Machine Learning (HWS24)

Assignment 2: Logistic Regression

Nursultan Mamatov, nmamatov, 1983726
Cagan Yigit Deliktas, cdelikta, 1979012

November 3, 2024

1 Task 1: Dataset Statistics

1.1 Part a: Look at the kernel density plot (code provided) of all features and discuss what you see (or don't see).

Most of the density values concentrated near zero on the left side of the x-axis because all features are not scaled. Some features have much larger values than others, thus, they dominate the features with small values. Thus, it is hard identify individual kernel density plots for each feature. Observing structures and specific patterns is not possible in this setting.

Selecting Three Unnormalized Features: In Figure 1, we only selected 3 features to make the plot clearer. With fewer features, we can more clearly see their individual density curves. The plot indicates that these three features have unique density shapes, but they all show a similar trend of high density close to zero and a steep drop-off.

1.2 Part c: Redo the kernel density plot on the normalized data.

KD Plot with Normalized Features: After normalization, as seen in Figure 2, the features are now standardized, meaning they have similar scales. The range of the x-axis has decreased, and no features are highly dominant. Most features are clustered close to zero and gradually taper off as we move to the right. This allows for clearer comparisons between the density shapes of different features.

Selecting Three Normalized Features: In Figure 3, we can see that "word_freq_adress" (yellow line) has a strong tendency to take on values close to zero because its density is highest around zero. As we approach one on the x-axis, "word_freq_all" (green line) has a density of around 0.1, whereas "word_freq_adress" approaches zero. This suggests

that the feature represented by the green line has more observations around the value of 1 compared to the yellow line.

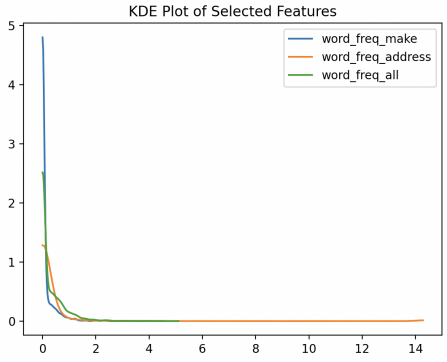

Figure 1: KD Plot of Three Unnormalized Features

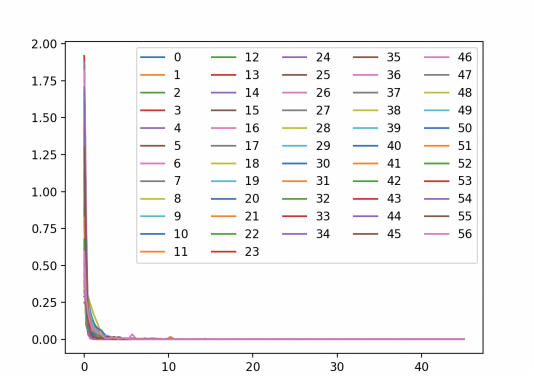

Figure 2: KD Plot of Normalized Features

2 Task 2: Maximum Likelihood Estimation

2.1 Part a: Same likelihood with a bias term, rescaling, and shifting.

Bias and Shifting: Adding a bias term and shifting the features do not change the overall distribution of the output; therefore, the MLE remains the same.

Multiplying by a constant: If we multiply X by c, it becomes $X' = Xc$, and we can define $\theta' = \frac{1}{c}\theta$ so that the MLE remains the same.

Distance based algorithms: Computing z-scores standardizes the features to have a mean of zero and a standard deviation of one. This ensures that all features contribute equally to the distance calculations. This is important when using ML algorithms such as k-NN and SVM, which are based on distance metrics.

Influence on Gradient Descent: Also, in gradient descent, applying z-scoring can enhance convergence rates by standardizing the features to a similar scale, which minimizes the likelihood of any single feature overshadowing the learning process. In addition, the overflow problem can be prevented by using z-scores.

2.2 Part e: Explore the behavior of both methods for the parameters provided to you.

As we can see from Figures 4 and 5, SGD has a lower negative log-likelihood than GD when epoch = 29. This means that it reaches a more optimal point faster than GD.

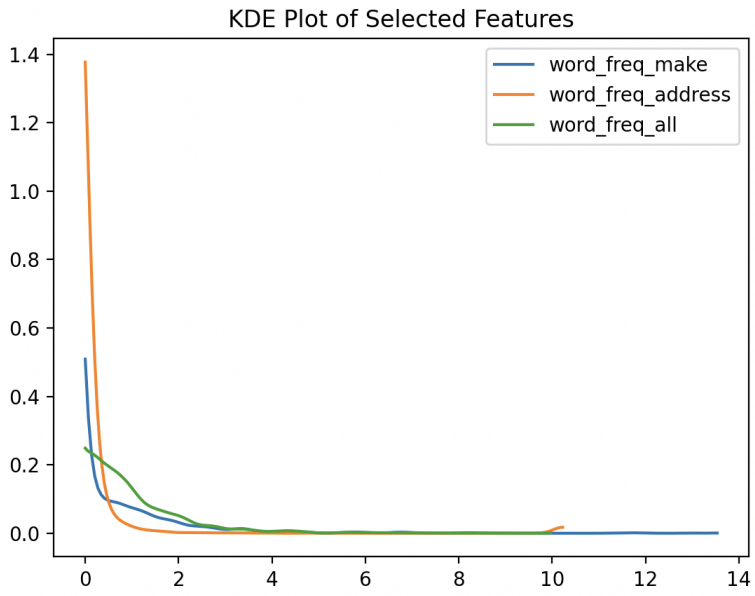

Figure 3: KD Plot of Three Normalized Features

However, when we look at the rest of the graph to the right, we can clearly see that gradient descent was able to reach a better solution with a lower negative log-likelihood.

3 Task 3: Prediction

In this section, we fit two models using the gradient descent and stochastic gradient descent functions that we implemented in the previous question, made predictions on the test set, and compared the performance of each model. We calculated the predicted probabilities as $\sigma(X\mathbf{w})$ and classified labels as 1 if the predicted probabilities were greater than 0.5, and 0 otherwise.

Classification Report: As we can see from Figure 6, both models performed almost the same. The only difference is that the precision scores for classes 0 and 1 are a bit higher with gradient descent compared to stochastic gradient descent. The accuracy and weighted F1 scores for both models are 0.92, which is high.

Feature Importance: We plotted feature weights in a bar chart, as seen in Figure 7. Features with larger absolute weights can be considered highly important. Positive weights indicate that the associated features contribute positively to spam detection, whereas features with negative weights contribute negatively. For instance, "word_freq_3d" has a weight of 8, which is the highest among all weights. Since it is

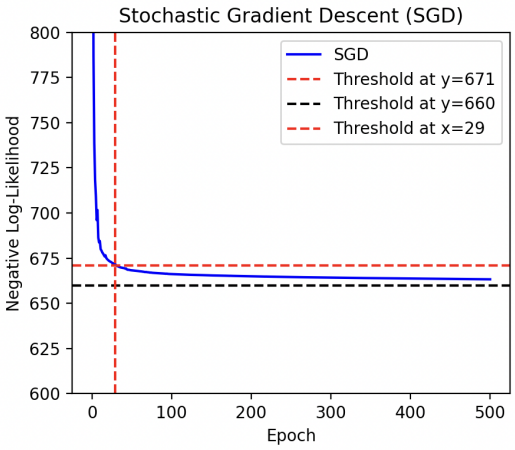

Figure 4: SGD Plot

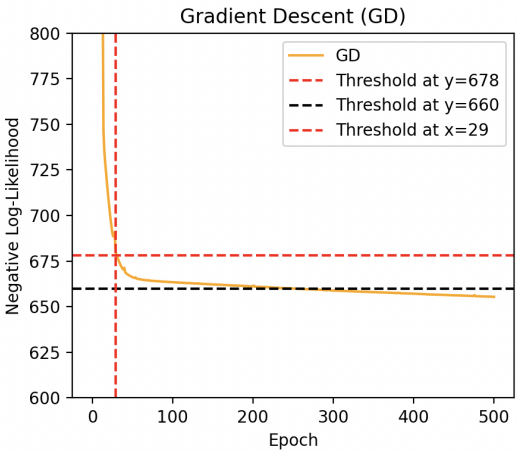

Figure 5: GD Plot

positive, an increase in this feature's value raises the likelihood of predicting spam for that example.

GD:

	precision	recall	f1-score	support
0	0.93	0.94	0.93	941
1	0.91	0.88	0.89	595
accuracy			0.92	1536
macro avg	0.92	0.91	0.91	1536
weighted avg	0.92	0.92	0.92	1536

****

SGD:

	precision	recall	f1-score	support
0	0.92	0.94	0.93	941
1	0.90	0.88	0.89	595
accuracy			0.92	1536
macro avg	0.91	0.91	0.91	1536
weighted avg	0.92	0.92	0.92	1536

Figure 6: Classification Report for GD SGD

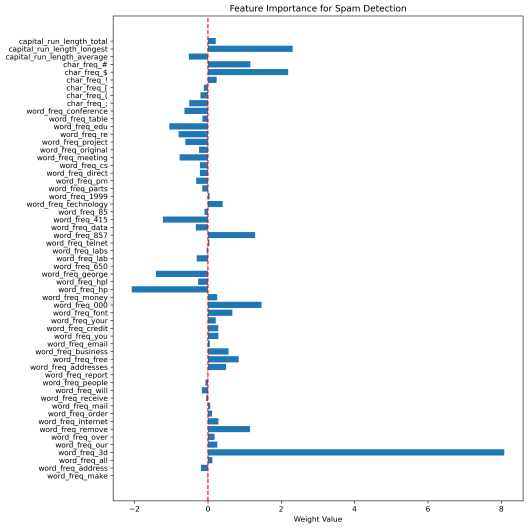

Figure 7: Feature Importance with GD

4 Task 4: Maximum Aposteriori Estimation

4.1 Part b: Effect of the prior on the result by varying the value of λ

Different Values of λ: We trained several logistic regression models with L2 regularization using the following λ values: 0.01, 0.1, 1, 10, 20, and 100. We then used these models to make predictions on the test set. For each λ value, we recorded the training log-likelihood, test accuracy, and F1 scores.

Training Log-Likelihood: As shown in Figure 8, we observe that the training log-likelihood increases as λ increases. This increase implies that the model's fit on the training data worsens with higher regularization. High regularization penalizes the weight magnitudes more aggressively, pushing the model towards simpler solutions, which reduces overfitting but also results in a poorer fit on the training data.

Test Log-Likelihood: The test log-likelihood also increases with increasing λ, but at a slower rate compared to the training log-likelihood, as shown in Figure 9. With high λ, the model's performance deteriorates. We conclude that selecting a high value for λ makes the model too simple, which prevents it from capturing patterns in the test data.

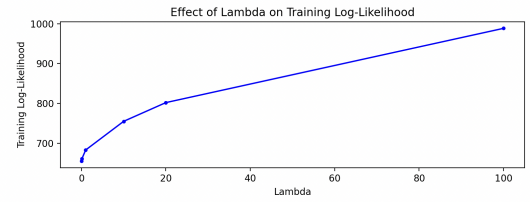

Figure 8: Training Log-likelihood

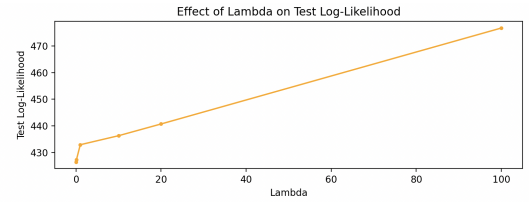

Figure 9: Test Log-likelihood

Test Accuracy and F1 Scores: From Figures 10 and 11, we observe that the model's test performance peaks when $\lambda = 10$, but then worsens as λ continues to increase. This suggests that a moderate amount of regularization helps to improve the model's performance on the test set by reducing overfitting and enabling better generalization. However, excessive regularization deteriorates predictive performance.

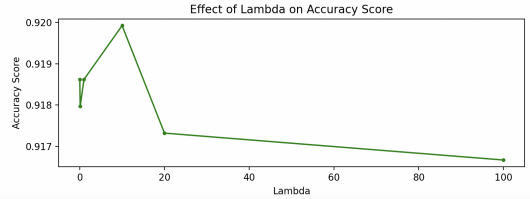

Figure 10: Accuracy score with varying λ values

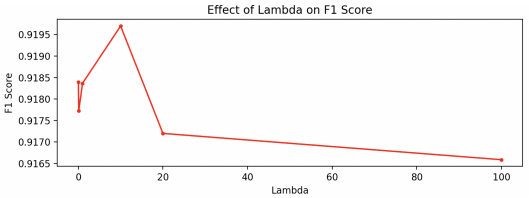

Figure 11: F1 score with varying λ values

4.2 Part c: Composition of the weight vector for varying choices of λ

Large λ and Feature Weigths: We again trained several logistic regression models with L2 regularization using higher λ values such as 10, 100, 300, and 450. We plotted the magnitudes of the weights for each λ. As we increase λ, the weights decrease in magnitude, as can be seen in Figure 12. Features with weights converging near zero may be less important for the classification task, while those remaining non-zero might indicate essential features, especially at high λ values. When $\lambda = 450$, the five features with the highest values are: `'word_freq_remove', 'word_freq_free', 'char_freq$', 'word_freq_000', and 'word_freq_your'`. These features can be considered as the most significant features.

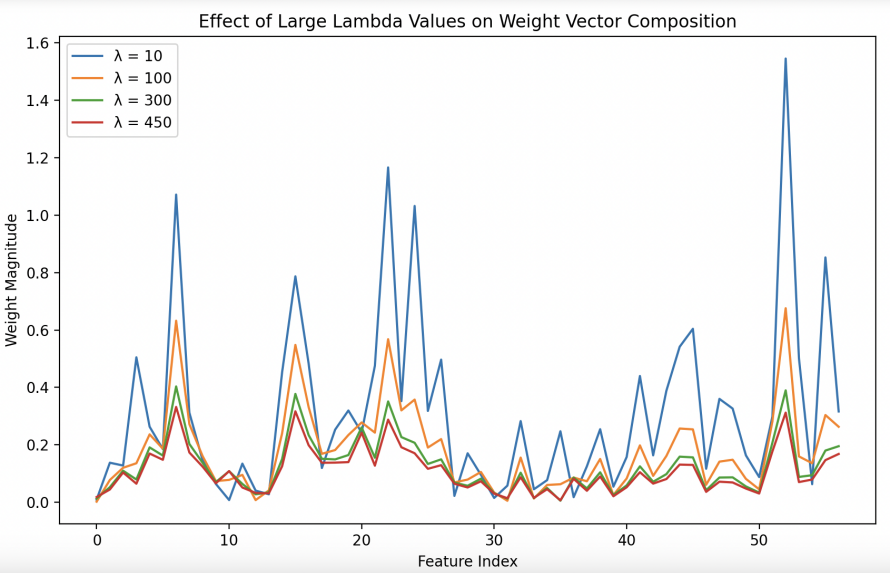

Figure 12: Effect of λ on the Weight Composition

5 Task 5: Exploration (Optional)

5.1 Part a: Try gradient descent on the original data without using z-scores

Overflow Issue: Since the features are on different scales, we encountered an overflow problem. The likelihood values in each epoch are NaN (not a number). Large feature values can cause gradients to explode during training, making it difficult for the model to converge and causing parameters to grow uncontrollably.

5.2 Part b: Add a bias feature

Slight Improvements: We added a bias column consisting of ones to the training and test sets and then fit a model using gradient descent. This time, the minimum log-

likelihood value decreased to 590 from 655, improving the training likelihood. Additionally, the accuracy and weighted F1 scores on the test set increased to 0.93, compared to 0.92 without the bias feature.

5.3 Part c: Try to reduce the training set size and compare MLE and MAP estimation

10-Fold Cross Validation: We reduced the training set size by half and applied 10-fold cross-validation with shuffle set to True. Then, we fit two models: one with standard gradient descent and the other with L2 regularized gradient descent with $\lambda = 0.1$. We calculated the weighted F1 and accuracy scores. For the MAP model, the mean accuracy and F1 score were 0.9445 and 0.9438, respectively. For the MLE model, the mean accuracy and F1 score were 0.9452 and 0.9445. The results are very similar.

5.4 Part d: Run a logistic regression method from some existing library

LogisticRegression from sklearn: We used LogisticRegression from sklearn and obtained nearly the same results as with our own gradient descent and optimization implementation. The accuracy and weighted F1 scores on the test set are both 0.92.

References

[1] Gemulla, R. (2024). *IE 675b: Machine Learning* [Lecture slides]. University of Mannheim.

[2] Murphy, K. P. (2022). *Probabilistic machine learning: An introduction*. The MIT Press. https://probml.github.io/pml-book/book1.html

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	+
ChatGPT	Reinforcing mathematical understanding	Q2 b,c,d, Q4 a	+

Unterschrift
Mannheim, den 3. November 2024