Assignment 2 - HWS24

Logistic Regression

Anh Tu Duong Nguyen 2115931 - Ilias Login: anguyea
&
Anh-Nhat Nguyen 2034311 - Ilias Login: anhnnguy

November 3, 2024

Contents

1 Dataset Statistics

1.1 a

Kernel density estimation (KDE) visualizes a probability density function from observed data. In this dataset, however, the first 54 features are relative frequencies, while the last three are absolute values, leading to different scales. As a result, Figure 3, which shows KDEs for all features, primarily indicates that most density is near zero, suggesting most values are small across both feature types, but provides little additional insight.

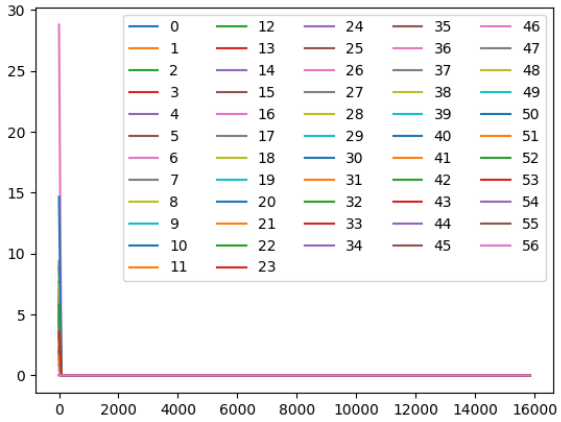
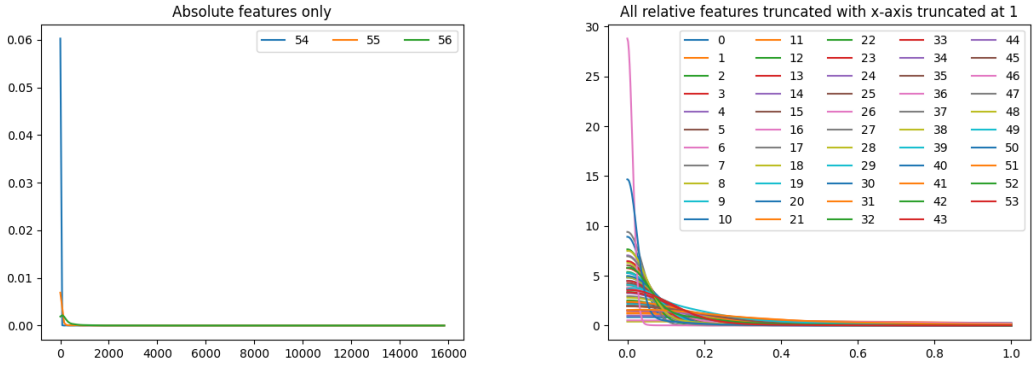


Figure 1: KDE Plot for all features

The kernel density plots in Figures 2a and 2b illustrate the distribution of features with different scaling, presented with truncated x-axes. Figure 2b reveals that for relative features, the majority of density lies within the 0 to 0.1 range, and this pattern holds for absolute features in Figure 2a as well. This concentration suggests a significant right skew in our data. To facilitate comparative visualization of these scaled features in a single plot, normalization is a logical step.



(a) KDE Plot for absolute features with truncated x-axis

(b) KDE Plot for relative features with truncated x-axis

Figure 2: Comparison of the different features

1.2 b

To ensure consistency, the test data must be normalized using the training set's mean and standard deviation. Using the test set's own statistics would result in different scales between training and test data, making accurate classification difficult.

1.3 c

When finish normalization, each feature has a mean of zero, concentrating most density around zero, as shown in Figure 4. This also introduces negative values, while the data remains right-skewed, though with less density clustered tightly at zero. This suggests that normalization has reduced the influence of numerous small observations. Now, as discussed in subsection 1.1, all features (relative and absolute) can be plotted together. Extending the x-axis range reveals local maxima around 7, likely due to the compressed data range from normalization.

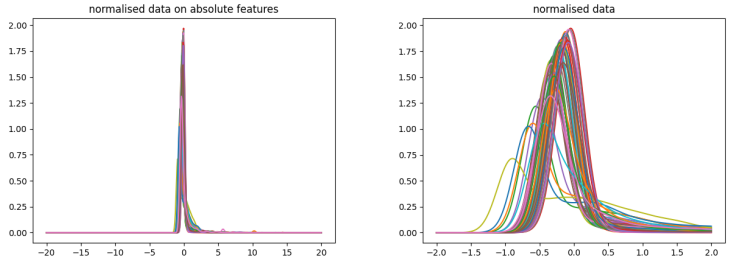


Figure 3: KDE plot for all normalized features

2 Maximum Likelihood Estimation

2.1 a

To show that rescaling (multiplying by a constant a) and shifting (adding a constant b) features do not affect Maximum Likelihood (ML) estimates in logistic regression, we use the log-likelihood function with these transformations:

$$p(y \mid aX + b, w) = \sum_{i=1}^{N} \sigma \left(w_0 + \sum_{j=1}^{D} w_j (a \cdot x_{ij} + b) \right)$$

Expanding this, we get:

$$= \sum_{i=1}^{N} \sigma \left(\left(w_0 + \sum_{j=1}^{D} w_j \cdot b \right) + \sum_{j=1}^{D} w_j \cdot a \cdot x_{ij} \right)$$

Here:

- **Shifting**: Adding b only shifts the bias w_0 by a constant $\sum_{j=1}^{D} w_j \cdot b$, which doesn't affect the overall log-likelihood.
- **Scaling**: Multiplying by a scales each weight w_j proportionally, preserving the log-likelihood due to the convex nature of the function.

Therefore, ML estimates remain unchanged with rescaling and shifting when using a bias term.

While not necessary for ML, z-scores (mean 0, variance 1) are helpful because they:

- **Improve Stability and Interpretability**: Standardized weights are comparable across features.
- **Support MAP Estimation**: Z-scores prevent large-scale features from disproportionately influencing regularized weights, enhancing MAP's effectiveness.

Thus, normalizing with z-scores improves model performance and interpretability in both ML and MAP estimation.

2.2 b to d

In code

2.3 e

As illustrated in Figure 5, Stochastic Gradient Descent (SGD) approaches the optimum rapidly, reaching near-optimal values within just 10 epochs. By the 15th epoch, there is no notable change in the objective function value. In contrast, Gradient Descent (GD)

requires more time to converge but ultimately achieves a closer proximity to the global optimum than SGD.

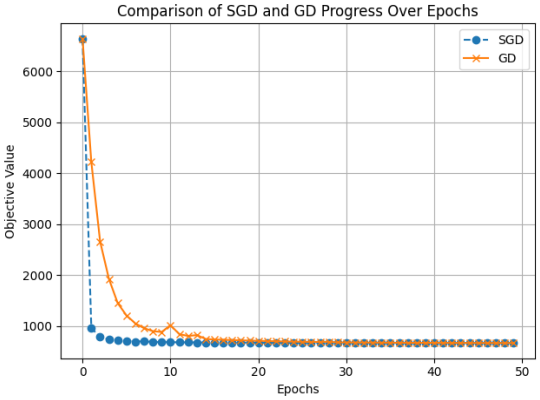


Figure 4: SGD and GD Comparison with Epoch

Figure 5 illustrates the progression of different components in the gradient descent (GD) and stochastic gradient descent (SGD) optimization processes. The top-left plot (vz_gd) shows the decline in the loss function over epochs, indicating that SGD achieves faster convergence towards an optimal value compared to GD, as it quickly reduces the loss within the initial epochs. The top-right plot (wz_gd) represents the fluctuation in the weights, with SGD exhibiting more variability around the optimal region, characteristic of its stochastic nature. The bottom plot (ez_gd) reveals the consistent decrease in objective values with increased epochs, showcasing the converging behavior. This suggests that while SGD achieves a near-optimal value quickly, it stabilizes with some oscillations, whereas GD steadily continues to improve but requires more epochs for full convergence. Consequently, there exists a trade-off between GD and SGD, where SGD offers faster initial convergence, making it suitable for scenarios with limited computational resources, while GD provides a more stable convergence over extended epochs

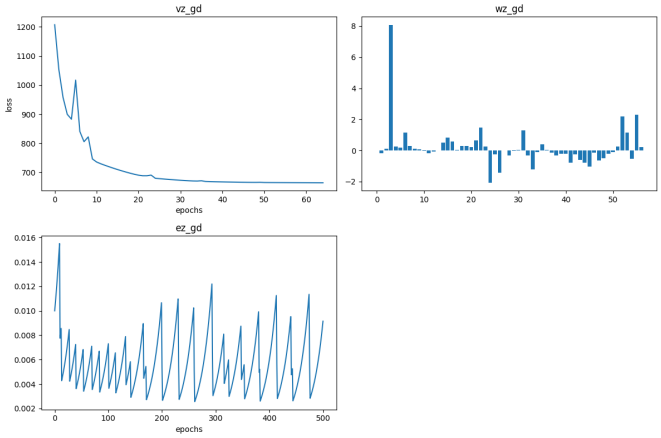


Figure 5: Learning rate of GD and SGD

Figure 5 also reveals that, aside from the fourth feature ("word freq 3d"), the final weights for most features are largely similar across methods. Both methods prioritize this feature, though the assigned weights differ substantially due to SGD's randomization. This emphasis on the fourth feature aligns with expectations, as "3d" frequently appears in spam instances within our training data.

3 Prediction

In Logistic Regression, an unseen record is classified by computing the original probability θ from the *logit scores* using the *logistic function*. These logit scores are assumed as a linear function, more precisely a linear combination of the weight vector and the D-dimensional feature vector X. The *logistic function* outputs a value for the original probability θ in the interval of $[0, 1]$. This is displayed in Figure 6. What is also evident from the plot is that the decision boundary for the classification is set at $\sigma(\eta) = 0.5$, which was implemented accordingly in the code.

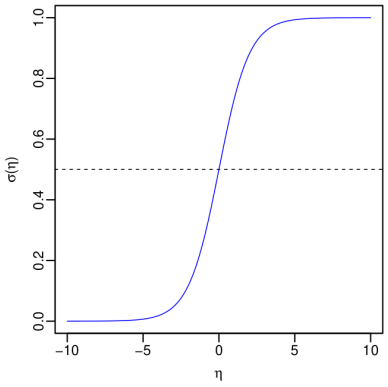


Figure 6: Logistic Function Plot

The implemented algorithm performs well, with an F1-Score of 0.89 for the positive class and 0.93 for the negative class. The Precision recall curve shows a tradeoff between the two metrics. By lowering the decision, the Recall gets maximized, but the Precision suffers from this, likewise, if the decision boundary is increased then the Precision gets maximized but the Recall suffers from this. To illustrate this tradeoff better, consider the following setting: With a small decision boundary, the Recall is maximized, meaning that the number of False Negatives is reduced, and therefore, it would be ensured that the number of correctly classified spam-mails is maximized, however since the Precision suffers from this, mails that are actually not a spam-mail would accidentally be classified as a spam-mail

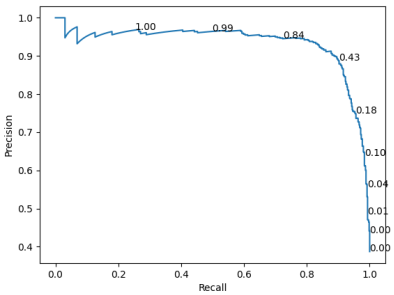


Figure 7: Precision-Recall Curve

Figure 7 displays this behavior tradeoff by plotting the Precision and Recall for different values of the decision boundary. Figure 7 shows that 0.5 is a reasonable choice for the decision boundary as it is a knee-value maximizing both Precision and Recall simultaneously.

To obtain interpretable values for the weights, odds ratios for a given weight are calculated by using the exponential function:

$$\frac{odss_{w_j}}{odds} = exp(w_j) \tag{1}$$

These interpretable values are then plotted in a scatterplot with the values sorted ascendingly according to the values. Figure 8 illustrates this scatterplot and what can be observed is that there is a spike at the feature **word_freq_3d**. For the Gradient Descent, the odds ratio is approximately 3249.5, meaning that a one-unit increase would lead to an increase of 3249.5 for the odds ratio.

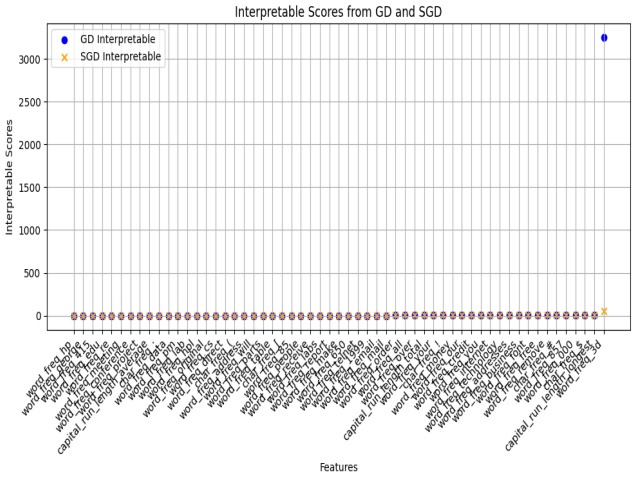


Figure 8: Odds Ratio Plot for every feature

By taking a look into the dataset, the reason behind this can be unraveled. The feature is only ever observed for a total of 35 times in the training set and in those cases, the corresponding email was spam 29 times. In a real-world scenario, the word '3d' does not appear that frequently in spam-mails compared to the training dataset. This could be a sign for overfitting. When looking at the subsequent most important features **capital_run_length_longest, char_freq_$, word_freq_000**, then these features are more intuitive as spam-mails are created by making use of more frequently used characters. So to mitigate overfitting, it is best to use a prior.

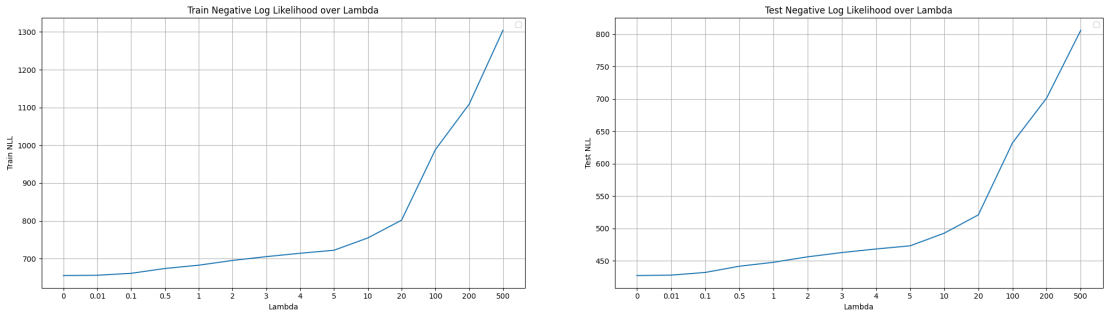
4 Maximumm Aposteriori Estimation

4.1 Gradient Descent for MAP estimation

cf. code

4.2 Effect of Prior

In this subsection, the effect of the prior is studied. Figure 9 display plots of the development for the final Negative Log Likelihood for different values of λ for both the Training and the Test set.



(a) Train NLL (b) Test NLL

Figure 9: Comparison of Train and Test NLL

What is evident from those plots is that the prior will get a higher influence on the results with an increase in the regularization parameter λ. This is applicable to both the training and the test set. This means that the more weight is put on the prior with a higher λ, the more the observations are obscured. In terms of accuracy, the accuracy of the predictions increases slightly with increases in λ at first, but with a sufficiently high λ the accuracy begins to drop drastically which is displayed in Figure 10.

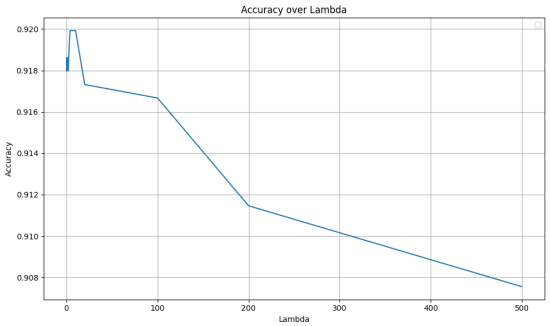


Figure 10: Accuracy for different values of λ

4.3 Weight Vector Composition

Figure 11 displays the composition of the weight vectors for different λ. What can be observed is that with increasing λ, the weights become more and more similar to each other and move close to zero. With this the effect of the feature ”3d” gets reduced and therefore overfitting was mitigated which leads to a better performance in terms of accuracy for certain values of λ.

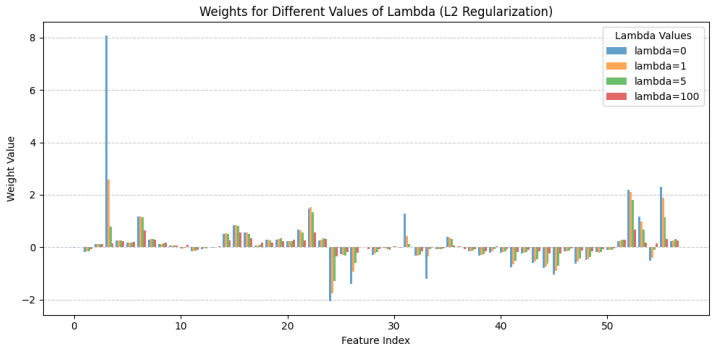


Figure 11: Weight Vectors Composition for different values of λ

Ehrenwörtliche Erklärung

Ich versichere, dass ich die beiliegende Bachelor-, Master-, Seminar-, oder Projektarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und in der untenstehenden Tabelle angegebenen Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Declaration of Used AI Tools

Tool	Purpose	Where?	Useful?
ChatGPT	Rephrasing	Throughout	+
DeepL	Translation	Throughout	+
ResearchGPT	Summarization of related work	Sec. **??**	-
Dall-E	Image generation	Figs. 2, 3	++
GPT-4	Code generation	functions.py	+
ChatGPT	Related work hallucination	Most of bibliography	++

Unterschrift
Mannheim, den XX. XXXX 2024