Density Estimation & Classification

Abhijit Chakraborty
Email: achakr40@asu.edu

***Abstract*—This is a study on the density estimation and classification done on dataset from MNIST dataset to help identify and classify the content into the digit 0 to 9. Details of the application modules, the approach to the solution and quick snapshot of the results are provided.**

***Keywords*—Naïve Bayes, classifiers, features, assumptions, confusion matrix**

I. Introduction

Naive Bayes is a statistical technique for building classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels belong to some finite set.

In this project, study was performed on Naïve Bayes Classifier analysis on a given sample image of data, against the MNIST data set images to classify them into two digit 0 to 9. The features identified for the classification of the data samples are the average brightness of each image and the standard deviation of the brightness of each image. The study was performed individually and no team effort was involved.

The project consists of 4 modules:

A. MNIST data set

The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems.[1][2] The database is also widely used for training and testing in the field of machine learning.[3][4] It was created by "re-mixing" the samples from NIST's original datasets.[5] The creators felt that since NIST's training dataset was taken from American Census Bureau employees, while the testing dataset was taken from American high school students, it was not well-suited for machine learning experiments.[6] Furthermore, the black and white images from NIST were normalized to fit into a 28x28 pixel bounding box and anti-aliased, which introduced grayscale levels.[6].The MNIST database contains 60,000 training images and 10,000 testing images.[7]

B. Sample data set

This consists of four sample datasets with two for each digit and categorized as training and testing samples. In this study there were training sets for each along with separate testing set for each digits were used.

C. Features

Classifier in machine learning is used to classify different objects based on certain features. In this project there were two features identified to discriminate the datasets into digit 0 to 9.These are:

- The average pixel brightness of each image
- The standard deviation of all the pixel brightness of each image

D. Code

A coding algorithm has been built to implement the classifier on the training and testing set. The detail of each is being mentioned in detail in the following sections of this document. Each steps involved experience programming knowledge and good understanding of the MNIST data classes.

II. Description of solution

The Naïve Bayes classification for density estimation and classification is one of the simplest of classification algorithms. To understand Naïve Bayes fully, we need to look at some basic probabilistic concepts-

A. Basic concept of Classifiers

Classification Algorithms form a crucial pillar of Machine Learning. The task of a classification model is simple- based on all the training samples provided to the model, determine the class a sample belongs to. The applications of classification models is vast & far-reaching. Digit & handwriting recognition on our phones use classification algorithms & Google push targeted ads to us using search patterns of thousands of similar users. Computer vision which is a scientific field in AI where computers analyzes images or videos is also an implementation of classification algorithms.

B. Conditional Probability

Conditional Probability gives the probability of a second event occurring given that we know the outcome of the first event.

$$P(B|A)= \frac{P(A \text{ and } B)}{P(B)}$$

C. Bayes Theorem

Conditional Probability can be further expanded by Bayes' Theorem. It is expressed as-

$$P(B|A)=\frac{P(A|B)P(A)}{P(B)}$$

Basically, it expresses the conditional probability of a second event B given an event A, when we know the probability of A given B.

D. Independence

An event A is independent from event B when the probability of event A is not influenced by the probability of event B. It can be expressed as-

$$P(A|B)=P(A)$$

When A and B are independent of each other, then they fulfill the condition: P (A and B) =P(A) P(B)

E. Classification

To classify the images using Naïve Bayes, we need to find P (class I image).

$$P\text{ (class | image)}=\frac{P(\text{image | class})*P(\text{class})}{P(\text{image})}$$

F. Main Logic

Now P (image) is equal for all image and can be ignored as this does not affect the outcome. Now to determine which class an image belongs to, we will find the probability of all classes (0 to 9) and assign the image to the class that gives us the highest probability.

$$\text{Assigned class, C= max}_c\text{ P(class | image) = max}_c\text{ (P(image | class)*P(class))}$$

P (class) can be found out by:

$$P\text{(class)}=\frac{\text{count(Images of the class)}}{\text{count(All images)}}$$

Now to calculate the P(image|class) a multivariate normal distribution function has been used.
Finally, in Naïve Bayes we make a naïve assumption that each pixel in an image is independent of the other image.
According to the independence condition
P (A, B) = P (A)P(B).
The **Gaussian Normal Distribution** can be represented by:

$$P\text{(image | class)} = \prod_{i=1}^{784} \frac{1}{\sqrt{2\pi\sigma_i{}^2}} \exp\left(-\frac{(x_i-\mu_i)}{2*\sigma_i{}^2}\right)$$

Where:

$$\mu_i = Mean\ values\ of\ each\ i^{th}\ image\ pixel\ of\ a\ class$$
$$\sigma_i{}^2 = Variance\ of\ each\ i^{th}\ image\ pixel\ of\ a\ class$$

Thus, the features are assumed to be independent. contribution to the outcome.

G. Assumptions

The fundamental assumption is that each feature makes an:

- Independent
- Equal.

With relation to our dataset, this concept can be understood as:

(a) Each feature is contributing equally to the outcome. For example, knowing only average of brightness and standard deviation alone can’t predict the outcome accurately. None of the attributes is irrelevant.

(b) It is assumed that none of the pair of features are dependent. For example, the average of brightness of each image and the standard deviation of brightness of each image are complete two independent features. Though mathematically mean and standard deviation has a relationship.

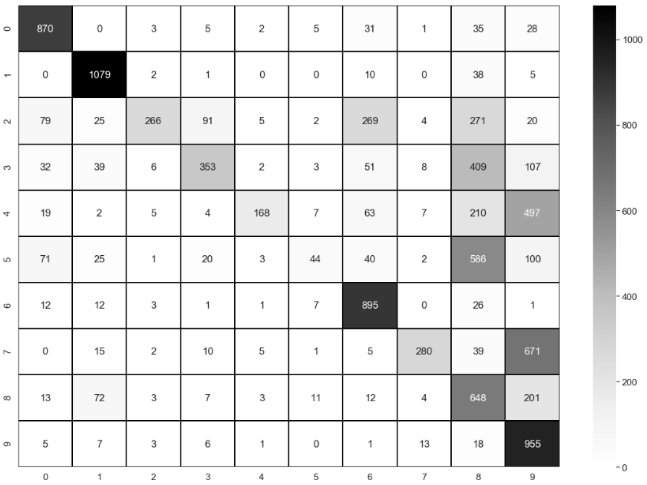
III. Results

After performing thorough analysis and review of the results, the process has been explained in the description and prior modules, the results are formatted in the tabular format.

A. Observations

Naive Bayes Classifier does not appear to perform well for MNIST data set as it produced an overall accuracy of 56%.

Looking at confusion matrix below, we can observe that (5,8), (5,9), (4,8), (4,9), (7,9) are some of the combinations where the classifier is confused in predicting the right label.

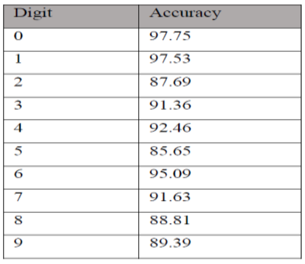


	0	1	2	3	4	5	6	7	8	9
0	870	0	3	5	2	5	31	1	35	28
1	0	1079	2	1	0	0	10	0	38	5
2	79	25	266	91	5	2	269	4	271	20
3	32	39	6	353	2	3	51	8	409	107
4	19	2	5	4	168	7	63	7	210	497
5	71	25	1	20	3	44	40	2	586	100
6	12	12	3	1	1	7	895	0	26	1
7	0	15	2	10	5	1	5	280	39	671
8	13	72	3	7	3	11	12	4	648	201
9	5	7	3	6	1	0	1	13	18	955

We can express 784 individual Gaussian distributed as a one long string of multivariate Gaussian, we can infer from that covariance matrix Σ except the diagonal of the matrix everything else will be zeros, hence only the 784 variances are stored at the diagonal.

The downside in the Naive Bayes classifier is that it assumes the all the dimensions present in the data set is independent to one another and which we all know that it’s not correct.

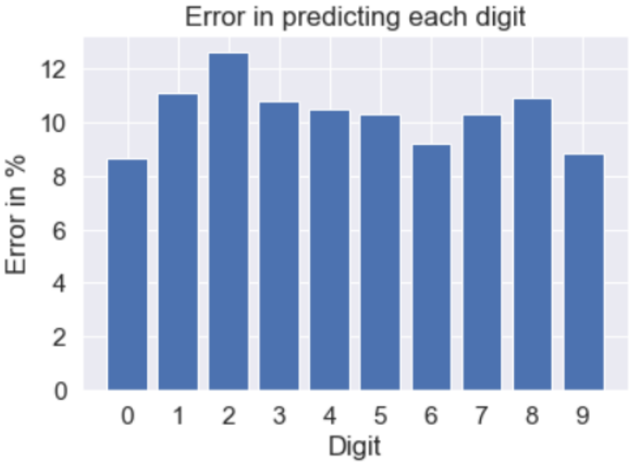
On accuracy plotting for each digit below is being observed



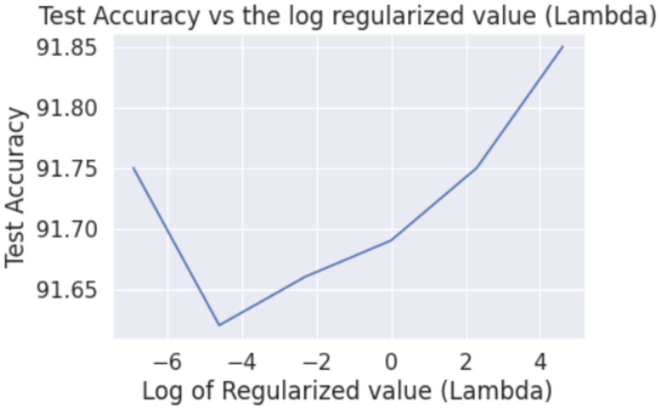
Digit	Accuracy
0	97.75
1	97.53
2	87.69
3	91.36
4	92.46
5	85.65
6	95.09
7	91.63
8	88.81
9	89.39

Overall accuracy for different lambda values were also tabulated below:

Lambda	0.0001	0.001	0.01	1	10	100
Accuracy	91.75	91.62	91.66	91.69	91.75	91.85



As we observe from the table above the accuracy is maximum for the lambda value 100. Also, computationally lambda=100 the classifier performs faster. But a constraint is that the selection of value was not dynamic in the code. The observation is only for this dataset and our model.



IV. Lessons learned

The conditional probabilities for each class given an attribute value are small. The product of them results in much small values, which can very difficult to represent in a programming language (floating point underflow).To fix this issue the log of the probabilities are added together.

was employed, it can also be extended at other distributions. The major difference would be different assumptions and relationship to the class functions.

Also this can be extended to support nominal attributes. Although in this implementation Gaussian Naive Bayes Naive Bayes algorithms are mostly used in sentiment analysis, spam filtering, and recommendation systems etc. They are fast and easy to implement but their biggest disadvantage is that the requirement of predictors to be independent. In most of the real life cases, the predictors are dependent, this hinders the performance of the classifier [8].

Logistic regression is more efficient than the Naive Bayes algorithm, the overall accuracy obtained is 91.85% where in Naive Bayes ended up 61.82% test accuracy. Logistic Regression being complex as compared to the Naïve Bayes algorithm, Logistic Regression model performed well in classifying the images.

Logistic Regression is more computationally expensive than Naive Bayes. The Naive Bayes took less than a minute to train and predict the labels, whereas Logistic Regression took about an hour to train and predict the labels.

Nave Bayes classifier assumes all the dimensions as independent to one another which is not true.

V. References

[1] "Support vector machines speed pattern recognition - Vision Systems Design". Vision Systems Design. Retrieved 17 August 2013

[2] Gangaputra, Sachin. "Handwritten digit database". Retrieved 17 August 2013.

[3] Qiao, Yu (2007). "THE MNIST DATABASE of handwritten digits". Retrieved 18 August 2013.

[4] Platt, John C. (1999). "Using analytic QP and sparseness to speed training of support vector machines" (PDF). Advances in Neural Information Processing Systems: 557–563. Archived from the original (PDF) on 4 March 2016. Retrieved 18 August 2013.

[5] Grother, Patrick J. "NIST Special Database 19 - Handprinted Forms and Characters Database" (PDF). National Institute of Standards and Technology.

[6] LeCun, Yann; Cortez, Corinna; Burges, Christopher C.J. "The MNIST Handwritten Digit Database". Yann LeCun's Website yann.lecun.com. Retrieved 30 April 2020.

[7] Kussul, Ernst; Baidyk, Tatiana (2004). "Improved method of handwritten digit recognition tested on MNIST database". Image and Vision Computing. 22 (12): 971–981. doi:10.1016/j.imavis.2004.03.008.

[8] https://medium.com/@meetpatel12121995/naive-bayes-machine-learning-algorithm-aaf57bdc8d87