# MNIST Classification and K-means Clustering

Junggil Park
ASU ID: 1223940471
*ASU Master of Computer Science,*
*Ira A. Fulton Schools of*
*Engineering*
Arizona, US
johnjgpark@gmail.com

***Abstract*— This document is a portfolio report on MNIST handwritten digit classification and K-means clustering for a statistical machine learning course. This paper has three main objectives: classification of MNIST datasets by using Naïve Bayes Classifier; classification of MNIST datasets by using Convolutional Neural Network (CNN); clustering the given dataset by using the K-means algorithm. Background details of machine learning technologies, the approach to the solution, and quick snapshots of the results are provided in each method.**

***Keywords*— *MNIST, classification, K-means, clustering, Convolutional Neural Network, machine learning (keywords)***

## I. INTRODUCTION

Statistical machine learning is the study of computer algorithms performing tasks automatically through the experience without being explicitly programmed to do so [1]. Machine learning algorithms make a model based on training data, in order to predict output or label from testing data.

Machine learning approaches are traditionally divided into three broad categories, but this paper involves only two categories which are supervised learning and unsupervised learning.

Supervised learning algorithms build a statistical model of a dataset that includes both the inputs and the labels [2]. The data which is known as training datasets consists of a set of training examples. Classification is an example of supervised learning. This paper involves two methods for classification which are Naïve Bayes Classifier and deep learning such as Convolutional Neural Network (CNN).

Unsupervised learning algorithms include a set of only input data and find patterns in the data, like grouping or clustering of data points. Thus, the algorithms learn from the testing dataset that is not labeled, classified, or categorized. One of the most popular unsupervised learning algorithms is K-means clustering.

This paper has three main objectives: 1) Naïve Bayes Classifier; 2) classification of MNIST datasets by using Convolutional Neural Network (CNN); 3) clustering the given dataset by using the K-means algorithm. Each technology includes the background of machine learning technology, the approach to the solution, and quick snapshots of the results. The three methods show how to use machine learning technologies in order to analyze datasets in practice.

## II. NAÏVE BAYES CLASSIFIER

By using Naïve Bayes Classifier, MNIST image datasets are classified as either digit “0” and digit “1”. The MNIST datasets could be handwritten image digits which are generally used to train and model diverse image processing systems.

### *A. Background*

In statistics and machine learning, naive Bayes classifiers are simple probabilistic classifiers which utilize Bayes' theorem with assumptions of independence between the features, which are among the easiest Bayesian network models [3]. In other words, a naive Bayes classifier considers each of the features to assume independently the probability, without any possible relationship between the features.

Total 4 sets of MNIST image datasets have been downloaded. Two sets are for training, and the other two sets are for testing. 5000 samples in the training set are for the digit “0”, and 5000 samples in the training set are for the digit “1”. Also, 980 samples in the testing set are for the digit “0”, and 1135 samples in the testing set are for the digit “1”.

In figure 1, Each sample in a set has a 28 x 28 matrix to represent digit 0 or digit 1. For example, the value of 0 in a matrix set means white, the value of 255 means black. Grayscale color can be represented in between 0 and 255

This project uses the MNIST datasets to extract two features for each image. Feature 1 is the average of all pixel brightness values within a whole image array. Feature 2 is the standard deviation of all pixel brightness values within an image array matrix. For naive Bayes and MLE Density Estimation, it is assumed that these two features are independent and that each image is drawn from a normal distribution.

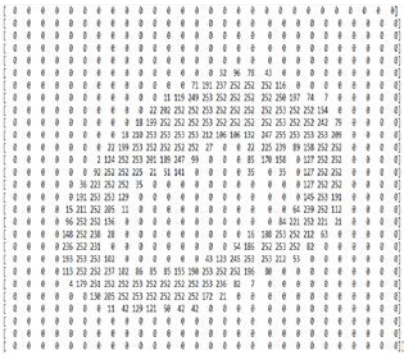

Figure 1. 28 x 28 matrix for digit 0

### B. Solution

Total 8 parameters are required for the modeling of the two-class naive Bayes classifiers. The mean and variance of feature 1(mean), and The mean and variance of feature 2(standard deviation) for 5000 training sets of digit 0. The mean and variance of feature 1(mean), and The mean and variance of feature 2(standard deviation) for 5000 training sets of digit 1.

Table 1. 8 parameters

| digit label | feature1 (mean) | feature1 (variance) | feature2 (mean) | feature2 (variance) |
|---|---|---|---|---|
| 0 | 44.283 | 114.410 | 87.511 | 100.356 |
| 1 | 19.335 | 31.346 | 61.296 | 82.046 |

This project uses use the Gaussian probability distribution function (GPDF) f(x) to be able to classify/predict unknown labels which are digit 0 or digit 1 in the testing dataset. The Gaussian function is used to represent the normally distributed probability density function such as below.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \quad (1)$$

In the above equation, f(x) is a Gaussian function, μ is a mean or expected value, $\sigma^2$ is a variance, and x is a mean(feature1) or standard deviation(feature2) of each testing set. Below formulas are used for Naïve Bayes Classifier. It is assumed that these two features are independent.

$$P(0|feature1,feature2) = P(0)P(feature1|0)\ P(feature2|0) \quad (2)$$

$$P(1|feature1,feature2) = P(1)P(feature1|1)\ P(feature2|1) \quad (3)$$

,where P(0)=P(1)=0.5

In the above equation (2), given feature1 and feature2, the probability of digit 0 is equal to or less than 1. In the above equation (3), given feature1 and feature2, the probability of digit 1 is equal to or less than 1. After calculating both the probability of digit 0 and the probability of digit 1 by using GPDF equation (1), it is possible to classify unknown labels in the testing set. If the probability of digit 0 is greater than the probability of digit 1, the unknown label is digit 0. If the probability of digit 1 is greater than the probability of digit 0, the unknown label is digit 1.

### C. Result

After successfully predicting the labels for all testing datasets, the accuracy of classification has been calculated for the testing dataset for both digit 0 and digit 1 respectively. The accuracy of the digit 0 testing dataset is 91.735 and the accuracy of the digit 1 testing dataset is 92.335, which results in quite a high accuracy using relatively simple classification.

## III. Convolutional Neural Network

By using a simple Convolutional Neural Network (CNN), MNIST image datasets are classified for the visual classification task. The MNIST datasets could be handwritten image digits which are generally used to train and model diverse image processing systems.

### A. Background

The convolutional neural network (CNN), one of the deep learning methods and a type of artificial neural network, is commonly utilized for analyzing visual images and being able to differentiate one from the others [4].

In figure 2, for the MNIST classification task, this paper considers a simple Convolutional Neural Network (CNN), including the whole process of compiling different layers (Convolutional Layer, Fully-Connected Layer, Max-Pooling Layer, ReLU Activation Layer, Loss function). Also, evaluation graphs are required to evaluate the trained CNN model to obtain the training and testing results with snapshots.

Similar to part1, the training and testing datasets from the MNIST are used. The given demo codes randomly select four different categories and 500 training and 100 testing samples for each category. Therefore, the total size of the training and testing samples is 2000 and 400 respectively. The batch size for training is 100, so 100 batches for 20 times are required to be trained. For CNN with a fixed epoch number and initialization of parameters, the fixed epoch number should be 10 and the learning rate should be 0.001. The batch size for the training is set to 100 and the testing process is set to 1. The number of feature maps in the convolutional layer should be 6, the size of the filters is set to 5x5, the size of the pooling layer is 2x2, and the ReLU activation function is set to default. The neurons' number of the first fully connected layer is set to 32. A cross-entropy loss with softmax activation function is utilized to train the CNN modeling.

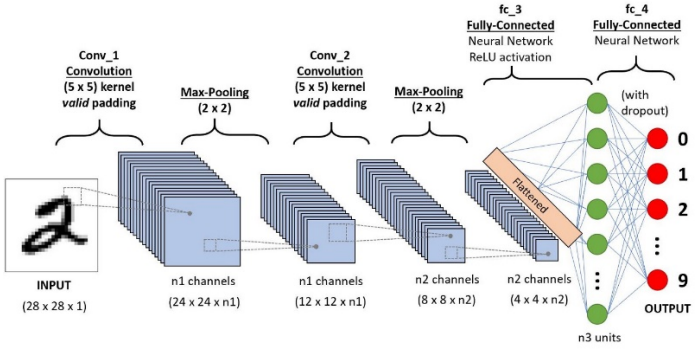


Figure 2. CNN architecture to classify handwritten digits [5]

### B. Solution

Without library tools such as TensorFlow, Keras, or PyTorch, the demo code is given for CNN project, which makes it possible to look at algorithms and read specific functions inside CNN. After training and testing the CNN with datasets which the total size of the training and testing samples is 2000 and 400 respectively, four values which are final training accuracy, training loss, testing accuracy, and testing loss, should be found after 10 epochs from python code like below.

```
=== Epoch:9 Train Size:2000,
Train Acc:0.839, Train Loss:0.435 ===
=== Epoch:9 Test Size:400,
Test Acc:0.785, Test Loss:0.564 ===
```

Figure 3. final four values after 10 epochs

The following four plots show training accuracy vs epochs, training loss vs epochs, testing accuracy vs epochs, and testing loss vs epochs.

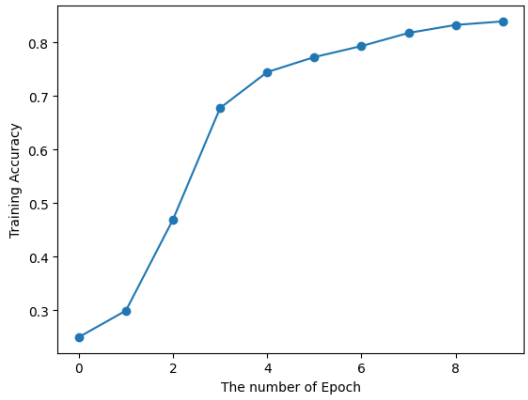


Figure 4. training accuracy vs epochs

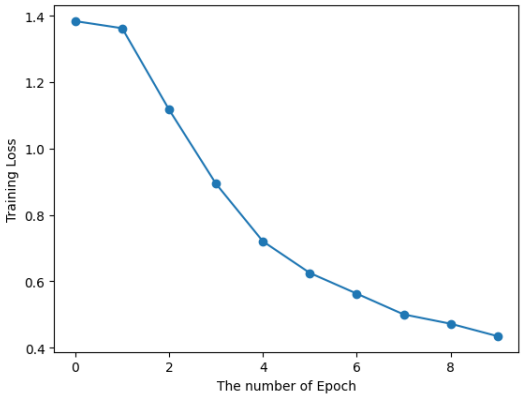


Figure 5. training loss vs epochs

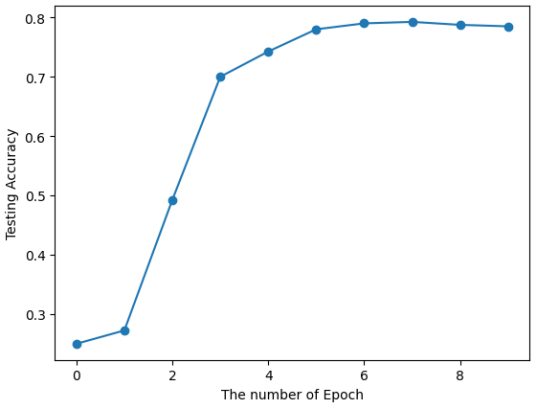


Figure 6. testing accuracy vs epochs

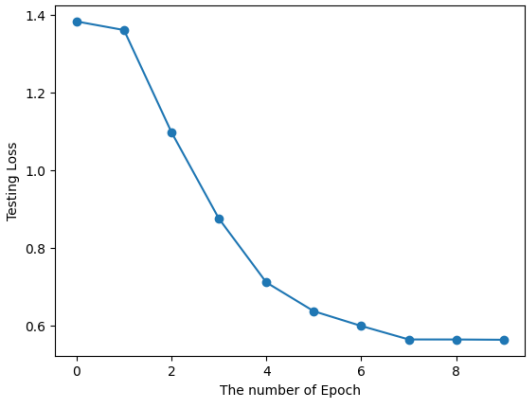


Figure 7. testing loss vs epochs

### C. *Result*

From the plots, both training and testing accuracy is increasing as the number of the training set (Epoch) is increasing, and both training and testing loss is decreasing as the number of the training set (Epoch) is increasing. Also, the plots show they are saturating at 7 or 8 Epochs for both the training set and testing set. In contrast to machine learning, training accuracy in deep learning increases as increasing training set. However, testing accuracy is saturating and even decreasing at 8 or 9 Epochs. Therefore, we should stop training set at 8 or 9 Epochs, in order to prevent overfitting.

## IV. K-MEANS CLUSTERING

This part uses the k-means algorithm for clustering and applies the implementation on the given dataset, which contains a set of 2-Dimension points.

### A. *Background*

Unsupervised learning algorithms include a set of only input data and find patterns in the data, like grouping or clustering of data points. Unsupervised learning learns from the testing dataset which is not labeled, classified, or categorized.

K-means clustering, one of the popular unsupervised learnings, targets to partition data points into k clusters in which each data point belongs to the cluster with the nearest cluster centroid. That is to say, the K-means algorithm figures out the k number of center points, and assigns all data points to the closest cluster, making the center points as small as possible.

The main issue of K-means clustering is where to locate initial centroids and how many initial centroids. This project introduces two strategies; 1) randomly pick the initial centers from the given samples; 2) pick the first center randomly, then choose a sample (among all possible samples) such that the average distance of this chosen one to all previous centers is maximal.

### B. *Solution and Result*

Strategy1 randomly picks the initial centers from the given samples and chooses the k initial cluster centers at k=3 and 5. Without sklearn library in python, final cluster centers and Sum of Squared Errors (SSE) are found at k=3 and 5. SSE is described as the sum of the Euclidean distance between the centroid and each point of the cluster. Therefore, lower SSE means generally better clustering.

```
Initial Cluster Centers
[[ 7.52963009  8.79617112]
 [ 1.20162248  7.68639714]
 [ 7.1712312   5.16316266]]
Final Cluster Centers
[[ 6.49724962  7.52297293]
 [ 2.56146449  6.08861338]
 [ 5.47740039  2.25498103]]
SSE:  1293.77745239
```

```
Initial Cluster Centers
[[ 2.58046907  6.53023549]
 [ 1.81229618  3.40781697]
 [ 2.0614632   8.22584366]
 [ 1.51180219  7.48293717]
 [ 7.25412082  2.77862318]]
Final Cluster Centers
[[ 5.29629878  6.64908797]
 [ 3.21257461  2.49658087]
 [ 7.75648325  8.55668928]
 [ 2.51976116  7.02028909]
 [ 7.25262683  2.40015826]]
SSE:  613.986628607
```

Figure 8. initial and final cluster centers, and SSE at 3 and 5 cluster centers for Strategy1

However, the optimum number of clusters needs to be selected. If the number of clusters is equal to the number of points, SSE is zero which means each cluster has only one point. Therefore, the elbow method is one of the most popular methods which is used to select the optimal number of clusters by fitting the model with a range of values for K in the K-means algorithm. To figure out the optimal number of clusters, plotting the objective function values(SSE) vs the number of clusters in the range from 2 to 10 is required.

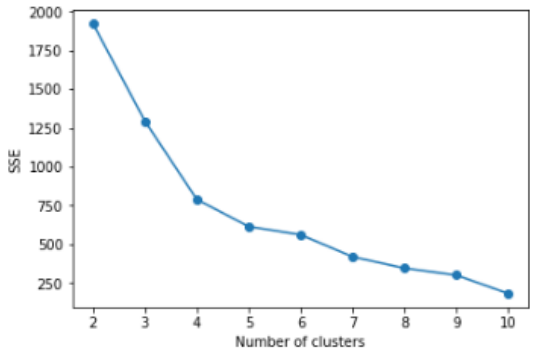


Figure 9. SSE vs number of clusters for strategy 1

Based on figure 9, the optimum number of clusters for the k-mean clustering is 4 or 5, at which the elbow is located.

Strategy2 picks the first center randomly, then chooses a sample (among all possible samples) such that the average distance of this chosen one to all previous centers is maximal. In other words, finding furthest initial centroids, after picking the first center randomly at k=4 and 6

```
Initial Cluster Centers
[[ 7.12751003  1.23747391]
 [ 2.95297924  9.65073899]
 [ 9.26998864  9.62492869]
 [ 3.85212146 -1.08715226]]
Final Cluster Centers
[[ 6.78374609  2.85019999]
 [ 3.34264769  6.92602803]
 [ 7.17928621  8.0520791 ]
 [ 2.85235149  2.28186483]]
SSE:  805.116645747
```

```
Initial Cluster Centers
[[ 2.97661653  6.01021497]
 [ 9.26998864  9.62492869]
 [ 3.85212146 -1.08715226]
 [ 2.95297924  9.65073899]
 [ 6.5807212  -0.0766824 ]
 [ 8.87578072  8.96092361]]
Final Cluster Centers
[[ 3.502455    3.62870476]
 [ 7.75648325  8.55668928]
 [ 3.14506148  0.90770655]
 [ 2.52382885  7.02897469]
 [ 7.41419243  2.32169114]
 [ 5.46427736  6.83771354]]
SSE:  476.296570527
```

Figure 10. initial and final cluster centers, and SSE at 4 and 6 cluster centers for Strategy2

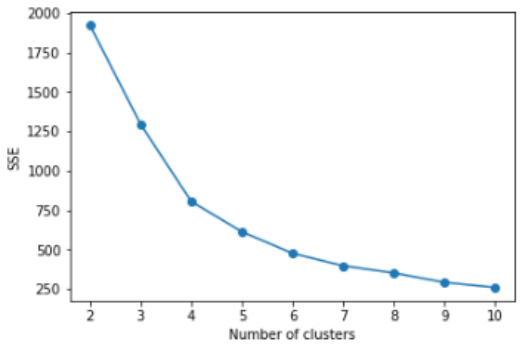


Figure 11. SSE vs number of clusters for strategy 2

Based on figure 11, the optimum number of clusters for the k-mean clustering is 4 or 5, at which the elbow is located.

## V. Lesson Learned and Conclusion

This paper investigated the performance of classification methods for MNIST handwritten digits datasets. Both machine learning method (naive Bayes classifiers) and deep learning method (CNN) are used for classification. Surprisingly, the accuracy of naive Bayes classifiers is higher than the accuracy of CNN, even though naive Bayes is much simpler. This is because the machine learning method is more useful for small training datasets such as given MNIST datasets. It is proved that the deep learning method should be only used in big datasets in terms of performance and accuracy.

This paper also included K-means clustering on the given dataset, which contains a set of 2-Dimension points. Depend on the initial centroid points and the number of clusters, the results would be very different. Since there are no labels for unsupervised learning such as clustering, choosing initial centroid points and the number of clusters is the most challenging and vital for reasonable clustering. Both random selection and the furthest selection of initial centroid points have good results to find the optimal number of clusters. But in figure 11, the plot of the furthest initial centroid points looks smoother than the graph in figure 9.

## VI. References


[1] T. Mitchell, *Machine Learning.* New York: McGraw Hill, 1997.

[2] S.J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach,* 3rd ed. Prentice Hall, 2010

[3] A. McCallum (2019, Oct 22). *Graphical Models, Lecture2: Bayesian Network Representation.* Available: https://people.cs.umass.edu /~mccallum/courses/gm2011/02-bn-rep.pdf

[4] M.V.Valueva, N.N.Nagornov, P.A.Lyakhova, G.V.Valuev, N.I.Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation", *Mathematics and Computers in Simulation,* Elsevier BV. 177: 232–243, Nov 2020.

[5] S. Saha. (2018, Dec. 15). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way* [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53