Polytechnic University of Turin
Master course in ICT for Smart Societies

ICT for Health

Lab 1 report

Regression on Parkinson data

Sections:

Student: **Jacopo Braccio (S273999)**
Academic year: 2019

1. Introduction

1.1 Parkinson’s disease overview

Parkinson’s disease is a neurodegenerative disorder that affects predominately dopamine-producing (“dopaminergic”) neurons in a specific area of the brain called *substantia nigra*. The cause of Parkinson’s disease is still unknown, currently it’s believed that a combination of genetic changes (those with a family member affected are more likely to get the disease themselves) and environmental factors may be responsible for the condition (it's been suggested that pesticides and herbicides used in farming and traffic or industrial pollution may contribute to the condition). According to GBD (Global Burden of Disease) 6.1 million individuals worldwide were affected by Parkinson’s disease in 2016.

1.2 Parkinson’s disease symptoms and cure

Symptoms generally develop slowly over years ad their progression is often a bit different from one person to another due to the diversity of the disease. People with PD may experience tremor, slowed movement (bradykinesia), rigid muscles, loss of automatics movements and speech changes.
In order to classify the various symptoms of PD in an easy and comprehensive way, a *Unified Parkinson’s Disease Rating Scale (UPDRS)* has been created in 1987 and widely used since then. The UPDRS is composed of 42 items grouped into four subscales:

- Part I: evaluation of mentation, behaviour and mood;
- Part II: self-evaluation of the activities of daily life such as speech, swallowing, handwriting, dressing, walking and cutting food;
- Part III: motor evaluation;
- Part IV: complications of therapy.

Unfortunately, Parkinson’s disease can’t be cured, but medications can help control the symptoms, aiming to help managing problems with walking, movement and tremor. These medications increase or substitute dopamine. In particular, Levodopa, the most effective Parkinson’s disease medication, is a natural chemical that passes into the brain and is converted to dopamine. Levodopa’s effects last for some time, and then a new dose should be taken.

1.3 Goal of the lab activity

Levodopa is prescribed by a neurologist that doses the quantity that a patient should take. Since the progression of PD is continuous, it’s difficult for the neurologist to optimize the treatment. It would be useful to find a faster and less costly way to measure the UPDRS score so that the neurologist could always be up to date on the patient status. Recent studies have shown that by analysing some parameters of samples of voice (recorded during the day, also with a common smartphone) of a patient, it is possible to predict total UPDRS. Some speech symptoms related to PD could be: overall loudness level reduction, slow rate of speech, difficulties initiating speech and voice is usually tremulous. The idea of the lab is to perform regression techniques to predict UPDRS on the parameters extracted by voice recordings.

2. Data preparation and analysis

2.1 Dataset description

The dataset use for this activity was created by the University of Oxford and it is composed of different biomedical voice measurements from 42 patients affected by Parkinson's Disease. The recordings were done in six months directly in the patient's house. The dataset is a 5875x22 matrix, in which columns contains subject number, subject age, subject gender, time interval from baseline recruitment date, motor UPDRS, total UPDRS, and 16 biomedical voice measures whereas each row corresponds to a single recording of these patients.

The complete list of features of the dataset is the following: subject, age, sex, test time, motor UPDRS, total UPDRS, jitter (%), jitter (Abs), jitter:RAP, jitter:PPQ5, jitter:DDP, shimmer, shimmer(db), shimmer:APQ3, shimmer:APQ5, shimmerAPQ11, shimmer:DDA, NHR, HNR, RPDE, DFA, PPE.

2.2 Data analysis

First thing done on the dataset is its normalization in order to improve the convergence of the algorithms. Figure 2.2.1 represent the covariance matrix which explains the relationship existing between the features. The covariance is higher for features that are highly correlated. In this case, the most correlated ones (squares in yellow) are the features regarding Jitter (the interval between two times of maximum effect of voice), Shimmer (related to the amplitude variation of the sound wave) and motor UPDRS with total UPDRS (being motor UPDRS subset of total UPDRS).

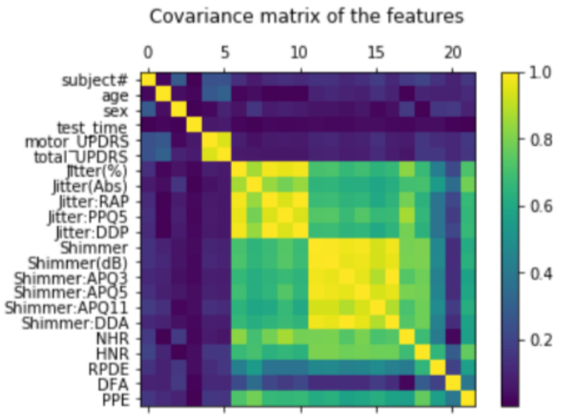


Figure 2.2.1 *Covariance matrix of the features.*

2.3 Data preparation

Features not useful to predict total UPDRS were dropped from the dataset (test_time and subject#). To prevent overfitting, the dataset has been shuffled, so that the order of the patients it's random. The regressand will be **total_UPDRS**, whereas all the other features will be the regressors.

The dataset is divided in three subsets:

- *Training (50%)*: set of data used to fit the features of the model. Through it the optimal solution for the regression problem can be found;
- *Validation (25%)*: set of data used to validate our model during training. The validation process if useful to adjust hyperparameters and to check that the model it's not affected by overfitting;
- *Test (25%)*: set of data used to assess the performance of the model (in this case, to find the total_UPDRS having the optimal solution).

The validation set will only be used in “Stochastic gradient using Adam method” and “ridge regression” algorithms.

3. Performing regression

The typical regression problem is $y(n) = [\boldsymbol{x}(n)]^T\boldsymbol{w} + v(n)$ where $\boldsymbol{x}(n)$ is the vector of the regressors (features), $\boldsymbol{w}$ is the set of weights to be found and $v(n)$ is the measurement error. Different algorithms will be used to find the optimal value of $\boldsymbol{w}$ to predict total_UPDRS ($\boldsymbol{y}$) having the features.

3.1 Linear Least Square Algorithm

The LLS method wants to find the $\boldsymbol{w}$ that minimizes the square error:

$$f(\boldsymbol{w}) = \|\boldsymbol{y} - \boldsymbol{Xw}\|^2 = [\boldsymbol{y} - \boldsymbol{Xw}]^T[\boldsymbol{y} - \boldsymbol{Xw}] = \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{Xw} - \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{Xw}$$

The gradient is evaluated ad set equal to zero:

$$\nabla f(\boldsymbol{w}) = -2\boldsymbol{X}^T\boldsymbol{y} + 2\boldsymbol{X}^T\boldsymbol{Xw} = \boldsymbol{0}$$

By solving the equation, the optimal weight vector is found:

$$\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

The optimum weight vector found during the training phase is (Figure 3.1.1):
[0.05689128, -0.05890853, 0.93525134, -0.12549652, 0.06279425, 4.89862231, -0.01384581, 4.79630332, -0.07294519, -0.01651538, -9.73921587, 0.15044839, -0.07371519, 9.72159729, -0.01296158, -0.04394021, 0.02565882, -0.01228605, -0.03004127].

The real vector $\boldsymbol{y}$ of the test dataset is compared with the estimated solution $\hat{\boldsymbol{y}}$, which is equal to the un-normalized value of $\hat{\boldsymbol{y}}_{norm} = \boldsymbol{X}_{test}\boldsymbol{w}^*$ (Figure 3.1.2).

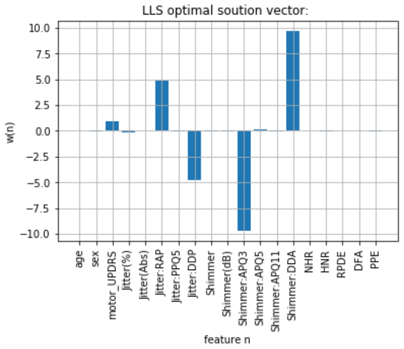


Figure 3.1.1 *Solution vector*

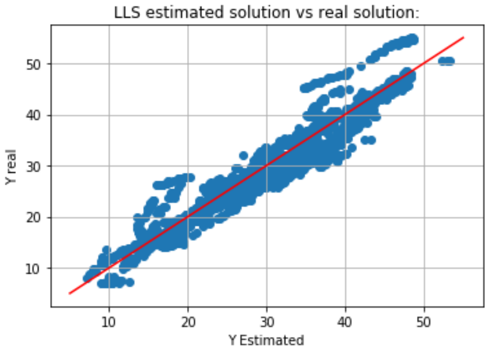


Figure 3.1.2 *Estimated test solution vs real test solution*

Figure 3.1.3 represents the histogram of the estimation error (difference between the true values of total_UPDRS and the estimated ones) un-normalized for training dataset and test dataset.

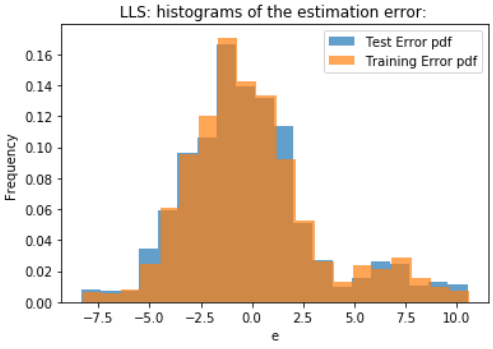


Figure 3.1.3 *Histogram of the density function of the errors*

3.2 Conjugate Gradient Algorithm

Consider the linear least square problem:

$$f(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2 = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}^T\boldsymbol{X}\boldsymbol{w} + \frac{1}{2}\boldsymbol{y}^T\boldsymbol{y} = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{Q}\boldsymbol{w} - \boldsymbol{b}^T + c$$

where $\boldsymbol{Q} = \boldsymbol{X}^T\boldsymbol{X}$, $\boldsymbol{b} = \boldsymbol{X}^T\boldsymbol{y}$ and c is a constant not influencing the value of optimum $\boldsymbol{w}^*$ that minimizes $f(\boldsymbol{w})$. By evaluating the gradient $\nabla f(\boldsymbol{w}^*) = \boldsymbol{Q}\boldsymbol{w}^* - \boldsymbol{b} = \boldsymbol{0} \rightarrow \boldsymbol{Q}\boldsymbol{w}^* = \boldsymbol{b}$.

Assuming a set f N **Q**-orthogonal vectors $\boldsymbol{d_0}, \ldots, \boldsymbol{d_{N-1}}$, we can write:

$$\boldsymbol{w}^* = \alpha_0\boldsymbol{d_0} + \cdots + \alpha_{N-1}\boldsymbol{d_{N-1}}$$

where the k-th coefficient is: $\alpha_k = (\boldsymbol{d}_k^T\boldsymbol{b})/(\boldsymbol{d}_k^T\boldsymbol{Q}\boldsymbol{d}_k)$. The solution can be found by using conjugate vectors and finding the coefficients α_k.

The algorithm starts with an initial solution $\boldsymbol{w}_0 = 0$ (at k =0) and then evaluates the gradient $\boldsymbol{g}_0 = \nabla f(\boldsymbol{w}_0) = -\boldsymbol{b} = -\boldsymbol{d}_0$. The new solution will be: $\boldsymbol{w}_1 = \boldsymbol{w}_0 + \alpha_0\boldsymbol{d}_0$. A new direction of movement $\boldsymbol{d}_1$ is required and has to be **Q**-orthogonal to $\boldsymbol{d}_0$. $\boldsymbol{d}_1$ is a linear combination of $\boldsymbol{d}_0$ and $\boldsymbol{g}_1$:

$$\boldsymbol{d}_1 = -\boldsymbol{g}_1 + \beta_0\boldsymbol{d}_0$$

where $\boldsymbol{g}_1 = \boldsymbol{Q}\boldsymbol{w}_1 - \boldsymbol{b}$ and $\beta_0 = (\boldsymbol{g}_1\boldsymbol{Q}\boldsymbol{d}_0)/(\boldsymbol{d}_0^T\boldsymbol{Q}\boldsymbol{d}_0)$.

The algorithm stops when k = N (number of features) and the optimal solution $\boldsymbol{w}^* = \boldsymbol{w}_N$.

The optimum weight vector found during the training phase is (Figure 3.2.1):

[0.05694059, -0.05906524, 0.93517743, -0.1285764, 0.06284267, 0.05304836, -0.01285586, 0.05149663, -0.0673201, -0.02003674, -0.009647, 0.15008881, -0.07392907, -0.00948088, -0.01325371, -0.04393572, 0.02570994, -0.01237694, -0.02983873].

The real vector $\boldsymbol{y}$ of the test dataset is compared with the estimated solution $\hat{\boldsymbol{y}}$ in Figure 3.2.2:

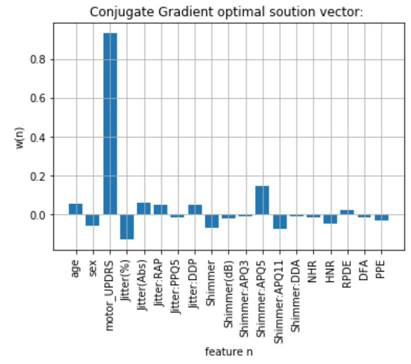


Figure 3.2.1 Solution vector

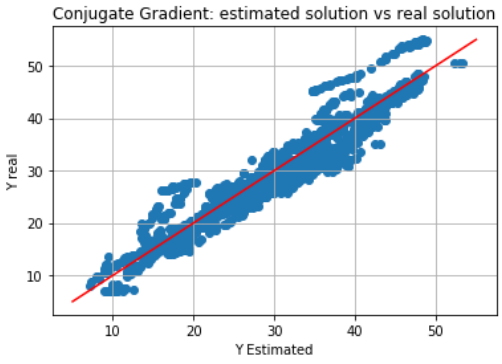
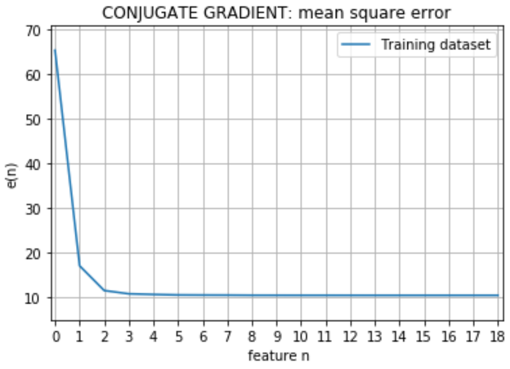


Figure 3.2.2 Estimated test solution vs real test solution

In Figure 3.2.3 it's represented the variation of the Mean Square Error of the training dataset during the 19 iterations done by the algorithm and Figure 3.2.4 represents the histogram of the estimation un-normalized for training dataset and test dataset.



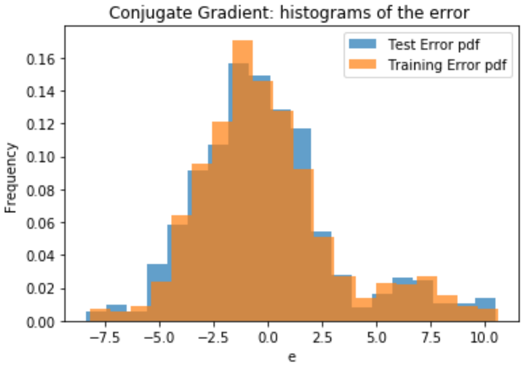


Figure 3.2.3 mean square error of the training dataset *Figure 3.2.4* Histogram of the density function of the errors

3.3 Stochastic Gradient using Adam optimizer

Stochastic gradient algorithm wants to find the optimal solution that minimizes the cost function:

$$f(\boldsymbol{w}) = \|\boldsymbol{Xw} - \boldsymbol{y}\|^2 = \sum_{n=0}^{N} [[\boldsymbol{x}(n)]^T \boldsymbol{w} - y(n)]^2 = \sum_{n=1}^{N} f_n(\boldsymbol{w})$$

The gradient of the cost function is:

$$\nabla f(\boldsymbol{w}) = \sum_{n=0}^{N} \nabla f_0(\boldsymbol{w}) = 2\sum_{n=0}^{N} [[\boldsymbol{x}(n)]^T \boldsymbol{w} - y(n)]^2 \boldsymbol{x}(n)$$

The algorithm starts with an initial vector $\boldsymbol{w}_0$ and finds the next solutions as:

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

where γ is a positive constant called *learning coefficient. "i"* is set equal to zero when it reaches the value N+1 and the loop restart until a stopping condition is met. By adding Adam optimizer to stochastic gradient, so by introducing the kth statistical moments of a random variable μ^k (in particular 1st moment and the 2nd moment, mean and mean square value), the new solution will be found as:

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i - \gamma \frac{\dfrac{\beta {\mu_{i-1}}^{(1)} + (1-\beta)[\nabla f(\boldsymbol{w}_i)]}{(1-\beta^{i+1})}}{\sqrt{\dfrac{\beta {\mu_{i-1}}^{(2)} + (1-\beta)[\nabla f(\boldsymbol{w}_i)]^2}{(1-\beta^{i+1})} + \varepsilon}}$$

where $\beta = 0.9$ for the 1st moment, $\beta = 0.999$ for the 2nd moment and $\varepsilon = 10^{-8}$ is a constant used to avoid the division by zero in the first iteration.

During the lab, the learning coefficient was set equals to 0.01. A stopping condition based on the validation dataset was chosen: when the mean square error of the validation dataset increases for 50 iterations, the optimal solution is the one at the *i-50* iteration. At this condition, the algorithm stopped at iteration number 5586 instead of 20000 (given as default). Figure 3.3.1 shows the trend of the m.s.e for both validation and training set during the iterative process (they appear to be overlapping, which means that the model fit the data of the previous dataset in the same manner).

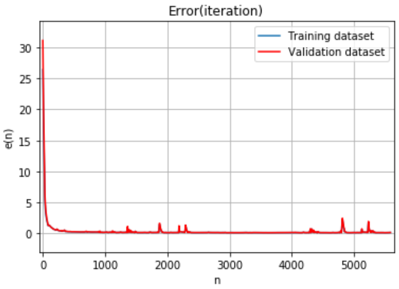


Figure 3.3.1 *Trend of m.s.e. for validation and training set*

The optimum weight vector found during the training phase is (Figure 3.3.2):
[0.00341829, -0.07275716, 0.93924621, -0.01197561, 0.10156112, 0.02276913, -0.07441154, 0.01434135, 0.3295802, -0.37016044, -0.21869517, 0.02853322, -0.02508509, 0.17149948, -0.057888, -0.13569864, -0.07408256, -0.08370503, 0.00839599]

The real vector $\boldsymbol{y}$ of the test dataset is compared with the estimated solution $\hat{\boldsymbol{y}}$ in figure 3.3.3:

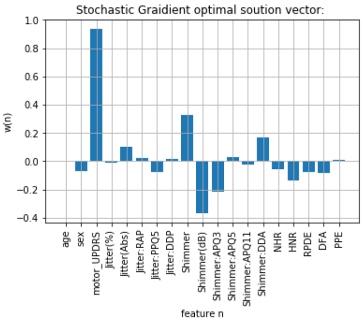


Figure 3.3.2 Solution vector

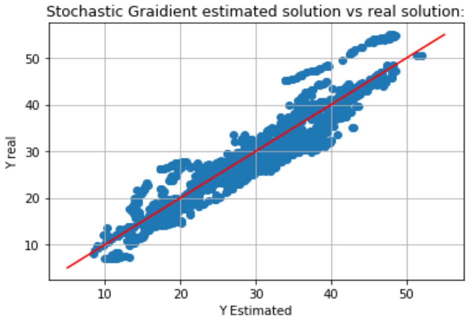


Figure 3.3.3 Estimated test solution vs real test solution

Figure 3.3.4 represents the histogram of the estimation error un-normalized for the training dataset, validation dataset and test dataset.

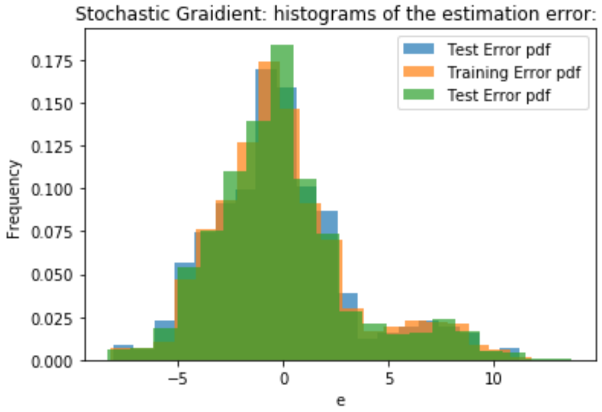


Figure 3.3.4 Histograms of the density of the error

3.4 Ridge regression

If $\boldsymbol{y} = \boldsymbol{Xw} + \boldsymbol{v}$ has some large values of noise it is possible that the vector $\boldsymbol{w}^*$takes very large values. It might be convenient to solve the new problem:

$$\min_{w}\|\boldsymbol{y} - \boldsymbol{Xw}\|^2 + \ \mu\|\boldsymbol{w}\|^2$$

where μ has to be set conveniently. If we consider $\boldsymbol{w}$ with all elements statistically independent with mean zero and variance s^2, the minimization problem is equivalent to:

$$\boldsymbol{w}^* = \min^{-1}\{\|\boldsymbol{y}_{meas} - \boldsymbol{Xw}\|^2 + \ \lambda\|\boldsymbol{w}\|^2$$

where $\boldsymbol{y}_{meas}$ are the measured values $\boldsymbol{y}$ and $\lambda = \ \sigma^2/s^2$ it's a positive parameter that can only be estimated. Different values of λ give different degrees of overfitting. Through the use of the validation set, the mean square error is calculated for a specific value of λ and the value corresponding to the smallest m.s.e. is kept.

Setting the gradient equal to 0 leads to the optimal solution:

$$\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}_{meas}$$

In the validation process λ had to be chosen in a set between 0 and 40, with a pace of 0.1. The algorithm assigned the optimal value $\lambda = \ 6.19$. Figure 3.4.1 represents the mean square error (normalized) for the validation set and the training dataset as a function of $\boldsymbol{\lambda}$**.**

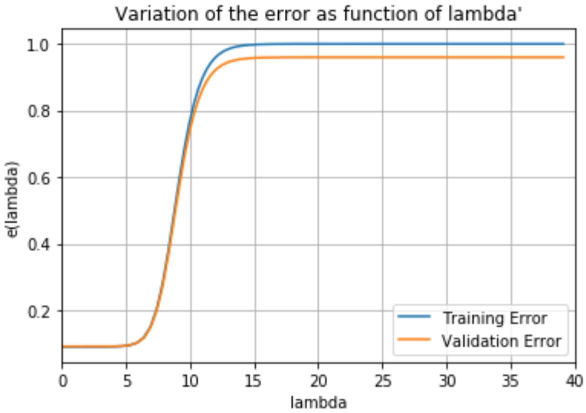


Figure 3.4.1 *Training and validation set error as function of λ*

The optimum weight vector found during the training phase is (Figure 3.4.2):
[0.0571196, -0.05900569, 0.93268705, -0.09986914, 0.05909737, 0.04309341, -0.01748858, 0.0412042, -0.05161923, -0.02786116, -0.00596746, 0.12956526, -0.06977025, -0.00577014, -0.01203217, -0.04468701, 0.0250945, -0.01246703, -0.03076672].
The real vector $\boldsymbol{y}$ of the test dataset is compared with the estimated solution $\hat{\boldsymbol{y}}$ in Figure 3.4.3:

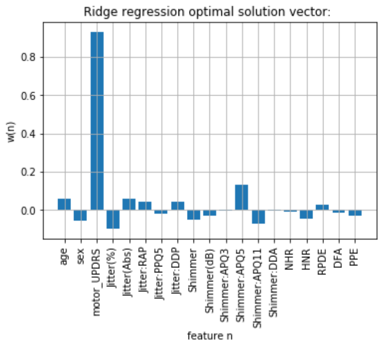


Figure 3.4.2 *Solution vector*

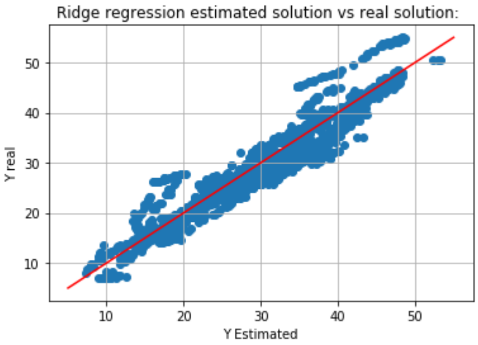


Figure 3.4.3 *Estimated test solution vs real test solution*

Figure 3.4.4 represents the histogram of the estimation error un-normalized for the training dataset, validation dataset and test dataset.

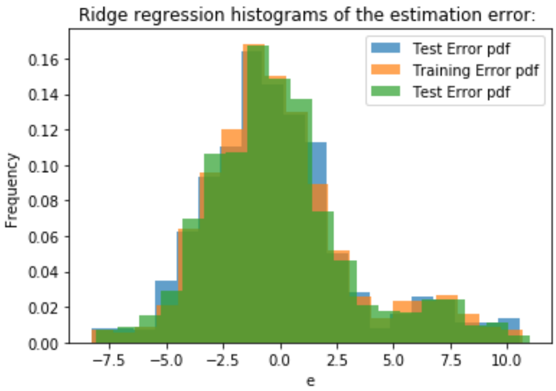


Figure 3.4.4 *Histogram of the density function of the errors*

4. Conclusions

Table A lists the values of some parameters (evaluated for every dataset) used to compare the different algorithms:

- *mean square error*: defined as $\frac{\|\boldsymbol{y}-\hat{\boldsymbol{y}}\|^2}{N}$;
- *mean of regression errors*: mean value of the regression errors $(y(n) - \hat{y}(n))$;
- *standard deviation of regression errors*: measure the amount of dispersion of the regression errors;
- *coefficient of determination R^2*: is a measure that assesses how well a model explains and predicts future outcomes and it usually ranges from 0 to 1.

	Mean square error			Mean of regression errors			Standard deviation of regression error			R^2
Algorithm	Tr.	Val.	Test.	Tr.	Val.	Test.	Tr.	Val.	Test.	
LLS	10.47	-	10.91	0	-	-0.094	3.236	-	3.302	0.91
Con. grad.	10.473	-	10.91	0	-	-0.097	3.236	-	3.302	0.91
Sto. grad.	11.886	11.952	12.342	0	0.023	-0.041	3.445	3.457	3.531	0.898
Rid. reg.	10.475	10.59	10.915	0	-0.005	-0.097	3.237	3.254	3.302	0.909

Table A

Values on Table A show that the use of the previous algorithms is fairly precise for solving the regression problem given. Considering the regression errors for every dataset, their mean value is almost zero and their standard deviation is fairly low which indicates the values tend to be close to the mean.

A confirm of the accuracy is given by the coefficient of determination whose value is almost 0.9 for every algorithm used.

The optimal solutions vector shown before are almost the same for stochastic gradient, conjugate gradient and ridge regression, and furthermore, their values respect the covariance matrix shown in figure 2.2.1: the higher weight for every vector is motor_UPDRS, that, as said before, is a subset of total_UPDRS and so highly correlated. LLS, on the other hand, has a set of values whose weights do not reflect reality.

For a rapid and still precise implementation, conjugate gradient could be chosen over the others due to its easier implementation and low number of iterations.