Individual Portfolio Report

Nigar Sultana
Arizona State University
Ira A. Fulton School of Engineering
nsultan5 @asu.edu

***Abstract*—This electronic document is the individual portfolio report for Online MSC course, CSE575 Statistical Machine Learning.**

Keywords—Naïve Bayes, K-Means, Neural Network, probability, centroids

I. Introduction

This course project work is an individual project. The course needs three project works to be submitted which cover a wide range of areas in Statistical Machine Learning including Naïve Bayes Classifier, K-Means strategy and Neural Networks and Deep Learning. Throughout this portfolio report, I will provide a detailed overview of each project, including the methodology, results, and insights gained.

A. Project One

Project One mandates the utilization of data that is exclusive to our Student ID, implementing Naive Bayes Classifier to extract the features, computing the mean and variance of the images, and ultimately determining the accuracy of the results. A pre-existing function is provided to load the trainset and testset for digit0 and digit1, respectively. These sets are derived from the MNIST dataset, which comprises 70,000 handwritten digit images partitioned into 60,000 training images and 10,000 testing images. Only a portion of images for digit "0" and digit "1" is used in this project, resulting in the following dataset statistics:

(a) Number of samples in the training set: "0": 5000; "1": 5000.

(b) Number of samples in the testing set: "0": 980; "1": 1135.

Although the two digits possess varying quantities of samples in testing sets, we assume that the prior probabilities are equivalent (P(Y=0) = P(Y=1) = 0.5). All datasets are represented as Numpy Arrays. The project is comprised of four tasks.

For Task 1,our initial objective is to extract features from the original trainset to transform the data arrays into 2-dimensional data points. Specifically, we must extract two features for each image. The first feature we need to extract is the average brightness of the image, which involves averaging the brightness values of all pixels within the entire image array.

We also need to extract the standard deviation of the brightness of each image as the second feature. This involves calculating the standard deviation of all the pixel brightness values within the whole image array. It is assumed that both of these features are independent and that each image is sampled from a normal distribution.

For Task 2, our next step is to compute the parameters for the two-class naive Bayes classifiers. This calculation will be based on the 2-dimensional data points that we generated in Task 1. We will need to calculate the parameters separately for each of the two classes.

We need to compute the following eight parameters for the two-class naive Bayes classifiers:

1.Mean of feature 1 for digit 0 2.Variance of feature 1 for digit 0 3.Mean of feature 2 for digit 0 4.Variance of feature 2 for digit 0 5.Mean of feature 1 for digit 1 6.Variance of feature 1 for digit 1 7.Mean of feature 2 for digit 1 8.Variance of feature 2 for digit 1

In Task 3, we will use the parameters of the naive Bayes classifiers obtained from Task 2 to implement their respective calculation formulas as per their mathematical expressions. Using the implemented classifiers, we will then predict/classify all unknown labels of newly incoming data points. For this task, we will work with the test sets for digit 0 and digit 1 and predict all the labels for them.

After successfully predicting the labels for all the test data in Task 3, our objective in Task 4 is to calculate the accuracy of our predictions for the test sets of both digit 0 and digit 1.

B. Project Two

We are given two parts as part of Project Two:

Part 1 of our task involves implementing the K-means algorithm and applying it to a given dataset consisting of a set of 2-dimensional points. We must follow the strategy of choosing the initial cluster centers as described in Strategy 1, which involves randomly selecting the initial centers from the given samples. We will test the implementation on the provided data by varying the number of clusters, k, from 2 to 10. For each case, we will output the final coordinates of the centroids and compute the loss based on the objective function.

Part 2 of our task involves implementing the K-means algorithm and applying it to a given dataset consisting of a set of 2-dimensional points. We must follow the strategy of choosing the initial cluster centers as described in Strategy 2, which involves randomly selecting the first center, and for the i-th center (i > 1), choosing a sample (among all possible samples) such that the average distance of this chosen one to all previous (i-1) centers is maximal. We will test the implementation on the provided data by varying the number of clusters, k, from 2 to 10. For each case, we will output the final coordinates of the centroids and compute the loss based on the objective function.

C. Project Three

In this section, our task is to comprehend the entire process of building a simple Convolutional Neural Network (CNN)

for visual classification purposes by combining different layers, such as Convolutional, Fully-Connected, Pooling, Activation, and Loss. Furthermore, we need to create our own evaluation code to assess the trained CNN and obtain the training and testing results. The layer definitions are already provided in the demo code, and we followed the steps to understand the underlying concepts of various layers.

The dataset we will use for classification purposes is a subset of the MNIST dataset. The demo code will randomly select four different categories and provide us with 500 training and 100 testing samples for each category. Thus, the total size of the training and testing samples is 2000 and 400, respectively, and the subset training and testing samples will be shuffled before providing them to us.

We need to write a function for the evaluation code to obtain the accuracy and loss for both the training and testing samples. Furthermore, we are required to train the CNN with a fixed epoch number and parameter initialization. The total epoch number is set to 10, and the learning rate is set to 0.001. The batch size for the training and testing process is set to 100 and 1, respectively. The convolutional layer should contain six feature maps, and the filter size is set to 55. The pooling layer size is 22, and the ReLU activation function is set to default. The number of neurons in the first fully-connected layer is set to 32, and we will use the cross-entropy loss with softmax activation function to train the CNN.

II. DESCRIPTION OF SOLUTION

A. *Project One*

For Project One, our task is to create a Naïve Bayes Classifier that will predict the labels of unknown data points. We have been provided with a training data set and have extracted the average brightness and standard deviation of each image in the data set. Based on this, we have calculated the mean and variance of both feature one and feature two for digit one and digit two. In total, we have obtained eight parameters that will be used to develop a mathematical expression to predict the label of a given input.

B. *Project Two*

For part two of this project we have two tasks:

Task 1 involves implementing the k-means algorithm using a training data set of 2-D points. The algorithm will randomly select initial centroids, and then we will run it against a provided test data set. The final step is to compute the loss based on the objective function.

Task 2 requires us to select the first center randomly, and then for subsequent centers ($i > 1$), we choose a sample that has the highest average distance from all previously chosen centers. We will then test this algorithm against a given output data set and compute the final centroid for the K-runs, as well as the loss based on the objective function.

C. *Project Three*

As part of project Three, we are given a partially written code for a CNN. We are asked to implement an evaluation function which will calculate the testing and training accuracies.

The requirements for CSE575 Statistical Machine Learning project have been fully met. As these were individual projects, all the work presented here has been solely contributed by me to fulfill the requirements.

III. RESULTS

A. *Project One*

Below are the estimated parameters
Mean_of_feature1_for_digit0 : 44.2434670918
Variance_of_feature1_for_digit0 :114.337528667
Mean_of_feature2_for_digit0 : 87.4766929442
Variance_of_feature2_for_digit0 : 99.9275409841

Mean_of_feature1_for_digit1 : 19.3745982143
Variance_of_feature1_for_digit1 : 31.2356382734
Mean_of_feature2_for_digit1 : 61.3624241811
Variance_of_feature2_for_digit1 : 82.332695446

Below is the calculated accuracy

Accuracy_for_digit0_testset : 0.9173469387755102
Accuracy_for_digit1_testset :0.9233480176211454

B. *Project Two*

K-Means Strategy One
Initial Centroids 1: [[6.11106851,6.23497555], [2.10054891 ,1.44144019], [8.87578072,8.96092361]] K2: 5

Initial Centroids 2: [[1.79534908,3.7348206], [6.03237178, 8.86195452], [8.03150205,8.88381354],[7.39793659,2.191 43804], [2.04945194,2.75937105]]
Cost1:1293.7774523911348

Final Centroids 1: [[2.56146449,6.08861338], [5.47740039, 2.25498103], [6.49724962,7.52297293]]

Cost2: 613.2824392056041
Final Centroids 2: [[2.60123296,6.91610506], [5.40252508,6.73636175], [7.75648325,8.55668928],

[7.25262683,2.40015826], [3.21257461,2.49658087]]
Once the centroids have been calculated, we need just run the old K-Means algorithm from the previous section, which is shown in Fig 1.

In Fig 1 & 2, as we increase the number of clusters, the loss function decreases because there are more centroids available to assign data points to.This means that the distance between the data points and their assigned centroids is smaller, resulting in a lower loss function value.

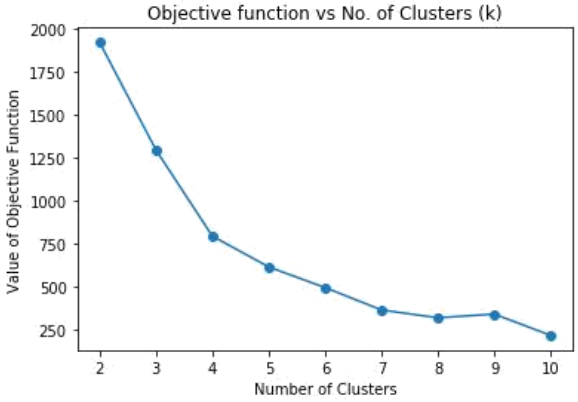

Figure 1: K-Means algorithm's Objective Function

K-Means Strategy Two
Below are the initial K-values and initial centroids provided.
K1: 4

Given first Centroids1: [6.40483149, 5.60578084]

Calculated Initial Centroids1: [[6.40483149, 5.60578084],[3.8 5212146,-1.08715226],[9.26998864,9.62492869],[1.201622 48, 7.68639714]]

Final Centroids 1: [[6.79532432, 2.78778512], [2.85235149, 2.28186483],[6.92822285,7.92187152],[3.19669343, 6.871 2608]]
Cost1:803.2167238057567

K2: 6

Given first Centroids2: [3.66118224, -0.63372377]
Calculated Initial centroid2: [[3.66118224,-0.63372377],[9. 26998864, 9.62492869],[1.20162248 ,7.68639714],[7.6809 7556 ,0.83542043],[2.95297924 ,9.65073899],[3.85212146 ,-1.08715226]]
Final Centroids 2: [[3.49556658, 3.56611232], [7.75648325,

8.55668928], [2.56333815, 6.9782248], [7.41419243, 2.32169114], [5.46427736, 6.83771354], [3.14506148, 0.90770655]]
Cost2: 476.118751676

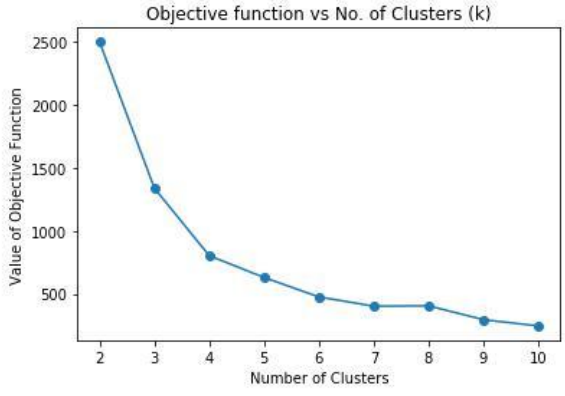

Figure 2: K-Means++ algorithm's Objective Function

The objective function versus cluster numbers graph can help to identify the optimal number of clusters for a particular dataset. The goal is to choose a value of K that results in a low value of the objective function while still maintaining a reasonable level of cluster coherence and interpretability.
From Fig 1& 2 it is observed that the K-means algorithm depends on the initial settings of the centroid, it is better to take different estimated initial centroids values to find for which cost function value is less.

C. Project Three

Below are the output values that we got for the
project training accuracy:0.894
training loss: 0.304
testing accuracy:0.887
testing loss: 0.343

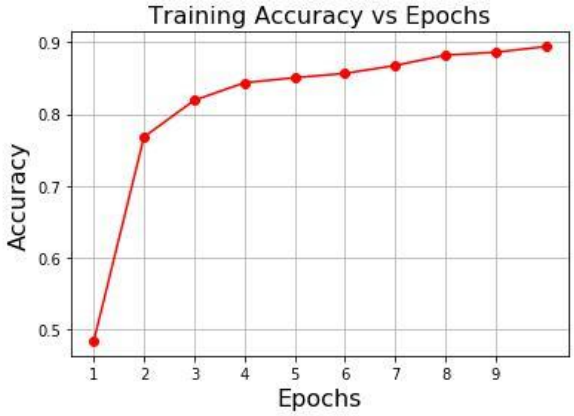

Figure 3: Training Accuracy vs Epochs

In Fig 3, at the start of the training process, the accuracy is usually low, as the CNN has not yet learned to recognize the patterns in the images. As the number of epochs increases, the CNN updates its internal parameters based on the gradients of the loss function, and the accuracy improves.

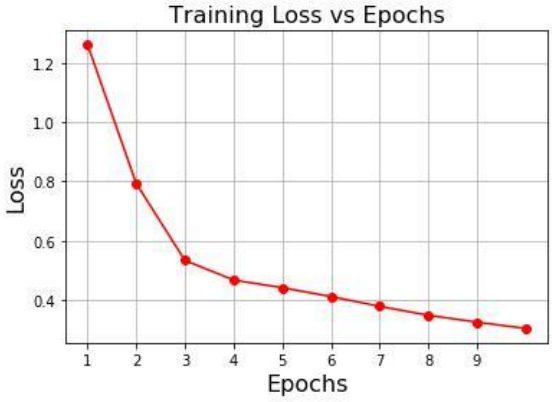

Figure 4:Training Loss vs Epochs

In Fig 4, at the start of the training process, the loss function is usually high, as the CNN has not yet learned to recognize the patterns in the images. As the number of epochs increases, the CNN updates its internal parameters based on the gradients of the loss function, and the loss function decreases.

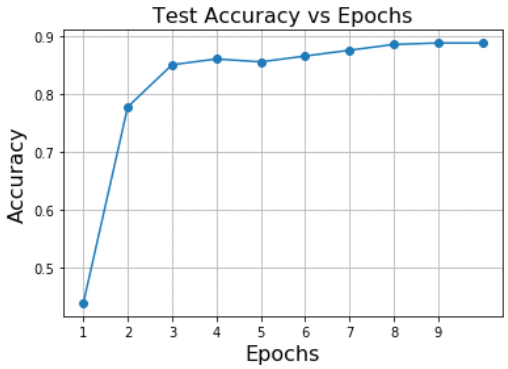

Figure 5: Test Accuracy vs Epochs

In Fig 5, at the start of the training process, the test accuracy is usually low.As the number of epochs increases, the CNN updates its internal parameters based on the gradients of the loss function, and the accuracy improves.The graph typically shows a trend of increasing accuracy over time.

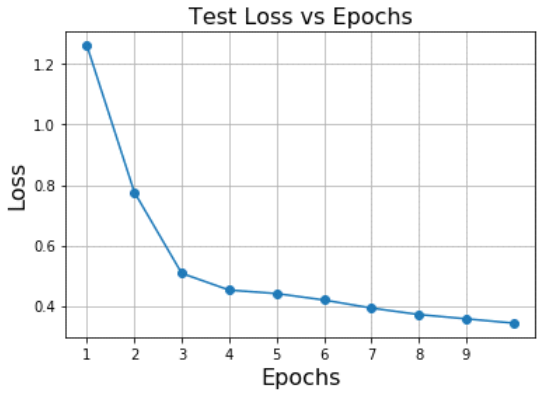

Figure 6:Test Loss vs Epochs

In Fig 6, at the start of the training process, the test loss is usually high, as the CNN has not yet learned to recognize the patterns in the images. As the number of epochs increases, the CNN updates its internal parameters based on the gradients of the loss function, and the test loss decreases.The graph typically shows a trend of decreasing test loss over time.

IV. Lessons Learned

I have a background in physics, having completed my graduation, and have more than 10 years of experience in teaching. Machine learning is a completely new field for me, and I have found it both interesting and significant in my life. During this statistical machine learning course, I have learned several fascinating and useful topics such as k-means clustering and it's variant k means++, artificial neural network (CNN), naive Bayes classifier, and Maximum Likelihood Estimation(MLE).

I aspire to become a data scientist, and I believe that the knowledge I have gained from this machine learning course is fundamental to the field. I plan to integrate this newly acquired knowledge into my daily work routine and use it to develop innovative ideas. This course has equipped me with the necessary skills to think differently, which will help me to make a unique and effective contribution to my organization's success. I am confident that this newfound knowledge will enable me to drive the company's organizational goals forward and help us succeed together.

V. References

[1]https://en.wikipedia.org/wiki/Machine_learning
[2]https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/
[3]https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c
[4]https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a
[5]https://www.analyticsvidhya.com/blog/2018/12/guide-convolutional-neural-network-cnn/

VI. Acknowledgements

The preparation of the files was done as part of submission of CSE575 Project. Thanks for reading this.