8.5/10

Report 1: Regression on Parkinson's disease data

Abdurasul Bobonazarov, s267331

ICT for Health attended in A.Y. 2021/22

November 11th 2021

1 Introduction

Patients affected by Parkinson's disease cannot perfectly control their muscles. In particular they show tremor, they walk with difficulties and, in general, they have problems in starting a movement. Many of them cannot speak correctly, since they cannot control the vocal chords and the vocal tract.

Levodopa is prescribed to patients, but the amount of treatment should be increased as the illness progresses and it should be provided at the right time during the day, to prevent the freezing phenomenon. It would be beneficial to measure total UPDRS (Unified Parkinson's Disease Rating Scale) many times during the day in order to adapt the treatment to the specific patient. This means that an automatic way to measure total UPDRS should be developed, using simple techniques easily managed by the patient or his/her caregiver.

One possibility is to use patient voice recordings (that can be easily obtained several times during the day through a smartphone) to generate vocal features that can be then used to regress total UPDRS.

In the following, linear regression is used, with three different algorithms, and it is applied on the public dataset that can be downloaded at [1].

2 Data analysis

The 22 features available in the dataset are listed in table 1: of these, subject ID and test time are removed, total UPDRS is considered as regressand and the remaining 19 features are used as regressors. In particular, regressors include many voice parameters (jitter and shimmer measured in different ways, Noise to Harmonic Ratio NHR, etc) and motor UPDRS. The number of points in the dataset is 5875; data are shuffled and the first 75% of the points are used to train the linear model, and the remaining 25% are used to test its performance. Each regressor X_m[1] is normalized using its mean μ_m and standard deviation σ_m, measured

[1] Capital letters are used to identify random variables, small letters instead identify measured values: X_m is the random variable, x_{nm} is the n-th measured value of X_m.

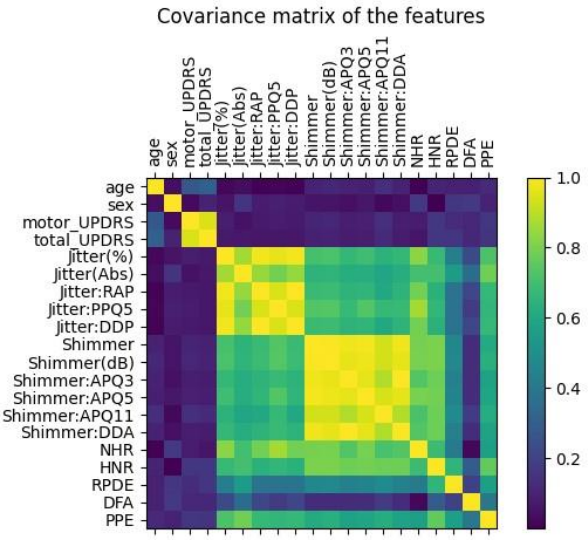


Figure 1: Covariance matrix of the features

on the training dataset:

$$X_{m,N} = \frac{X_m - \mu_m}{\sigma_m} . \tag{1}$$

Similarly, the regressand is normalized

$$Y_N = \frac{Y - \mu_Y}{\sigma_Y} . \tag{2}$$

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

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

3 Linear regression

The model assumed in linear regression is

$$Y = w_1 X_1 + \ldots + w_F X_F = \mathrm{X}^T \mathbf{w} \tag{3}$$

where Y is the regressand (total UPDRS), $\mathrm{X}^T = [X_1,\ldots,X_F]$[2] 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

$$e(\mathbf{w}) = \mathbb{E}\left\{\left[Y - \mathrm{X}^T\mathbf{w}\right]^2\right\}. \tag{4}$$

The algorithms described in the following sections use the normalized training dataset: the measured values of regressors are stored in matrix $\mathbf{X}_N$ (size of 4406×19) and the corresponding measured values of the normalized regressand are in vector $\mathbf{y}_N$.

3.1 Linear Least Squares (LLS)

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

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

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

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

where $\mathbf{X}_{N,te}$ has the size of 1469×19. 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.

3.2 Stochastic gradient algorithm with Adam optimization (SG)

Minimization of eq. (4) can be obtained iteratively using the stochastic gradient algorithm. At the i-th step

$$\hat{\mathbf{w}}_{i+1} = \hat{\mathbf{w}}_i - \gamma \nabla f_i(\mathbf{x}(i))$$

where

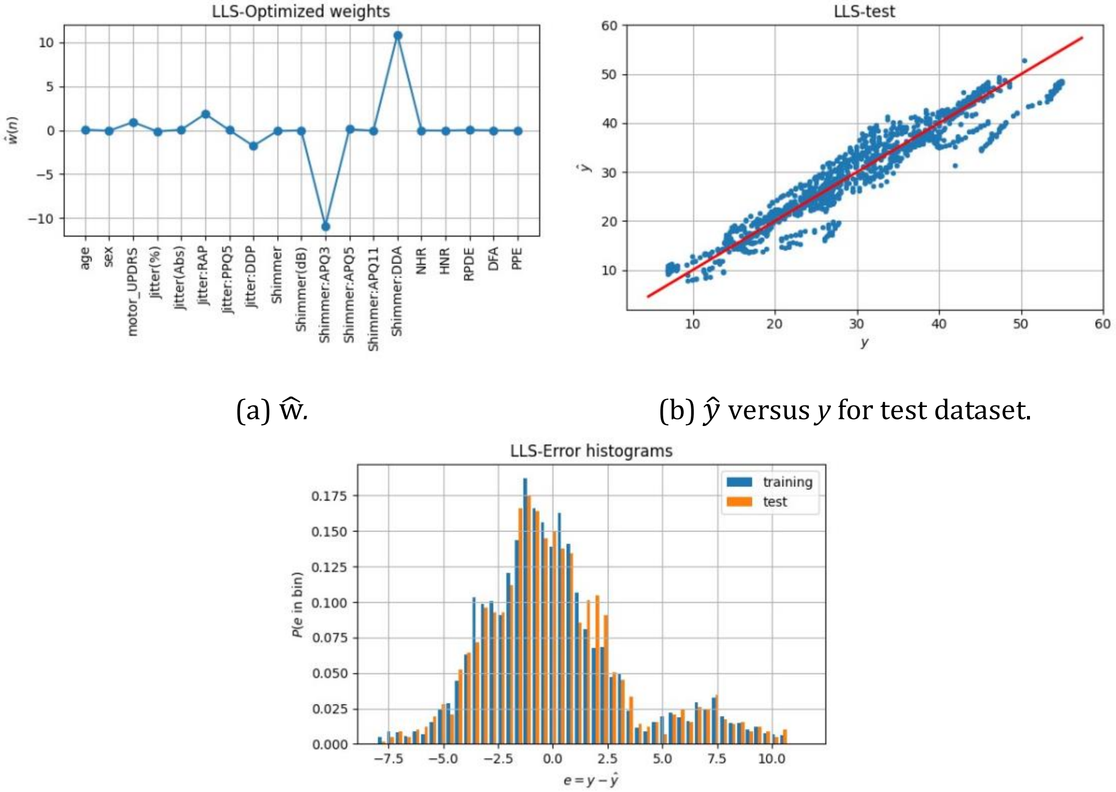
$$f_i\big(\mathbf{x}(i)\big) = [\mathrm{x}^T(i)\hat{\mathbf{w}}_i - \mathrm{y}(i)]^2, \qquad \nabla f_i\big(\mathbf{x}(i)\big) = 2[\mathbf{x}^T(i)\hat{\mathbf{w}}_i - \mathrm{y}(i)]\mathbf{x}(i)$$

being $\mathbf{x}^T(i)$ the i-th row of matrix $\mathbf{X}_N$ and $y(i)$ the i-th element of the regressand vector $\mathbf{y}_N$ for the normalized training dataset. Adam optimization was applied and therefore $\nabla f_i(\mathbf{x}(i))$ was

[2] Note that $\mathbf{x}$ is a column vector, and x^T is its transpose

substitued with its ”mean” value, according to [2], using exponential decay rates $\beta_1 = 0{,}9$ (for the mean) and $\beta_2 = 0{,}999$ (for the mean square value). The used value of the learning coefficient γ was 10^{-3} and the stopping condition for the algorithm was a number of steps sufficient to build a good linear regression model. The resulting number of iterations was 30 (epochs).

Results are shown in Figure 3.

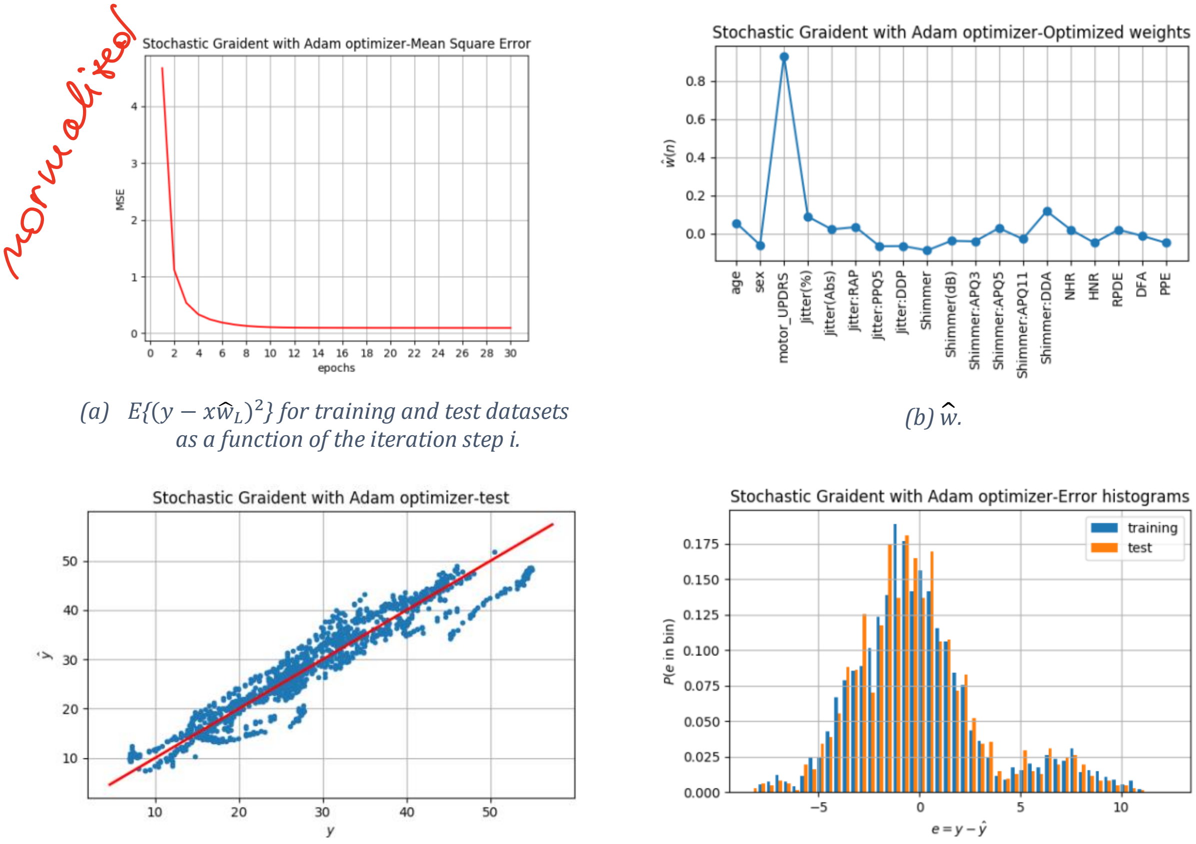


(a) $\widehat{\mathrm{w}}$.

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

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

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



(a) $E\{(y - x\hat{w}_L)^2\}$ for training and test datasets as a function of the iteration step i.

(b) $\hat{w}$.

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

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

Figure 3: Results for stochastic gradient (SG).

3.3 Gradient algorithm with minibatches (GAM)

The gradient algorithm with minibatches iteratively finds $\hat{\mathbf{w}}$ dividing the training dataset into minibatches. Being ($\mathbf{y}_i$, $\mathbf{X}_i$) the regressand and regressor values of the i-th minibatch (from the normalized matrix $\mathbf{X}_N$), the algorithm finds the next solution $\hat{\mathbf{w}}_{i+1}$ as

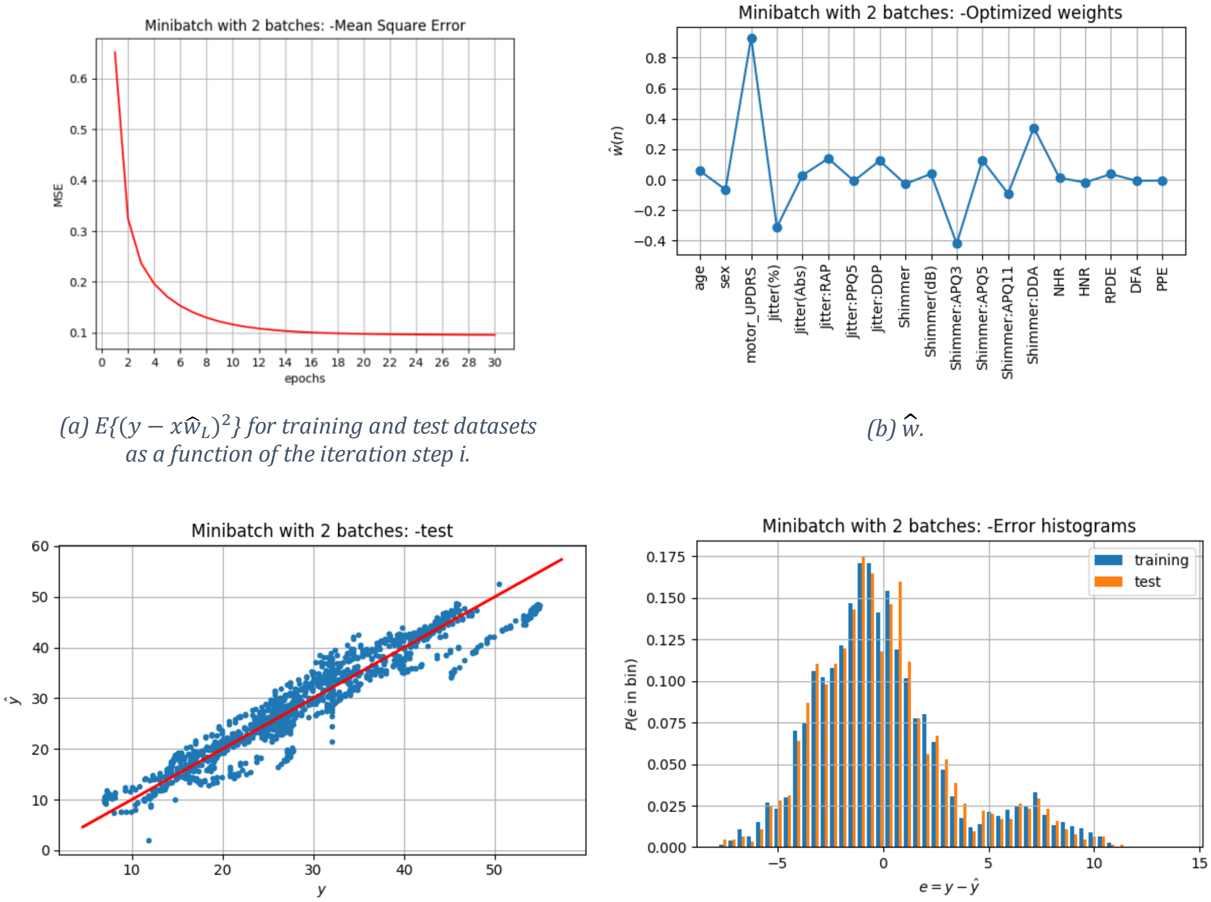
$$\hat{\mathbf{w}}_{i+1} = \hat{\mathbf{w}}_i - \gamma(2X_i^T y_i + 2X_i^T X_i \hat{\mathbf{w}}_i) \tag{8}$$

too small

where γ is the learning rate. The minibatch size was set equal to 2 because in practice, it has been observed that when using a larger batch there is a slight degradation in the quality of the model, as measured by its ability to generalize. γ was set equal to 10^{-4}, and the stopping condition was to complete a number of steps sufficient to build a good linear regression model resulting in 30 iterations (epochs). In this case adam optimization was not used. Results are shown in Figure 4.

Parameter

by you? can you give evidence?



(a) $E\{(y - x\hat{w}_L)^2\}$ for training and test datasets as a function of the iteration step i.

(b) $\hat{w}$.

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

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

Figure 4: Results for gradient algorithm with minibatches (GAM)

3.4 Numerical 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 R^2, for the three analyzed methods.

	Dataset	μ_e	σ_e	MSE	R^2
LLS	Training	3.01×10^{-13}	3.244	10.519	0.989
	Test	0.133	3.266	10.674	0.989

You see? word splits tables in different pages. Latex would never produce such a low quality result!

SG	Training Test	-6.43×10^{-15} -0.060	3.300 3.172	10.887 10.057	0.989 0.989
GAM	Training Test	2.620×10^{-14} -0.030	3.301 3.201	10.892 10.240	0.989 0.989

✓

Table 2: Comparison among the three methods: LLS, SG and GAM.

4 Conclusions

~~To sum up everything that has been stated in Table 2,~~ A all three analyzed methods provide similar performance on the regression of total UPDRS. Specifically, in all methods, the error mean is almost zero in training dataset, while it is slightly different in test dataset. Moreover, the standard deviation of regression errors is around 3.3 and 3.2, in training and test dataset, respectively. These results are fairly low enough considering that most of the time the regression error is between 3-6. On the other hand, it is possible to see a slightly different trend in Mean Square Error value, for instance, while the LLS method showed almost same value for training and test datasets, the other two methods showed around 10.89 and 10.15 for training and test datasets, respectively. However, a confirm of the accuracy, which is given by the coefficient of determination, is equal for all the three methods, 0.989.

Additionally, as the all models do not have ~~a~~ problems with overfitting, the results of the linear regression on the prediction of total UPDRS are acceptable. Nevertheless, it is important to keep in mind that it is a prediction for medical purpose, so this does not actively demonstrate that linear regression is adequate from the medical point of view.

All three analyzed methods have their advantages as well as disadvantages. It is not like a one of these methods is used more frequently than the other two. Every linear regression model is used uniformly depending on the situation and the context of the problem.

Some more comments from the medical point of view

References

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

[2] D.P.Kingma, J.Ba, ”Adam: A Method for Stochastic Optimization”, *arXiv:1412.6980 [cs.LG]*