EN3150 Assignment 02: Learning from data and related challenges and classification

Sampath K. Perera

Aug. 19, 2025

1 Linear Regression

1. A set of data points $(x_i,\ y_i)$ are known to form a line. The ordinary least squares (OLS) is performed on this dataset. The OLS minimizes a loss function which is defined as $\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$ with y_i and $\hat{y}_i$ are true and OLS outputs, respectively. The fitted OLS line and data points are shown in Figure 1. It is observed that the OLS fitted line is not aligned to majority of data points.
   What is the reason behind this? [10 marks]

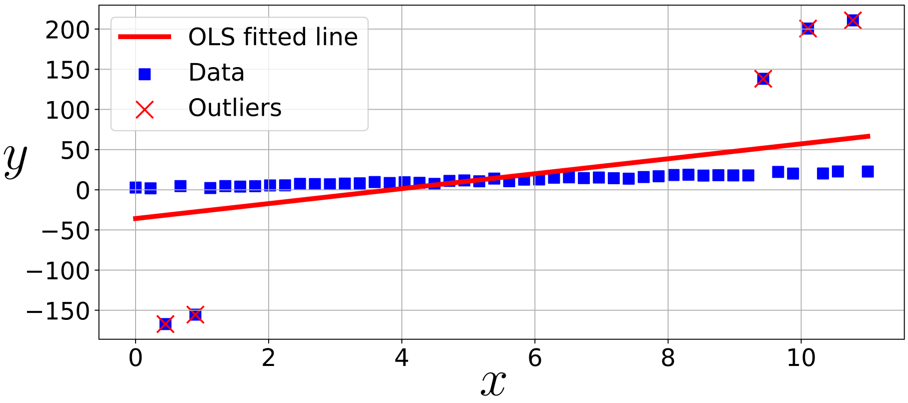

Figure 1: Ordinary least squares fit on data.

2. To reduce the impact of outliers, a modified loss function is introduced. It is given as $\frac{1}{N}\sum_{i=1}^{N}a_i(y_i - \hat{y}_i)^2$. There are two schemes are proposed for setting a_i.
   - Scheme 1: for outliers $a_i = 0.01$ and for inliers $a_i = 1$,
   - Scheme 2: for outliers $a_i = 5$ and for inliers $a_i = 1$.

Under which scheme do you expect a better fitted line for inliers than the OLS fitted line in Figure 1. Justify your answer. [30 marks]

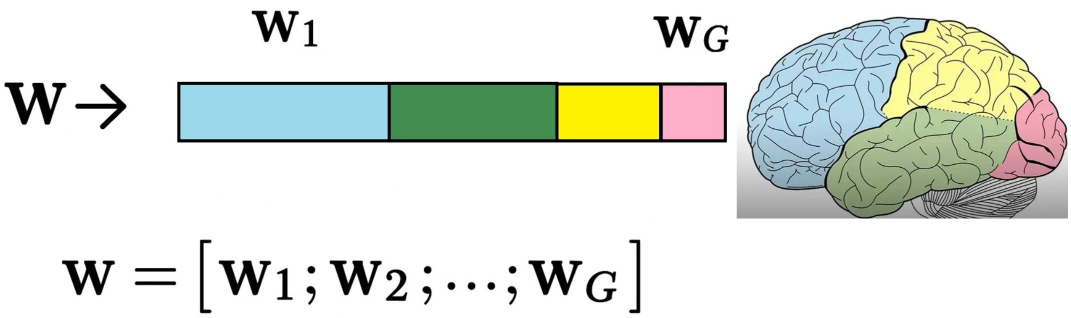
$\mathbf{w}_1$
$\mathbf{w}_G$
$\mathbf{w} \rightarrow$
$\mathbf{w} = [\mathbf{w}_1; \mathbf{w}_2; \ldots; \mathbf{w}_G]$

Figure 2: Simple Brain segmentation.

3. In brain image analysis (e.g., fMRI), the brain is divided into multiple regions as show in Figure 2, each consisting of many voxels (pixels). A researcher wants to identify which brain regions are most predictive of a specific cognitive task.
   Why linear regression is not suitable algorithm for the above task ? [20 marks]

4. Next, the following two methods are being considered:

   - **Method A**: Standard LASSO, which selects individual voxels independently. The lasso objective is to minimize:

   $$\min_{\mathbf{w}} \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \mathbf{w}^\top \mathbf{x}_i \right)^2 + \lambda \|\mathbf{w}\|_1 \right\}$$

   - **Method B**: Group LASSO. The group lasso objective is to minimize:

   $$\min_{\mathbf{w}} \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \mathbf{w}^\top \mathbf{x}_i \right)^2 + \lambda \sum_{g=1}^{G} \|\mathbf{w}_g\|_2 \right\}$$

   where $\mathbf{w}_g$ is the sub-vector of weights corresponding to group g, and G is the number of groups (e.g., brain regions).

5. Which method (LASSO or group LASSO) is more appropriate in this setting, and why? [40 marks]

2 Logistic regression

1. Use the code given in listing 1 to load data.

2. Now, use the code given in listing 2 to train a logistic regression model. Here, did you encounter any errors? If yes, what were they, and how would you go about resolving them ? [20 marks]

3. Why does the saga solver perform poorly? [15 marks]

4. Now change the solver to "liblinear" by using logreg = LogisticRegression(solver='liblinear'). What is the classification accuracy with this configuration? [5 marks]

5. Why does the "liblinear" solver perform better than "saga" solver ? [15 marks]

6. Explain why the model's accuracy (with saga solver) varies with different random state values ? [15 marks]

7. Compare the performance of the "liblinear" and "saga" solvers with feature scaling. If there is a significant difference in the accuracy with and without feature scaling, what is the reason for that.
   You may use Standard Scaler available in sklearn library. [15 marks]

8. Suppose you have a categorical feature with the categories 'red', 'blue', 'green', 'blue', 'green'. After encoding this feature using label encoding, you then apply a feature scaling method such as Standard Scaling or Min-Max Scaling. Is this approach correct? or not?. What do you propose ? [15 marks]

```
import seaborn as sns
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# Load the penguins dataset
df = sns.load_dataset("penguins")
df.dropna(inplace=True)
# Filter rows for 'Adelie' and 'Chinstrap' classes
selected_classes = ['Adelie', 'Chinstrap']
df_filtered = df[df['species'].isin(selected_classes)].copy()
    # Make a copy to avoid the warning
# Initialize the LabelEncoder
le = LabelEncoder()
# Encode the species column
y_encoded = le.fit_transform(df_filtered['species'])
df_filtered['class_encoded'] = y_encoded
```

```
# Display the filtered and encoded DataFrame
print(df_filtered[['species', 'class_encoded']])
# Split the data into features (X) and target variable (y)
y = df_filtered['class_encoded']  # Target variable
X = df_filtered.drop(['class_encoded'], axis=1)
```

Listing 1: Data load and preprocessing.

```
#Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.2, random_state=42)
Train the logistic regression model. Here we are using saga
    solver to learn weights.
logreg = LogisticRegression(solver='saga')
logreg.fit(X_train, y_train)
# Predict on the testing data
y_pred = logreg.predict(X_test)
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print(logreg.coef_, logreg.intercept_)
```

Listing 2: Training LR model.

3 Logistic regression First/Second-Order Methods

1. Use the code given in listing 3 to generate data. Here, variable y and X are class labels and corresponding feature values, respectively.

2. Implement batch Gradient descent to update the weights for the given dataset over 20 iterations. State the method used to initialize the weights and reason for your selection. [20 marks]

3. Specify the loss function you have used and state reason for your selection. [5 marks]

4. Implement Newton's method to update the weights for the given dataset over 20 iterations. [20 marks]

5. Plot the loss with respect to number of iterations for batch Gradient descent and Newton method's in a single plot. Comment on your results. [25 marks]

6. Propose two approaches to decide number of iterations for Gradient descent and Newton's method. [10 marks]

7. Suppose the centers in in listing 3 are changed to centers = [[2, 2], [5, 1.5]]. Use batch Gradient descent to update the weights for this new configuration. Analyze

the convergence behavior of the algorithm with this updated data, and provide an explanation for convergence behavior. [20 marks]

```
import numpy as np
import matplotlib.pyplot as plt

import numpy as np
from sklearn.datasets import make_blobs
# Generate synthetic data
np.random.seed(0)
centers = [[-5, 0], [5, 1.5]]

X, y = make_blobs(n_samples=2000, centers=centers, random_state=5)
transformation = [[0.5, 0.5], [-0.5, 1.5]]
X = np.dot(X, transformation)
```

Listing 3: Data generation.

4 Additional Resources

1. Introduction to sparsity in signal processing
2. Sklearn Lasso
3. Sklearn linear regression
4. Sklearn logistic regression
5. Tibshirani, Robert. "Regression shrinkage and selection via the lasso." Journal of the Royal Statistical Society Series B: Statistical Methodology 58.1 (1996): 267-288.
6. Meier, Lukas, Sara Van De Geer, and Peter Bühlmann. "The group lasso for logistic regression." Journal of the Royal Statistical Society Series B: Statistical Methodology 70.1 (2008): 53-71.
7. Yuan, Ming, and Yi Lin. "Model selection and estimation in regression with grouped variables." Journal of the Royal Statistical Society Series B: Statistical Methodology 68.1 (2006): 49-67.

5 Submission

- Upload a report as a pdf file named as "your_indexno_EN3150_A02.pdf". Include the index number and the name within the report as well. Please include all your answers in the report.
- Pay careful attention to formatting such as font size, spacing, and margins.

- Include a title page with necessary information (e.g., title, author, date, index no).
- Use consistent and professional formatting throughout the document.
- Plagiarism will be checked and in cases of plagiarism, an extra penalty of 50% will be applied. In case of copying from each other, both parties involved will receive a grade of zero for the assignment. Academic integrity is of utmost importance, and any form of plagiarism[1] or cheating will not be tolerated.
- An extra penalty of 15% is applied for late submission.

References

[1] Robert Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 58, no. 1, pp. 267–288, 1996.

[2] Lukas Meier, Sara Van De Geer, and Peter Bühlmann, "The group lasso for logistic regression," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 70, no. 1, pp. 53–71, 2008.

[3] Ming Yuan and Yi Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 68, no. 1, pp. 49–67, 2006.

[1] https://en.wikipedia.org/wiki/Plagiarism