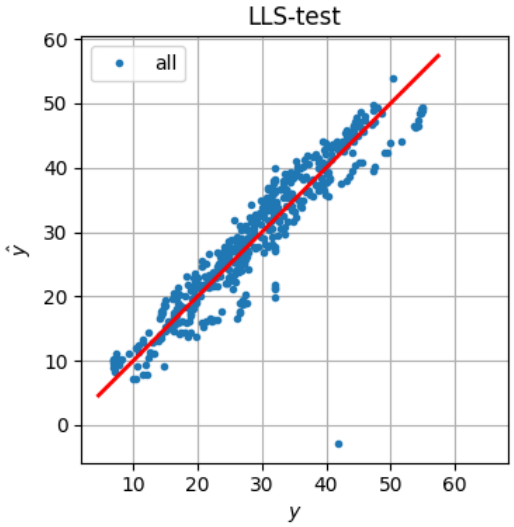
Fig. 2a shows the weights associated with each considered feature. Fig. 2b shows $\hat{y}$ versus y, the predicted results is close to the expected one, indeed table 2 confirms the quality of the prediction since the coefficient of determination R^2 is close to 1. Lastly, Fig 2c shows the estimation error histogram and there is not much difference in the training and test subsets, which means that there is no overfitting.

3.2 Linear regression based on Steepest Descent

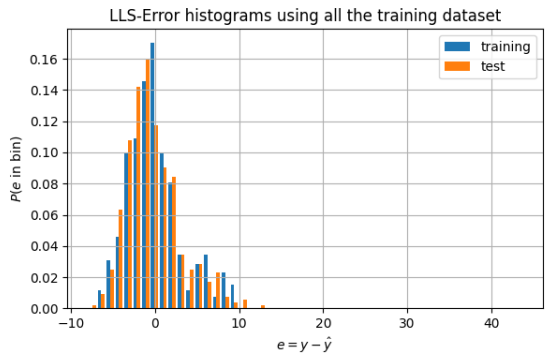
Steepest descent aims to minimize the MSE through iterations. It starts from a random weight vector $\boldsymbol{w}$ and at each step minimizes the following function:



(a) LLS weights



(b) $\hat{\mathbf{y}}$ vs $\mathbf{y}$



(c) LLS error histogram

Figure 2: LLS results

$$f(\boldsymbol{w}) \simeq g(\boldsymbol{w}) = f(\boldsymbol{w}_i) + \nabla f(\boldsymbol{w}_i)^T(\boldsymbol{w} - \boldsymbol{w}_i) + \frac{1}{2}(\boldsymbol{w} - \boldsymbol{w}_i)^T \boldsymbol{H}(\boldsymbol{w}_i)(\boldsymbol{w} - \boldsymbol{w}_i) \tag{3}$$

	min	max	mean	std	MSE	R^2	corr_coeff
Training	-6.430	9.885	5,14e-15	3.319	11.015	0.906	0.952
test	-7.979	44.737	-1.482e-01	3.806	14.506	0.869	0.935

Table 2: Results obtained with linear regression based on linear least square

Where $\boldsymbol{H(w_i)}$ is the Hessian matrix evaluated at $\boldsymbol{w_i}$.

In order to compute each element of the formula the quadritic problem can be assumed.

To perform linear regression, the following parameters were chosen to determine the stopping condition: number of iterations = 1000 and $|f(\boldsymbol{w}_{i+1}) - f(\boldsymbol{w}_i)| < \epsilon$, with $\epsilon = 10^{-8}$.

The goal was to ensure a high number of iterations to let converge the algorithm, also a stopping condition was applied if no progress are made.

Figure 3 and table 3 show the results obtained with Steepest Descent. Also in this case the predicted result is close to the expected one since R^2 is close to 1 and there is no overfitting since the error probability distribution is the same for the two sets.

Fig. 3a shows the weights associated with each characteristic considered, Fig. 3b shows ŷ versus y, whereas Fig 3c shows the estimation error histogram.

	min	max	mean	std	MSE	R^2	corr_coeff
Training	-6.507	9.680	4,543e-15	3.325	11.054	0.906	0.952
test	-7.842	44.301	-1.520e-01	3.769	14.226	0.872	0.937

Table 3: Results obtained with linear regression based on steepest descent

4 Local Linear Regression

Local linear regression is a linear regression based not on the whole training dataset but only on the k points closest to the point under consideration.

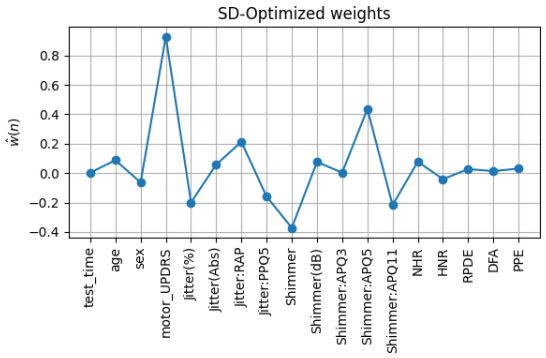
Figure 4a shows the relationship between the chosen number of points (k) and the MSE on the test set; $k = 40$ was chosen since it was the one that achieved the best results. Also, the more points that are taken into account, the closer we get to the results of a linear regression based on the entire training dataset.

4.1 Local Linear Regression based on Steepest Descent

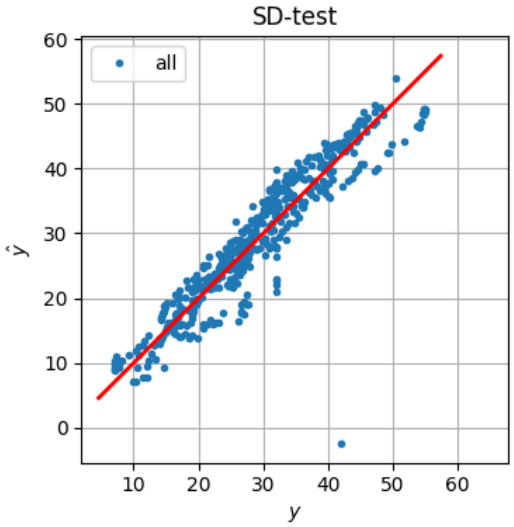
The stopping condition applied to Local linear regression is the same as the one used on Steepest Descent.

Results are shown in Figure 4 and table 4.

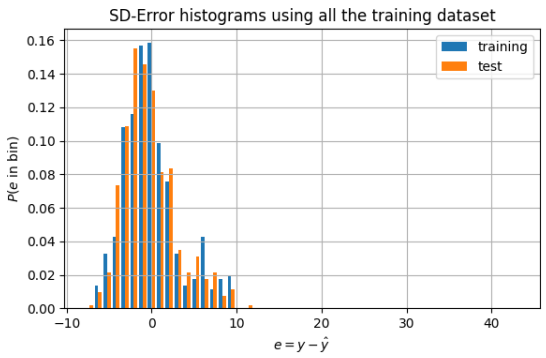
Fig. 4b shows ŷ versus y and it stands out that the estimated values ŷ are close to the true values y. In order to assess overfitting the algorithm was applied both on the test and training set, when applied on training the point being tested was not part of the local



(a) Steepest Descent weights



(b) $\hat{\mathbf{y}}$ vs $\mathbf{y}$



(c) Steepest Descent error histogram

Figure 3: Steepest Descent results

trainset. The results are shown in Fig 4c, also in this case the error probability distribution

is the same for both datasets.

	min	max	mean	std	MSE	R^2	corr_coeff
Training	-7.208	6.289	-0.019	2.081	4.330	0.963	0.982
Test	-8.092	20.944	-0.025	2.348	5.512	0.950	0.975

Table 4: Results obtained with local linear regression based on steepest descent

5 Additional test

The algorithms were also tested with 20 different randomly generated seeds to verify the reliability of the previously obtained results. Averaged results are shown in table 5, table 6, and 7.

	min	max	mean	std	MSE	R^2	corr_coeff
Training	-7.145	10.299	-5.198e-16	3.153	9.947	0.912	0.955
Test	-8.269	21.736	2.271e-03	3.446	11.961	0.896	0.947

Table 5: Results obtained with local linear regression based on linear least square with 20 different seeds and averaged

	min	max	mean	std	MSE	R^2	corr_coeff
Training	-7.138	10.266	-2.074e-16	3.161	9.995	0.912	0.955
Test	-8.118	21.499	5.563e-03	3.437	11.892	0.897	0.947

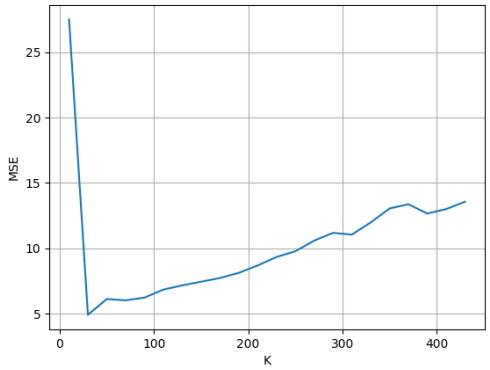
Table 6: Results obtained with linear regression based on steepest descent with 20 different seeds and averaged

	min	max	mean	std	MSE	R^2	corr_coeff
Training	-8.980	9.417	0.070	2.130	4.555	0.960	0.980
Test	-8.225	9.743	0.041	2.102	4.439	0.961	0.981

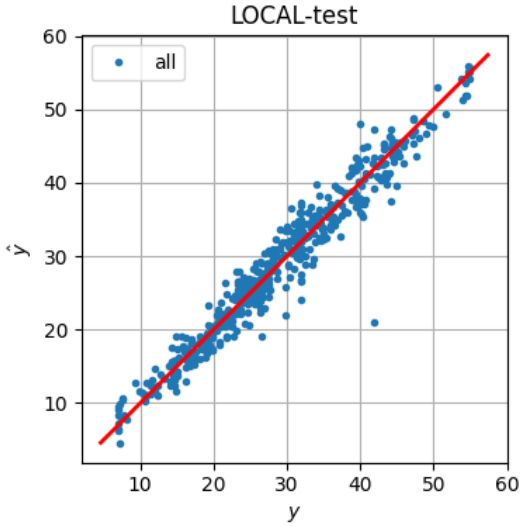
Table 7: Results obtained with local linear regression based on steepest descent with 20 different seeds and averaged

6 Comparison

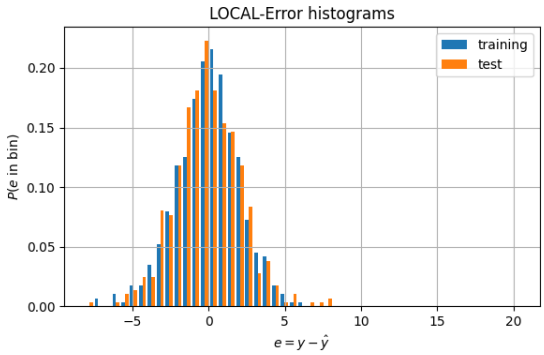
As expected there are no significant differences between Linear regression based on LLS and the one based on Steepest Descent since both methods aim to optimize the same goal,



(a) Test set MSE in relation to the change of K



(b) $\hat{\mathbf{y}}$ vs $\mathbf{y}$



(c) Local Linear Regression error histogram

Figure 4: Local Linear Regression results

but using different approaches. Indeed given a sufficient number of iterations the Steepest descent converges to the results obtained with LLS. The advantage of the steepest descent over LLS is that it ensures a faster convergence.

Local linear regression, on the other hand, obtains better results since it creates an optimal training set for each data point being tested.

7 Conclusions

In this paper, three different regressions were tested to asses UPDRS in relationship with features extracted from voice sampling.

The local linear regression based on SD achieved better results than the other methods based on the entire dataset. The accuracy of the results can be further improved by implementing the model on a larger dataset.

Better prediction of UPDRS could aid in identifying the progression of Parkinson's disease and provide specialists with additional tools to make a diagnosis.

References

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