CS 440 Machine Problem 4 Report

Deep Patel

November 28, 2018

1 Naive Bayes Classifier

1.1 Joint Probability Representation

The drawback to using the multiplicative representation of joint probability is that the multiplication of too many matrices often results in underflow, meaning the product becomes too small to be represented accurately as a floating point value. This is because in the multiplication, there are naturally many probabilities that are very small, so they should bring the product down to a number that may be lower than the epsilon that is the smallest number the machine can represent.

Another drawback of taking the product is that it can become very expensive in terms of computation. This is why taking the log of both sides of the expression helps to avoid this underflow, as adding the log of a small probability will not cause this. Because we are not looking for the probability product itself, but the argmax of the label, y, that maximizes this probability, and taking the log will not change these results, it is suitable to use the logarithm to make the computation cheaper and to avoid underflow.

1.2 Laplace Smoothing Parameter

$$P(F_i = f_i | Y = y) = \frac{c(f_i, y) + k}{\sum_{f_i' \in \{0,1\}} (c(f_i', y) + k)}$$

Figure 1: Laplace Smoothing Equation

Through trial and error, I found that keeping my k value low resulted in higher accuracy against the validation data. because the training data is relatively small, adding a whole observation per feature per pixel location smooths out the distribution in an undesirable way, so as to make it hard for the classifier to distinguish between similar digits. Keeping k low, makes the discrepancies which are encoded through these features probabilities more clear cut. I believe this is because the role of k should just be to avoid singularities when taking the log (avoiding log(0) situations). Increasing k brings the distribution closer to a uniform distribution, which is what tending k towards infinity does exactly.

Looking more deeply, when the probability of a feature given a label is exactly 0, smoothing this probability using the equation in Figure 1 increases it from 0 by a small amount. The smaller this amount is, the more negative the log of the probability will be. Thus, by if this feature is feature of an image we are testing, it will receive a very negative log probability because it is very unlikely that this label will be it because its probability of having this feature was 0, which is the desired behavior we would like. Thus, I set my k parameter to 0.00001.

1.3 Classifier Results using Basic Feature Set

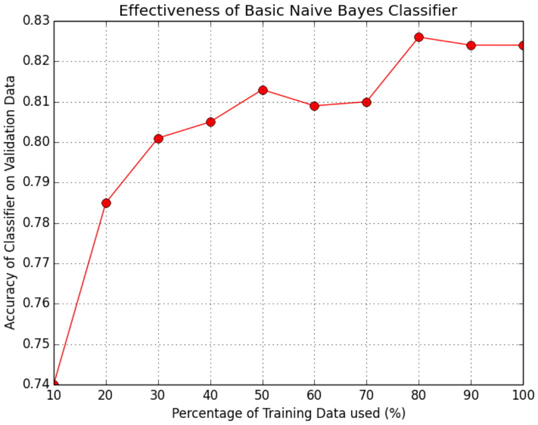

Figure 2: Basic Naive Bayes Training Data Percentage vs. Accuracy Results

Table 1. Results for Naive Bayes with Basic Features

Percentage of Training Data Used	***Accuracy on Validity Data***
10%	0.740
20%	0.785
30%	0.801
40%	0.805
50%	0.813
60%	0.809
70%	0.810
80%	0.826
90%	0.824
100%	0.824

Training the classifier with 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and, 100%, we get the results shown below in Table 1 and Figure 2. As we can see, there is a general improvement with an increase in the training data used. It is interesting, for example, when increasing the training data used from 50% to 60%, we see a slight decrease in accuracy. After trying to tweak the Laplace smoothing parameter, k, I found that it did not increase the accuracy and concluded that this may be due to overfitting of the data or, for instance, that percentage of data from 50-60% influencing the model negatively for this specific validation set. Nevertheless, the overall trend assures us that the model improves in performance with more training examples.

2 Enhancing Features

2.1 Feature 1: Number of Non-Zero Pixels in Diagonals

This feature takes the sum of the number of non-zero pixels in each diagonal direction and for each pixel, takes the takes the sum of the two diagonal sums for the two diagonals that that pixel belongs to. The motivation behind this came from paper by Patel referenced here [2]. This feature brought about an accuracy of 0.685 when used alone on all of the training data.

2.2 Feature 2: Horizontal-Celled Projection

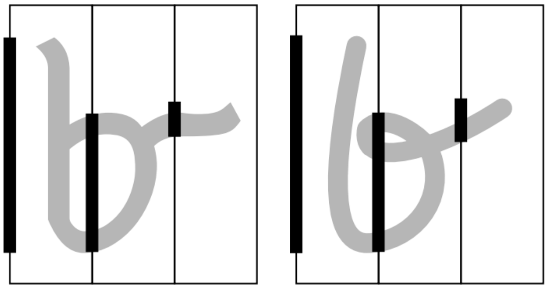

Figure 3: An example of the celled projection. It is noticeable that although the digits are written in different styles, their cell projections are quite similar

This method is described in Hossain's paper, referenced here: [1]. The main concept is to partition the grid into k regions and take the projection of each region. Please refer to the paper for more detail and the algorithm used. This algorithm significantly increased the accuracy of the data from 0.685 with just feature 1 to 0.791 with both feature 1 and 2.

2.3 Feature 3: Crossings

The last feature used is the number of horizontal and vertical crossings that occur for each row and column, respectively. A crossing is defined as an occurrence of a change in the sign of a pixel along a linear sweep. Thus, if we count the number of crossings in each row and column, this feature is can hold rich information about an image, making it a good feature for this case. Adding this feature gave a generous improvement to the accuracy of the data.

2.4 Advanced Feature Set Results

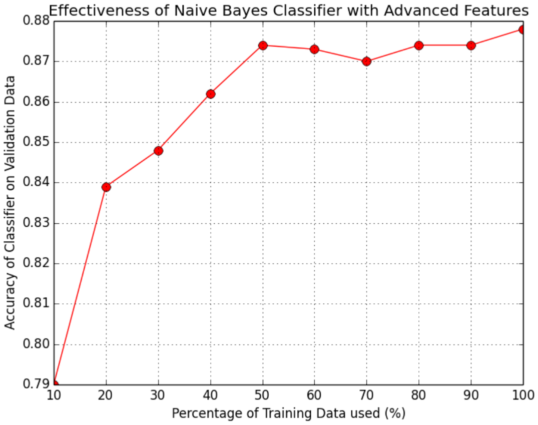

Figure 4: Naive Bayes with Advanced Feature Set Training Data Percentage vs. Accuracy Results

Table 2. Results for Naive Bayes with Advanced Feature Set

Percentage of Training Data Used	***Accuracy on Validity Data***
10%	0.790
20%	0.839
30%	0.848
40%	0.862
50%	0.874
60%	0.873
70%	0.870
80%	0.874
90%	0.874
100%	0.878

2.5 Basic & Advanced Feature Sets Combined Results

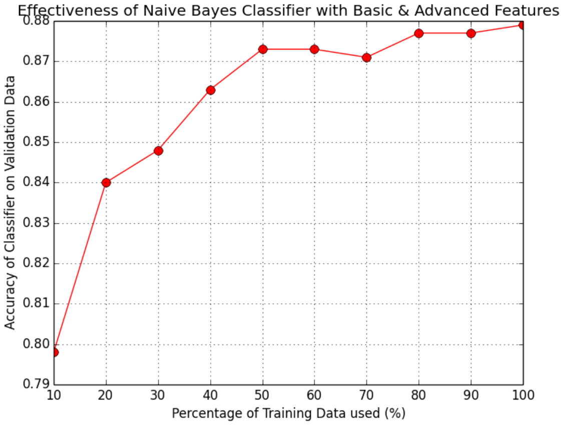

Figure 5: Naive Bayes with Basic & Advanced Feature Sets Training Data Percentage vs. Accuracy Results

Table 3. Results for Naive Bayes with Basic & Advanced Feature Sets

Percentage of Training Data Used	***Accuracy on Validity Data***
10%	0.798
20%	0.840
30%	0.848
40%	0.863
50%	0.873
60%	0.873
70%	0.871
80%	0.877
90%	0.877
100%	0.879

It seems that combining the basic and advanced feature sets generally results in the highest accuracy on the classification task for any amounts of training data used. Still, without the advanced feature set, results with the basic feature set is not able to surpass a threshold of about 0.83 accuracy with all the training data used. With the advanced feature set used alone, the classifier is able to get above 0.87 accuracy in the same situation. Combining the basic feature with the advanced feature set only added a small improvement in accuracy ranging from 0.01 to 0.1.

2.6 Final Feature Set

Based on the performance I have seen, for final features, I decided to use the advanced feature set along with the basic feature set with a slight tweak. Instead of the basic feature set being a binary feature set, I let it have 3 values: 0, 1, or 2, which correspond to the pixel value for each pixel. This seemed to give a measurable gain in accuracy with barely any added computational effort.

References

[1] M. Zahid Hossain, M. Ashraful Amin, and Hong Yan. Rapid feature extraction for optical character recognition. *CoRR*, abs/1206.0238, 2012.

[2] Ishani Patel. A survey on feature extraction methods for handwritten digits recognition. *International Journal of Computer Applications (0975 – 8887)*, 107(12):846–894, 2014.