Assignment 01

Learning from data and related challenges and linear models for regression

Thuvaragan S. 210657G

01 October 2024

Submitted in partial fullfillment of the requirements for the module **EN3510** : **Pattern Recognition**
from *Electronics and Telecommunication Department, University of Moratuwa*

1 Data pre-processing

Feature 1 - Max-Abs Scaling

The data is sparse, with most values at or near zero there are a few large positive and negative outliers. Max-abs scaling would preserve the zero values and the relative magnitudes of the outliers, while scaling all values to a [−1, 1] range. This method maintains the sparsity of the data, which benifits certain machine learning algorithms.

Feature 2 - Standard Scaling

The data appears to have a roughly normal distribution around zero. There's a wide range of values, but no extreme outliers compared to feature 1. Standard scaling will center the data around zero and scale it to unit variance, which is appropriate for normally distributed data to maintain its overall structure.

2 Learning from data

2.1 Data Generation

The initial data generation (in listing 1) uses random number generation for both x values and epsilon values. this means each time you run the code, you get a slightly different dataset.

2.2 Data Visualization

In listing 2, the `train_test_split` function uses a random state `r = np.random.randint(104)` to split the data. This random state changes with each run, resulting in different data points being assigned to the training and testing sets each time. With that the underlying data generation process also includes random noise `epsilon`, which affects the `Y` values differently in each run.

2.3 Linear Regression

The linear regression model is observed to be different from one instance to another because, with different training data in each iteration, the model learns slightly different parameters (slope and intercept) to best fit that particular subset of data.

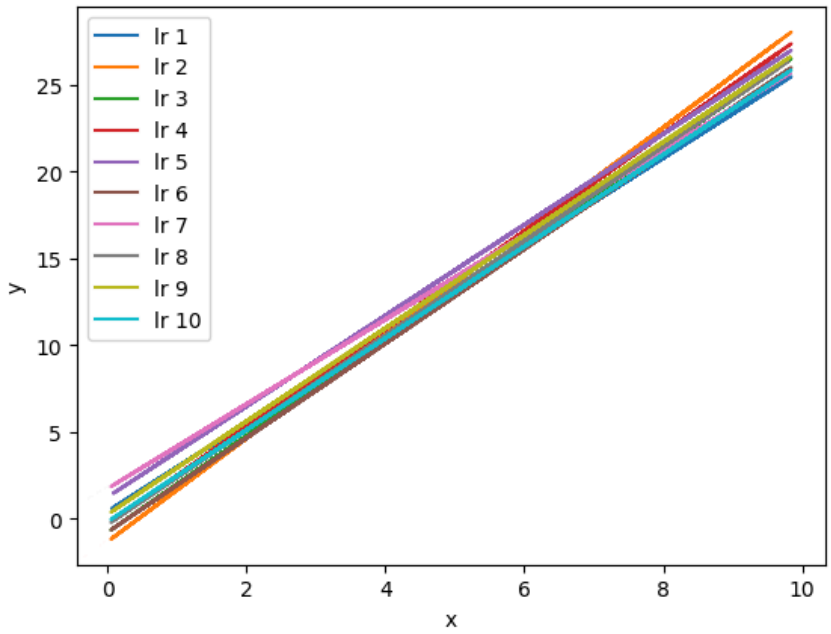


Figure 1: 10 random iterations of Linear Regression with 100 sampls

2.4 Increasing sample size

It can be observed that the linear regression models from 10000 samples are,

- More consistent
- Represent the underlying $Y = 3 + 2X + \varepsilon$ better

The reason for this different behavior is that larger sample sizes provide more information about the underlying data distribution and reduce the impact of random noise and outliers. This leads to more stable and accurate models that are less sensitive to the particular subset of data used for training. This is why usually the estimates become more efficient and consistent as the sample size increases.

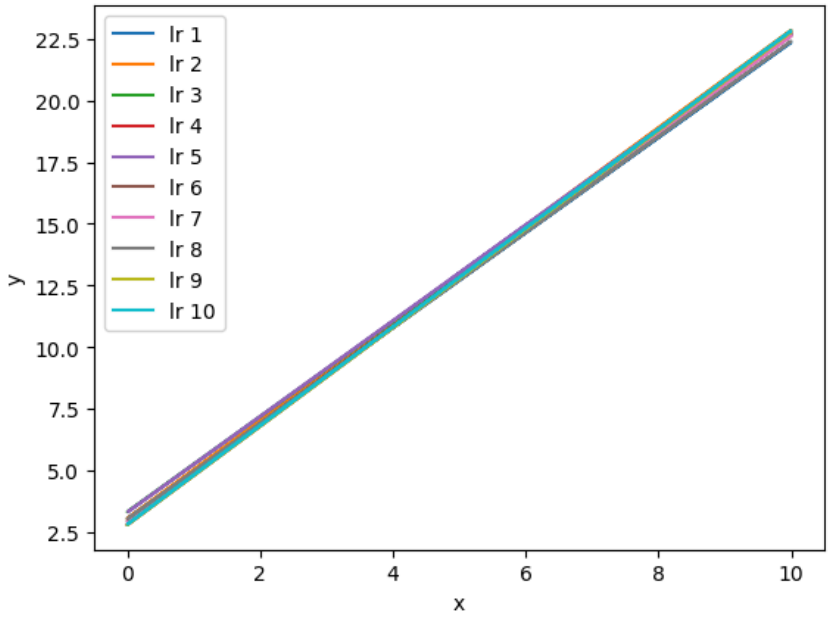


Figure 2: 10 ramdom iterations of Linear Regression with 10000 sampls

3 Linear regression on real-world data

3.1 Loading dataset

3.2 Variable analysis

The dataset is intended to be used in a regression task to predict the oral temperature using the environment information as well as the thermal image readings.

1. Independent variables (features) - **33** - Consist of gender, age, ethnicity, ambient temperature, humidity, distance, and other temperature readings from the thermal images.
2. Dependent variables (targets) - **2** - `aveOralF` and `aveOralM` (oral temperature measured in fast mode and monitor mode, respectively).

3.3 Linear regression applicability

We **cannot directly apply** linear regression to this dataset without some preprocessing steps. Categorical variables should be addressed before applying linear regression. Convert categorical variables like Gender, Age, and Ethnicity into numerical format using techniques like one-hot encoding or label encoding. We can also consider averaging the the range of values for each category to create a numerical representation. We can also not use them in the regression model if they are not relevant to the prediction.

3.4 Correct approach NaN removal

The approach is not appropriate because it handles `X` and `y` separately, which can lead to misalignment of data points. It is better to remove rows with missing values from both `X` and `y` to maintain the correspondence between input features and target values.

data cleaning can be done as follows after concatnating `X` and `y` columns.

```
# the following implementation of data cleaning is wrong
# X = X.dropna()
# y = y.dropna()

# corrected implementation
# drop rows with missing values from both X and y at the same time
data = pd.concat([X, y], axis=1)
data_cleaned = data.dropna()
X = data_cleaned[X.columns]
X.info()
y = data_cleaned[y.columns]
y.info()
```

3.5 Select features

3.6 Split data

3.7 Train a linear regression model

```
Intercept: 12.916980107531447
Coefficients:
    Feature  Coefficient
0       Age     0.001718
1     T_RC1     0.708483
2     T_atm    -0.051306
3  Humidity     0.001446
4  Distance     0.004664
```

3.8 Contribuition of features

From the selected features `T_RC1` seemed to have the most influence on the dependent feature.

3.9 Retrain and estimate coefficients

```
Intercept: 6.951744716410143
Coefficients:
      Feature  Coefficient
0       T_OR1    -0.068227
1   T_OR_Max1     0.637790
2  T_FHC_Max1    -0.048580
3   T_FH_Max1     0.320878
```

3.10 Calculate the following

```
Residual Sum of Squares: 16.109558448106856
Residual Standard Error: 0.28452162486463545
Mean Squared Error: 0.07896842376522968
R-squared: 0.7209108053457133
Standard Error of Coefficients: [
  1.37948932
  1.70083398
  1.70168529
  0.07643973
```

```
  0.08898284
]
T-values: [ 4.78065388  0.5453274  -0.21840103 -0.75100896  4.08784018]
P-values: [
  3.39193968e-06
  5.86139053e-01
  8.27340471e-01
  4.53534423e-01
  6.31488415e-05
]
```

3.11 Discarding features based on p-value

Yes, typically a lower p-value is significant for a statistical relationship. So we can safely ignore features with higher p-value `> 0.05`. We can consider removing features 2, 3 and 4 as they have higher p-values.

4 Performance evaluation of Linear regression

4.2 RSE calculation

$$\text{RSE} = \sqrt{\frac{\text{SSE}}{N}}$$

$$\text{RSE}_A = \sqrt{\frac{9}{10000}} = 0.03$$

$$\text{RSE}_B = \sqrt{\frac{2}{10000}} = 0.014$$

Since lower RSE corresponds to better model performance, model B is better.

4.3 R^2 calculation

$$R^2 = 1 - \frac{\text{RSE}}{\text{TSS}}$$

$$R^2_A = 1 - \frac{0.03}{90} = 0.99967$$

$$R^2_B = 1 - \frac{0.014}{10} = 0.99860$$

Since higher R^2 corresponds to better model performance, model A is better.

4.4 Performance metric comparision

R^2 is typically fair when comparing 2 models because it is independent of the scale of the data while also accounting for the inherent variablility in the dataset. This is not the case with RSE, which will have higher value errors for larger datasets. Even though model A and model B have the same sample size in this case, R^2 will still be better and fair campared to RSE as it accounts for the inherent variability in the dataset.

5 Linear regression impact on outliers

5.1 Modified loss functions

$$L_1(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{r_i^2}{a^2 + r_i^2}\right) = \frac{1}{N}\sum_{i=1}^{N}(L_{1,i})$$

$$L_2(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N}\left(1 - e^{-2\frac{|r_i|}{a}}\right) = \frac{1}{N}\sum_{i=1}^{N}(L_{2,i})$$

5.2 Analysis w.r.t. $a \to 0$

$$\text{as } a \to 0 \ : \ L_1(\boldsymbol{w}) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\frac{r_i^2}{r_i^2}\right) = \frac{1}{N}\sum_{i=1}^{N}1 = 1$$

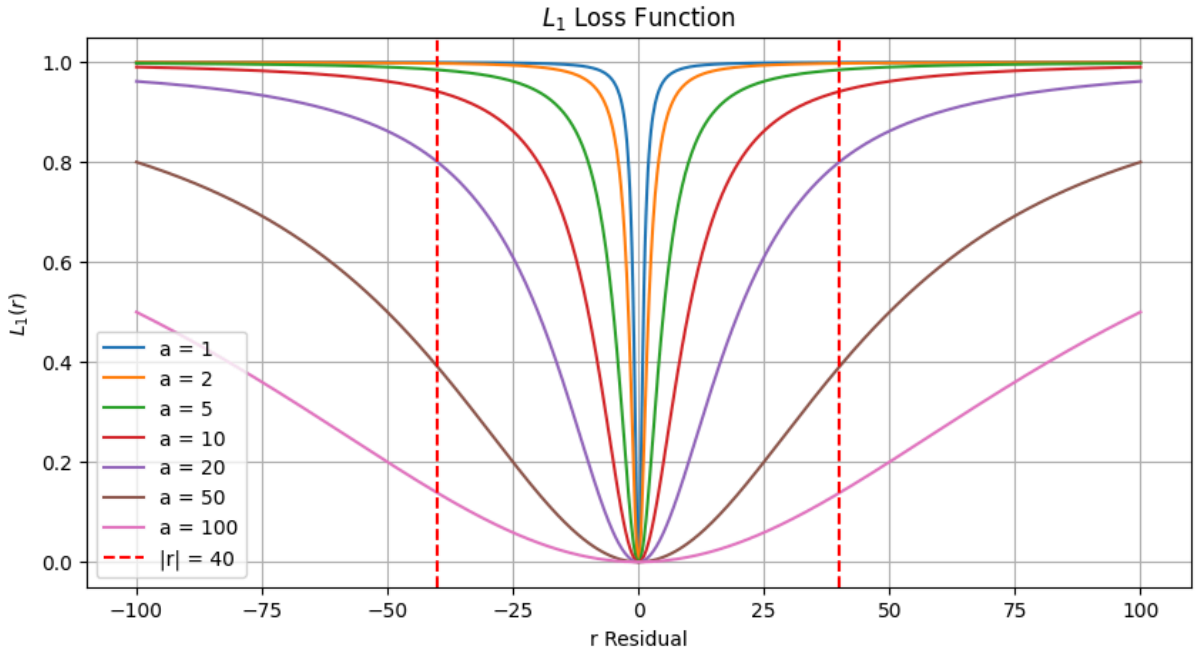
$$\text{as } a \to 0 \ : \ L_2(\boldsymbol{w}) \approx \frac{1}{N}\sum_{i=1}^{N}(1 - e^{-\infty}) \approx \frac{1}{N}\sum_{i=1}^{N}(1 - 0) = 1$$

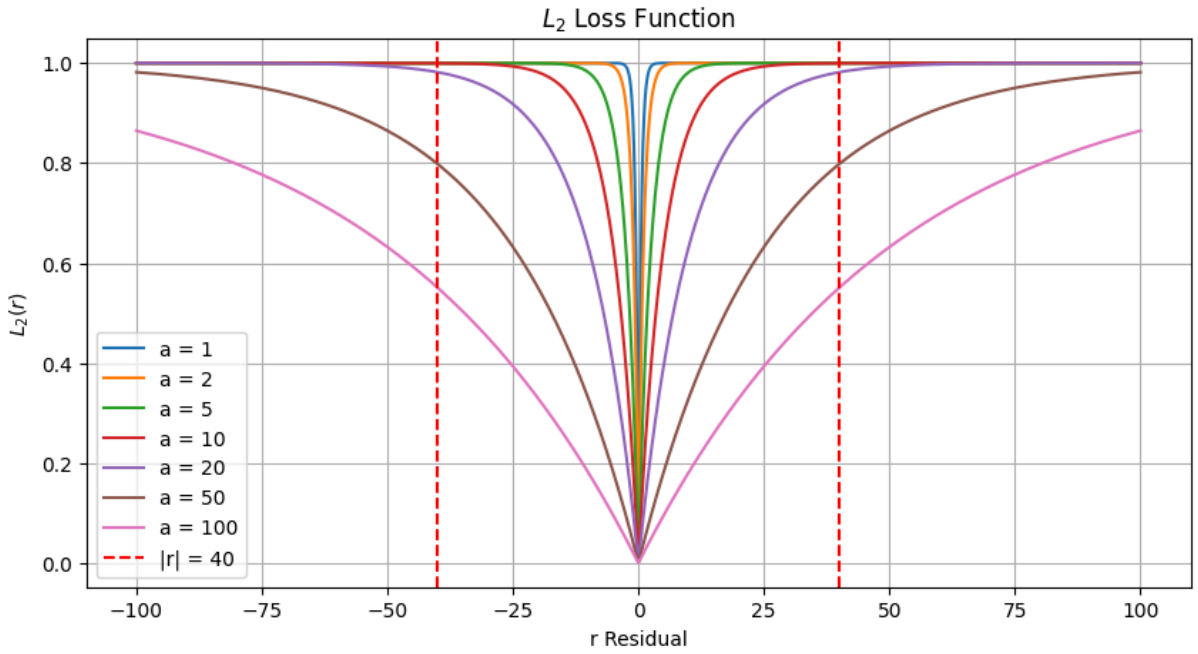
Considering the situation where residuals are relatively larger than the hyper-parameter `a` or `a` being relatively small or close to zero, both loss functions reach 1. This behaviour is contrastive of usual loss functions which are less robust in the presence of outliers. Hence it can be observed that the objective of reducing the impact of outliers is achieved through clamping the loss value as residual values increase. Compared to a standard loss function such as `MSE` which doesn't limit the effect of outliers at all, this will reduce the impact of outliers in the dataset.

5.3 Choosing appropriate loss function

Analysing the loss functions reveals that both functions are good at handling residuals. $L_2(w)$ can be choosen on the basis of aggressive clamping of residual values using exponential scaling.

A relatively lower value of `a` will suffice for $L_2(w)$ in the range of 5 to 20 to have a balance between clamping and not restricting data too much.




L_2 Loss Function
1.0
0.8
0.6
0.4
0.2
0.0
$L_2(r)$
a = 1
a = 2
a = 5
a = 10
a = 20
a = 50
a = 100
|r| = 40
−100
−75
−50
−25
0
25
50
75
100
r Residual