Machine Learning (HWS24)

Assignment 1: Naive Bayes

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

October 13, 2024

1 Task 3: Experiments on MNIST digits data

1.1 Part a: Train the Naive Bayes Model and Evaluate Accuracy

In this section, we trained our Naive Bayes model on the MNIST training dataset with a parameter $\alpha = 2$ for Laplace smoothing. The model was trained to classify handwritten digits (0-9) based on pixel intensity features from 28x28 images.

Training Process:

- The training dataset consisted of 60,000 images of handwritten digits.
- We applied a symmetric Dirichlet prior to account for the categorical nature of the data.
- The model was trained using the `nb_train` function, which computes the class priors and class-conditional probabilities.

Predictions:

- After training, we predicted the labels of the MNIST test dataset, which contains 10,000 images.
- The accuracy of the model was calculated as follows:

$$\text{Accuracy} = 1 - \text{Misclassification Rate} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

Results: The accuracy obtained was approximately **83.63%**. This indicates how well our model performed in classifying the digits in the test set.

1.2 Part b: Plotting Test Digits and Analyzing Misclassifications

To further evaluate the performance of our model, we visualized some test digits along with their predicted class labels.

Test Digits Visualization: Next, we plotted a selection of test digits for each predicted class label. This visualization allowed us to manually inspect the predictions and spot any discrepancies.

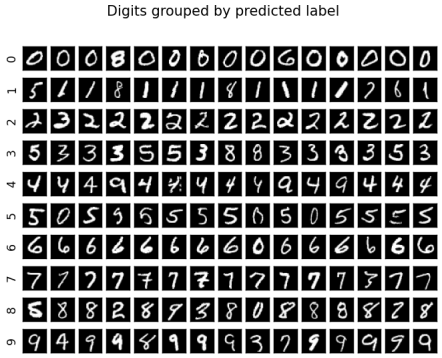

Figure 1: Test Digits with Predicted Class Labels

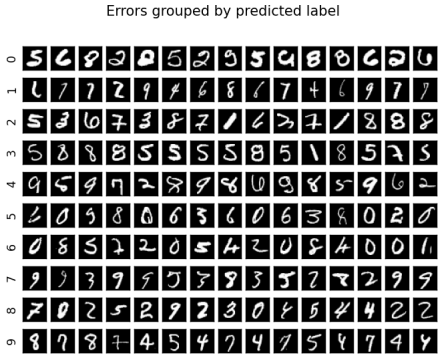

Figure 2: Misclassified Test Digits

Errors Identification: From the visual inspection (see Figure 1), we identified certain misclassifications (for example, in the case of 0, the images of 8 and 6 were misclassified) where the predicted labels differed from the true labels. These instances provide insight into the model's limitations and the digits that are frequently confused.

Misclassified Test Digits Visualization: Following this, we plotted the misclassified digits (see Figure 2) for each predicted class label to analyze the nature of the errors further.

Confusion Matrix: Finally, we computed the confusion matrix (see Figure 3) to evaluate our model's performance more quantitatively. The confusion matrix summarizes the number of correct and incorrect predictions made by the model, allowing us to analyze the specific digits that were often misclassified.

Classification Report: The classification report (see Figure 4) provides a detailed breakdown of the model's performance for each class. The F1 score, which balances precision and recall, is especially valuable for evaluating models in cases of imbalanced datasets.

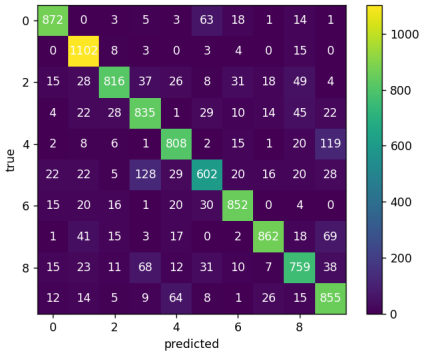

Figure 3: Confusion Matrix of Model Predictions

	precision	recall	f1-score	support
0	0.91	0.89	0.90	980
1	0.86	0.97	0.91	1135
2	0.89	0.79	0.84	1032
3	0.77	0.83	0.80	1010
4	0.82	0.82	0.82	982
5	0.78	0.67	0.72	892
6	0.88	0.89	0.89	958
7	0.91	0.84	0.87	1028
8	0.79	0.78	0.79	974
9	0.75	0.85	0.80	1009
accuracy			0.84	10000
macro avg	0.84	0.83	0.83	10000
weighted avg	0.84	0.84	0.84	10000

Figure 4: Classification Report

Discussion on Errors: The confusion matrix revealed specific patterns in misclassifications. For instance:

- The digit '5' was frequently misclassified as '3'. A total of 128 different images were predicted as '3', but in fact, they were '5'.
- The digit '4' was often mistaken for '9'. In 119 instances, images were incorrectly labeled as '4' when they were actually '9'.
- In the classification report, we can clearly conclude that the model performs best in predicting classes 0, 1, and 6, as indicated by the F1 score.

These patterns suggest that the model struggles with distinguishing between certain shapes or features of similar digits, possibly due to variations in handwriting styles.

2 Task 4: Model Selection

In this section, we implemented a 5-fold cross-validation code to select the alpha value with the maximum accuracy and F1 scores. Since the number of classes in the data is not equal, we used the weighted F1 score, which is the mean F1 score calculated by accounting for the number of true instances in each class. The alpha values that we observed are 1, 2, 3, 4, and 5. Since we have 5 folds and 5 different alpha values, there were a total of 25 fits. For each alpha value, the model is trained 5 times using 4 different folds each time, and predictions are made on the unused fold. Lastly, we calculated the cross-validation (CV) results for each alpha and metric by taking the mean.

Discussion on CV Results:

- As we can see in Figure 5, we obtained the highest weighted F1 and accuracy scores at alpha = 2, with values of 0.8267 and 0.8274, respectively.

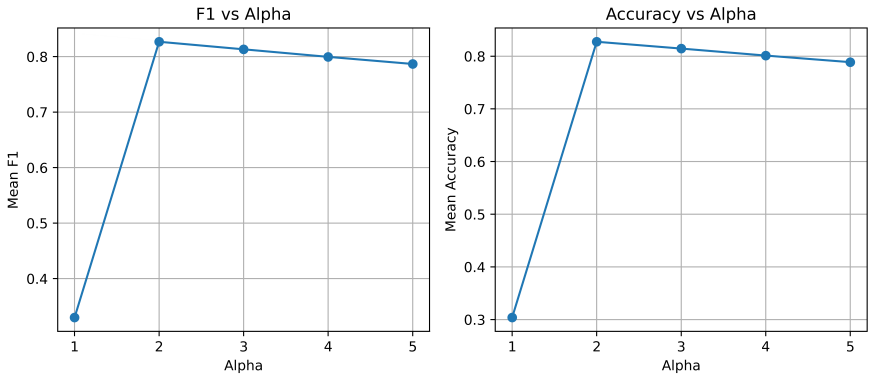

Figure 5: Cross Validation Results

- When alpha is 1, it corresponds to maximum likelihood estimation, and overfitting is present. The model cannot generalize to unseen data. When alpha is 2, it indicates that we are using the MAP estimate with a Dirichlet prior, which corresponds to add-one smoothing. In this case, overfitting decreases, and prediction performance on unseen data increases. However, if we keep increasing alpha, considering the bias-variance tradeoff, our bias also increases, leading to a model that underfits. The model's reliance on the prior increases as we continue to raise alpha. Therefore, we obtain lower F1 and accuracy scores in cross-validation.

3 Task 5: Generating Data

3.1 Part b: Generate Some Digits of Each Class

Getting the Probabilities by Exponentiation: We first converted the log probabilities provided by the model to probabilities:

$$P(X_j = k \mid Y = c) = \frac{e^{\log P(X_j=k'|Y=c)}}{\sum_{k'=0}^{K-1} e^{\log P(X_j=k'|Y=c)}}$$

Random Sampling From Categorical Distribution For each row i and feature j, we randomly sample a value k according to the probabilities that we calculated in the previous step:

$$X_{\text{gen}}[i,j] \sim \text{Categorical}(P(X_j = k \mid Y = c), k = 0, 1, \ldots, K-1)$$

Generating Digit Images We generated two different digit images using two different models, with alpha values equal to 2 and 5.

Discussion on Images: Alpha values affect the images generated by the models.

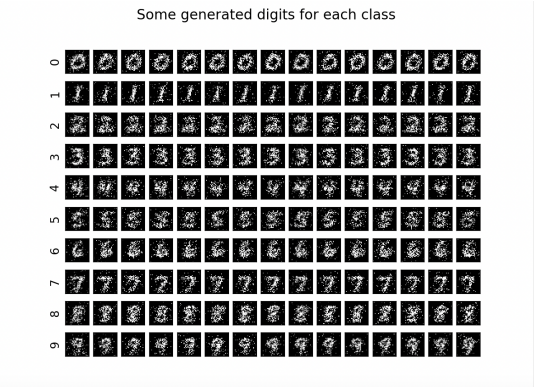

Figure 6: Digits Generated by the Model with alpha=2

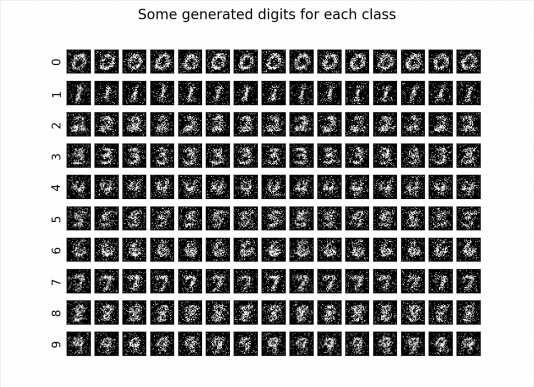

Figure 7: Digits Generated by the Model with alpha=5

- As we increase alpha, we reduce overfitting but increase bias. The model places more emphasis on the prior as alpha continues to rise, which prevents it from detecting distinctive patterns in the data. As a result, the images become noisier, and the digits are less defined, as shown in Figure 6 compared to Figure 5.

4 Task 6: Missing Data

4.1 Part a: $p(y \mid x_{1:D})$

This is the model we trained in Q1 using all 784 features. By applying Bayes' Theorem and the assumption of independence between features in Naive Bayes, we arrive at the following formula:

$$p(y \mid x_{1:D}) = \frac{p(y) \cdot \prod_{j=1}^{D} p(x_j \mid y)}{p(x_{1:D})} \propto p(y) \cdot \prod_{j=1}^{D} p(x_j \mid y)$$

We can ignore the probability in the denominator, as it is a constant. This is useful when all the features are available in the dataset.

4.2 Part b: $p(y \mid x_{1:D'})$ for $1 \leq D' < D$

Again, we can exploit the assumption of independence between features in Naive Bayes and use Bayes' Theorem to derive the following formula:

$$p(y \mid x_{1:D'}) = \frac{p(y) \cdot \prod_{j=1}^{D'} p(x_j \mid y)}{p(x_{1:D'})} \propto p(y) \cdot \prod_{j=1}^{D'} p(x_j \mid y)$$

This time, instead of using all the features, we use a subset of them. If some features are unavailable or have missing values, this approach can be useful. Additionally, we can observe how prediction performance changes when not all features are used. This could help us identify the importance of each feature and lead to better feature selection.

4.3 Part c: $p(x_{D'+1:D} \mid x_{1:D'})$ for $1 \leq D' < D$

In order to derive the conditional distribution for the given D discrete features, each taking values in $\{0, \ldots, K-1\}$, and C classes, we can use sum rule, Bayes Theorem and the assumption of independence between features in Naive Bayes. The formula can be expressed as:

$$= \frac{p(x_{D'+1:D}, x_{1:D'})}{p(x_{1:D'})}$$

$$= \frac{\sum_{y=0}^{C-1} p(x_{D'+1:D}, x_{1:D'}, y)}{p(x_{1:D'})}$$

$$= \frac{\sum_{y=0}^{C-1} p(x_{D'+1:D}, x_{1:D'} \mid y) \cdot p(y)}{p(x_{1:D'})}$$

$$= \frac{\sum_{y=0}^{C-1} p(x_{D'+1:D} \mid y) \cdot p(x_{1:D'} \mid y) \cdot p(y)}{p(x_{1:D'})}$$

$$= \sum_{y=0}^{C-1} \frac{p(y) \cdot p(x_{1:D'} \mid y)}{p(x_{1:D'})} \cdot \prod_{j=D'+1}^{D} p(x_j \mid y)$$

Using Bayes' Theorem, $p(y \mid x_{1:D'}) = \frac{p(y) \cdot p(x_{1:D'} \mid y)}{p(x_{1:D'})}$

$$= \sum_{y=0}^{C-1} p(y \mid x_{1:D'}) \cdot \prod_{j=D'+1}^{D} p(x_j \mid y)$$

This distribution can be particularly useful in **imputation tasks**, where we want to predict missing features based on the available features and the classification. By utilizing the available features, we can estimate what the missing features are likely to be, improving the robustness of the model and enhancing prediction accuracy.

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	Q5,6	+

Unterschrift
Mannheim, den 13. October 2024