CS/EE 5841

Classifying MNIST dataset with Naive Bayes and Logistic Regression

Date: 03/20/2020

Prateek Goyal

Data Science, Michigan Technological University

Submitted to Dr. Timothy Heavens in partial fulfillment for the course CS/EE 5841 Machine Learning

Abstract- The report aids in understanding the Naïve Bayes and the Logistic Regression Classifiers and their application to classify the MNIST dataset which contains images of handwritten digits. The two classifiers are compared with each other on the basis of the performance and properties. The improvements and future work is also explained.

Index Terms- Naïve Bayes, Logistic Regression, Ridge Regularization, log-likelihood, gradient descent

I. MNIST Database

The The MNIST database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image. The original black and white (bilevel) images from NIST were size normalized to fit in a 20x20 pixel box while preserving their aspect ratio. The resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. The images were centered in a 28x28 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 28x28 field.

II. Naïve Bayes Classifier

The naïve bayes classifier is based on the concept of Bayes' theorem. We assume that the features are independent of each other i.e one feature does not affect any other feature and hence the concept of being naïve is introduced. To classify a given data point, the naïve bayes classifier finds the class with the maximum probability given the data points features.

III. Logistic Regression

It's a classification algorithm that is used where the response variable is categorical. The idea of Logistic Regression is to find a relationship between features and probability of particular outcome. We model several classes of events such as determining which digit between 0-9 is represented by the 784 dimension vector. Each class is assigned a probability between 0 and 1 and sum of the probability of all the classes' equals 1.

IV. Ridge Regularization

Ridge regularization is a linear regression model also called as (L2 regularization). Ridge regularization will inhibit the model from over-fitting and thereby reduce the model complexity. We perform ridge regression by introducing a penalty term which helps to reduce over-fitting or high bias in the data. The penalty term that we introduce to the loss function is the square of the coefficients (β). We also have the scaling term (α) that takes a value between 0 and 1 and decides the severity of the regularization.

V. Log-Likelihood

The value generated by the loss function can be thought of as a mean of assessing how well our algorithms models the given dataset. If the value of the loss function is higher that means that the prediction our model makes are not accurate, similarly, if the value of our loss function is a lower number then it means that the predictions are good.

VI. Gradient-Descent

Gradient descent is an optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient. The objective of gradient descent is to find the optimal parameters that result in optimizing a given function. In logistic regression we find the optimal parameters by minimizing the loss function.

VII. One-VS-Rest

In One-VS-Rest (OVR) we train N distinct binary classifiers where N is the number of distinct classes, which in our problem statement is 10 i.e (0-9). It assumes that each classification problem is independent. We take values of one class and turn them into positive examples, and the rest of classes - into negatives, we repeat this for the rest of the classes. We then choose the class with the maximum conditional probability.

1. Naïve Bayes Classification on MNIST Dataset

1.1 Model optimization using Hyper-parameter

In statistics, Smoothing is a technique to smooth categorical data. Smoothing is introduced to solve the problem of zero probability. We tested various values for smoothing which is tunable parameter in the Naïve Bayes Classification Model.

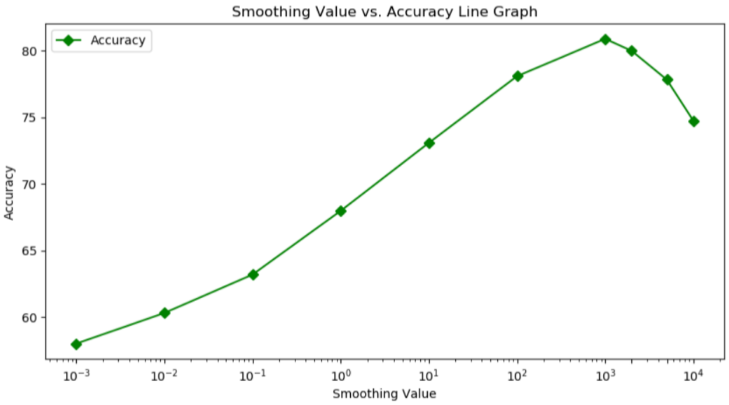


Figure 1: Smoothing values VS Accuracy

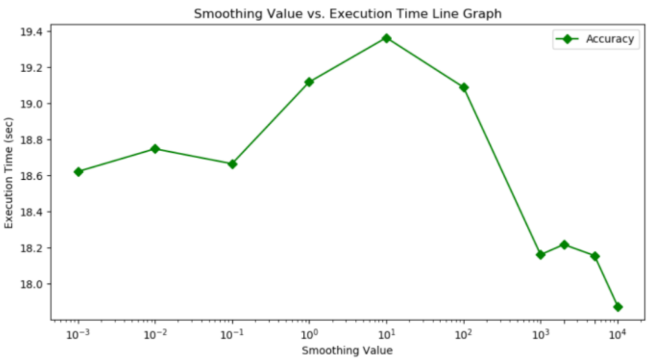


Figure 2 : Value of smoothing parameter and the run time with corresponding value

Naive Bayes classification model is an extremely fast and robust when it high-dimensional dataset is considered. The impact of the various values of smoothing parameter on the run time help us understand that although there is a small depreciation in the run time as the smoothing value appreciates, the difference is negligible. We can thus infer that Naïve Bayes classification models are fast and can be instrumental in finding a baseline for classification problems.

1.2 Confusion Matrix

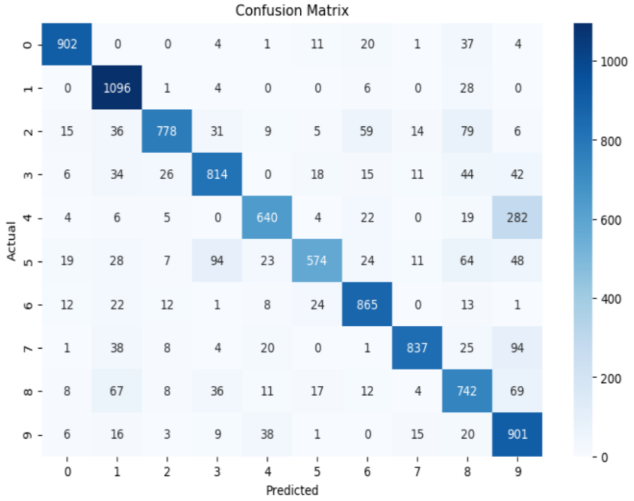
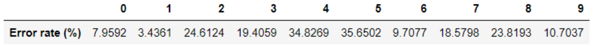


Figure 3: Confusion Matrix

Looking at the “Confusion Matrix” above, we can infer that (9,4),(1,8), (3,5), (2,6), (8,5), (2,8), (9,7) and few more are some tricky combinations of the digits for which the classifier has trouble identifying the right label.

1.3 Error Matrix



	0	1	2	3	4	5	6	7	8	9
Error rate (%)	7.9592	3.4361	24.6124	19.4059	34.8269	35.6502	9.7077	18.5798	23.8193	10.7037

Figure 4: Test Accuracy for Each Digit

- Naïve Bayes seems to perform fairly OK on MNIST data, we were able to get ~81.5% accurate prediction of the class.
- According to the above stats and confusion matrix, seems Naïve Bayes face few challenges in differentiating class digit: 2, 4, 5 and 8

1.4 Conditional Probability Images

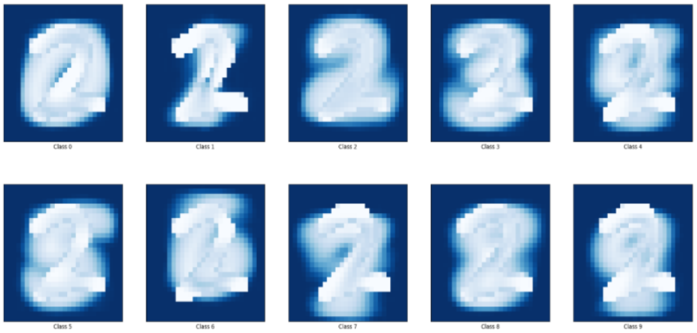


Figure 5: Displaying image for each class given conditional probability

We have displayed above the image for each of the 10 classes, which show an image of the 784 conditional probabilities as a 28 x 28 image. As we can see in the 10 subplots above, the number in the background is the class name and the test data is displayed on top to get the probability. The Naïve Bayes Classifier finds the probability of the match between the class background image and the test data image on top.

2. Logistic Regression on MNIST Dataset

2.1 Derivation

$$\mathrm{L(w)} = \sum_1^N log(1 + exp(-y_i w^T x_i)) + \frac{\lambda}{2}||w||_2^2$$

$$\frac{d}{dw} L(w) = \frac{d}{dw} \sum_1^N log(1 + exp(-y_i w^T x_i)) + \frac{\lambda}{2}||w||_2^2$$

$$\frac{d}{dw} L(w) = \sum_1^N \frac{1}{(1+exp(-y_i w^T x_i))} * \frac{d}{dw}(1 + exp(-y_i w^T x_i)) + \lambda w$$

$$\frac{d}{dw} L(w) = \sum_1^N \frac{exp(-y_i w^T x_i)}{(1+exp(-y_i w^T x_i))} * \frac{d}{dw}(-y_i w^T x_i) + \lambda w$$

$$\frac{d}{dw} L(w) = \sum_1^N \frac{-y_i x_i * exp(-y_i w^T x_i)}{(1+exp(-y_i w^T x_i))} + \lambda w$$

The l2 regularized logistic regression uses the log-likelihood given in the first equation, the regularization parameter (lambda) is added in the equation. We take the derivate of the loss function which is added to the update equation for our weights given below.

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla \mathcal{L}(\mathbf{w})$$

2.2 Test Accuracy VS Regularization Value

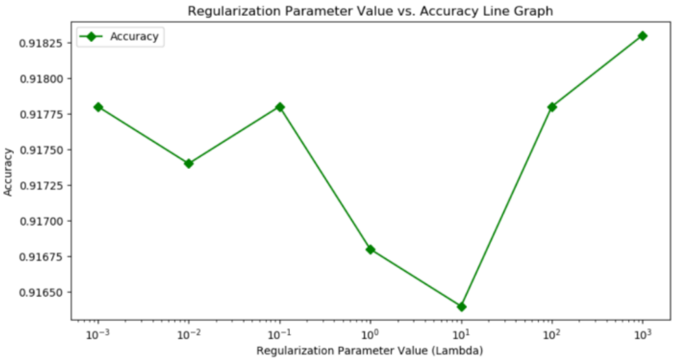


Figure 6: Regularization parameter (λ) VS Accuracy

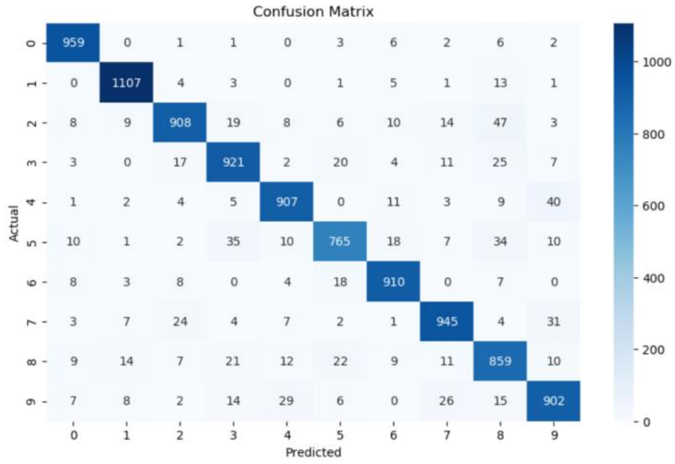
Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error.

- As per the above graph, with lambda = 1000, we got ~92% accuracy, hence avoiding over-fitting.

- Adding the L^2 term usually results in much smaller weights across the entire model. Few sample values mentioned below. [2.01082784e-02, 2.43528738e-02, 2.26555276e-02, 1.36941683e-02,..]

- As we increase the regularization value, penalty value increased for the loss function.

2.3 CONFUSION MATRIX

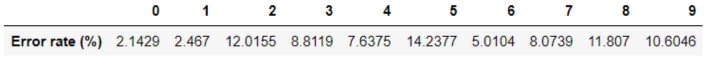


	0	1	2	3	4	5	6	7	8	9
0	959	0	1	1	0	3	6	2	6	2
1	0	1107	4	3	0	1	5	1	13	1
2	8	9	908	19	8	6	10	14	47	3
3	3	0	17	921	2	20	4	11	25	7
4	1	2	4	5	907	0	11	3	9	40
5	10	1	2	35	10	765	18	7	34	10
6	8	3	8	0	4	18	910	0	7	0
7	3	7	24	4	7	2	1	945	4	31
8	9	14	7	21	12	22	9	11	859	10
9	7	8	2	14	29	6	0	26	15	902

Figure 7: Confusion Matrix

Looking at the "Confusion Matrix" above, we can infer that (4,9), (3,5), (2,8), (5,8), (9,4) and few more are some tricky combinations of the digits for which the classifier has trouble identifying the right label.

2.4 Error Matrix



	0	1	2	3	4	5	6	7	8	9
Error rate (%)	2.1429	2.467	12.0155	8.8119	7.6375	14.2377	5.0104	8.0739	11.807	10.6046

Figure 8: Error Matrix for Each Digit

- Compared to the Naïve Bayes, Logistic Regression seems to performs good on MNIST data, we were able to get ~88% accurate prediction of the class.

- According to the above stats and confusion matrix, seems Logistic Regression face few challenges in differentiating class digit: 2, 5, 8 and 9.

2.5 Image Results

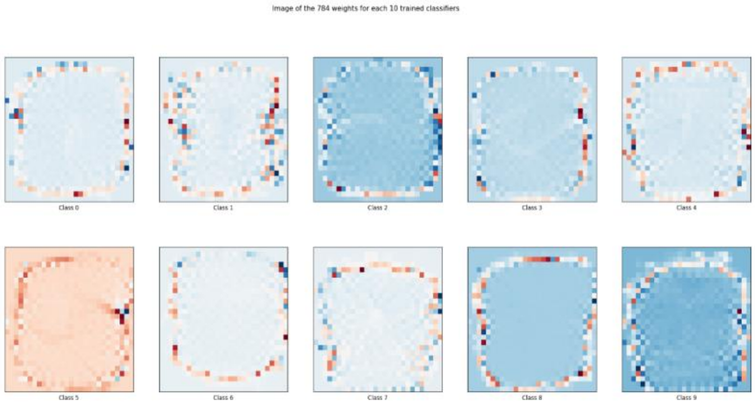


Figure 9 : Digit Classified by Logistic Regression

The above images are unclear to distinguish to the naked eye, as we are plotting the weight of the each feature (pixel) instead of pixel color density value. The weight value represents the importance of a pixel in determining the class. The value of the weight is more for the pixels that appear in the digit, and less for pixels that appear in the background. Hence, weight plays an important role in classifying an image

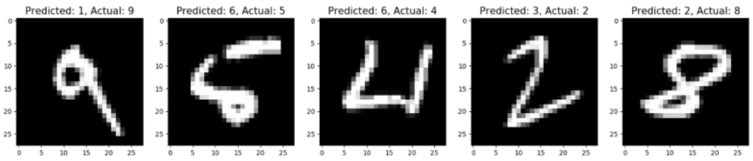


Figure 10 : Misclassified Digits

The above figure shows us a few of the digits that were incorrectly classified. The actual class and the predicted incorrect class are mentioned in the figure above each digit.

3. Comparison Between Naïve Bayes and Logistic Regression

3.1 Algorithm's Learning mechanism

The learning mechanism is a bit different between the two models, Naive Bayes is a generative model and Logistic regression is a discriminative model.

Generative model: The posterior probability is predicted using the joint distribution of feature X and target Y.

Discriminative model: The posterior probability is modeled directly by learning the input to output mapping by minimizing the error.

3.2 Classification Algorithms

Both the algorithms are used to carry out tasks pertaining to classification. The algorithms figure out if a given data point belongs to a certain class or not.

3.3 Approach to be followed to improve model results

Naïve Bayes: When the training data size is small relative to the number of features, the information/data on prior probabilities help in improving the results

Logistic regression: When the training data size is small relative to the number of features, including regularization such as Lasso and Ridge regression can help reduce over-fitting and result in a more generalized model.

4. Conclusion

4.1 Performance Matrix

Logistic Regression outperforms Naïve Bayes with achieving 7% more accuracy in the testing dataset. Logistic Regression is able to distinguish more classes with good accuracy compared to Naïve Bayes

4.2 Time Complexity

Naïve Bayes runs faster than the Logistic Regression for each run. Naïve Bayes took ~1.7 seconds whereas Logistic Regression classifier took more than 2.5 seconds

5. Areas of Improvement

- In case of Naïve Bayes, we can optimize our classifier by tuning its hyper-parameter, in this case, smoothing parameter. We have tried for various smoothing parameter and found that smoothing=1000 value to be the best one among all.

- In case of Logistic Regression (with Gradient Descent and L2 Regularization), we can optimize our classifier by tuning following parameters:
  * Learning Rate (Gradient Descent)
  *Max Iterations (Gradient Descent)
  *lambda (regularization)
  *Solver (Logistic Regression)

- CNN using Deep Learning: We can classify the images using neural network with more than 98% accuracy by implementing various techniques like Feature Mapping, RELU Layer, Max pooling, Flattening, Full connection etc.

6. Acknowledgment

We would like to express our gratitude to Dr. Timothy Havens, who helped us along the project with his insightful notes and lectures. We would also like to thank the TAs for their guidance in moments of difficulty.

7. References

- https://www.kaggle.com/jeppbautista/logistic-regression-from-scratch-python

- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

- https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc

- https://towardsdatascience.com/introduction-to-na%C3%AFve-bayes-classifier-fa59e3e24aaf

- Prof. Timothy Haven's Lecture Notes.