Optimization and Data Analytics Project

David Jensen, Aarhus University, Denmark, student nr. 11229

***Abstract*— This document gives the results of 5 classification methods. The data used on the methods is The MNIST (database of handwritten digits) and ORL (The Database of Faces)**

The 3 classification methods implemented is:

1. **Nearest class centroid classifier**
2. **Nearest sub class centroid classifier using number of subclasses in the set {2,3,5}**
3. **Nearest Neighbor classifier**
4. **Perceptron trained using Backpropagation**
5. **Perceptron trained using MSE (least squares solution)**

Keywords— Machine Learning; Image Classification;

nearest neighbor classifier, nearest centroid classifier,

Perceptron

I. Introduction

This report details a project done in Optimization and Data Analytics with a mini-project. The mini-project aims to implement and evaluate the performance of Nearest class centroid classifier, Nearest sub-class centroid classifier, using number of subclasses in the set {2,3,5}, Nearest Neighbor classifier, Perceptron trained using Backpropagation, Perceptron trained using MSE(least squares solution). The data used is the MNIST data set, which has a training set of 60000 and 10000 test samples. This is a small subset of the larger dataset, NIST. The hand written digits have been pre-processed so they are size-normalized and centred in a fixed-size image. This specific dataset gives a good intuition for newcomers in the field of learning techniques and pattern recognition methods on realistic data, without spending too much time doing pre-processing and formatting.

II. The Classification Schemes Used

A. *Nearest Class Centroid classifier*

Nearest class centroid classifier is a classification technique that uses already labelled observations of the data. Nearest Class Centroid is characterized by its centroids. In this classification each c_k has the following mean vector

$$u_k = \frac{1}{N_k}\sum_{i,l_i=k} x_i \,, k = 1, \dots, K$$

given the mean vectors a given $x_*$is classified by the distance

$$d(x_*, u_k) = ||x_* - u_k||_2^2$$

During the training phase each sample is mapped to a centroid, hence the mean which is the closest to the observation. In the testing phase each mean sample is classified to the nearest centroid classifier class.

B. *Nearest Sub Class Centroid classifier using the number of classes in the subclasses in the set{ 2,3,5 }*

The Nearest Sub Class Centroid classifier differs from the classical Nearest Class Centroid classifier because its algorithm splits the data into more subclasses. Predefining the subclasses m of c_k , given they follow a normal density distribution, they are represented by

$$u_{km} = \frac{1}{N_{km}} \sum_{i,l_i=k,q_i=m} x_i$$

where the number of subclasses is needed to be decided upon before starting. In order to define each subclass, we choose to apply a algorithm such as k-means.

C. *Nearest Neighbor classifier*

Nearest Neighbor classifier is a well known algorithm, that can be applied to classification problems. It works by defining an initial set of clusters, assigning all vectors to a cluster by

$$l^* = \arg min\, ||x_i - u_l||_2^2$$

and update the cluster mean vectors by

$$u_k^* = \frac{1}{N_k}\sum x_i \in D_k.[1]$$

D. *Perceptron trained using Backpropagation*

A perceptron trained using backpropagation is yet another way to solve classification problems when it is a linearly-separable case. It works by having a randomly initialized W and an eta value, also known as the learning rate, which is usually a small value. Then we identify all the training vectors that have been wrongly classified by

$$\sum_{x_i \in X} l_i w^t * x_i < 0$$

and if this is indeed the case then x_i is wrongly classified and assigned to X. After this this step the backpropagation occurs by updating W using

$$n(t) * \sum_{x_i \in X} l_i w^t * x_i$$

or in other words, in each run the vector is moved towards, the before mentioned vector by scaling it and therefor the result will be higher.

E. *Perceptron trained using MSE(least squares solution)*

The Perceptron trained using MSE, is a method to get a solution by using the least squares solution. It defines a matrix Y which has the first column specified with the first half of the rows as 1 and the remaining rows as -1, while in the other columns the categories are defined. Using the pseudoinverse a decision boundary can be found, and our solution is given by

$$a = y^\dagger b$$

where b is an arbitrary margin such as

$$b = (1,1,1,1)^t.$$

III. DATASETS

Two datasets are used, in this report each will be described below

A. MINST

MNIST dataset can be downloaded from the course blackboard page[2]. This dataset contains handwritten digits, which has a training set of 60,000 and 10,000 test samples. This is a small subset of the larger dataset, NIST. The hand written digits have been pre-processed so they are size-normalized and centred in a fixed-size image. This specific dataset gives a good intuition for newcomers in the field of learning techniques and pattern recognition methods on realistic data, without spending too much time doing pre-processing and formatting.

Figure 1 Sample from MNIST

B. ORL

The dataset ORL also known as Database of Faces. Have been used in face recognition projects, where the goal is to recognize the specific faces. It consists of ten different faces of 40 different people. As seen in figure 2 they were taken at difference times with varying lighting, facial expressions. Each image is 92x112 pixels, grayscale.

Figure 2 ORL preview images[3]

IV. ORGANIZING THE DATA FOR TESTING

A. Splitting the data.

The MNIST dataset from the course website comes pre-sorted into 60k for training and 10k for testing. But for the ORL dataset this is done manually in MATLAB[4] by splitting it into 70% training and 30% test images. This is done at random, but still uniformly hence the train and test sets will always contain samples from every class.

B. Pre-proccessing the data using Principal component analysis.

To make the data easier to plot and get the most important features Principal component analysis(PCA) is applied. An advantage of PCA is to make the classification faster to run, thereby optimizing the speed of classification. To accomplish this, the

$$m = \frac{1}{N}\sum_{i=1}^{N} y_i$$

is the mean vector of the data. This is used in the S_T also called the covariance matrix and calculating the eigen vectors. Of all the vectors the two largest ones were taken and multiplied by the data from the dataset currently being used. This approach allows us to extract the data, that has the biggest variance hence the data that matters the most for classification.

C. Evaluation criteria

Most of the classifiers can be validated by looking at their confusion matrix. A confusion matrix works by displaying the true positive, true Negative, false positive and positive false.

V. EXPERIMENTAL PROTOCOLS AND RESULTS

This section will now display the results that were archived in this article, while an in-depth discussion will come later. One thing to note is that the scores are generally low for the PCA data because of the massive reduction in dimensions.

For MNIST

A. Nearest Class Centroid classifier

PCA data: 43.6%
Raw data: 82.03%

B. Nearest Sub-Class Centroid Classifier 2

PCA data: 42.86%	2.7 sec
Raw data: 86.06%	105.1 sec

C. Nearest Class Centroid classifier 3

PCA data: 42.43%
Raw data: 88.18%

D. Nearest Sub-Class Centroid Classifier 5

PCA data: 41.93%
Raw data: 90.39%

E. Nearest Neighbor classifier

PCA data: 38.63%
Raw data: 96.91%

F. Perceptron with MSE

PCA data: 90.393%
Raw data: 96.35%

G. Perceptron with backpropagation

PCA data: 84.31%
Raw data: 98.5%

For ORL

H. *Nearest Class Centroid classifier*

PCA data: 38.33%
Raw data: 95%

I. *Nearest Sub-Class Centroid Classifier 2*

PCA data: 43.33%	0.8 sec
Raw data: 95.833%	1.1 sec

J. *Nearest Class Centroid classifier 3*

PCA data: 40%
Raw data: 97.5%

K. *Nearest Sub-Class Centroid Classifier 5*

PCA data: 40.833%
Raw data: 98.33%

L. *Nearest Neighbor classifier*

PCA data: 41.67%
Raw data: 98.33%

M. *Perceptron MSE*

PCA data: 97.5%
Raw data: 99.5%

VI. Visualize the 2D data

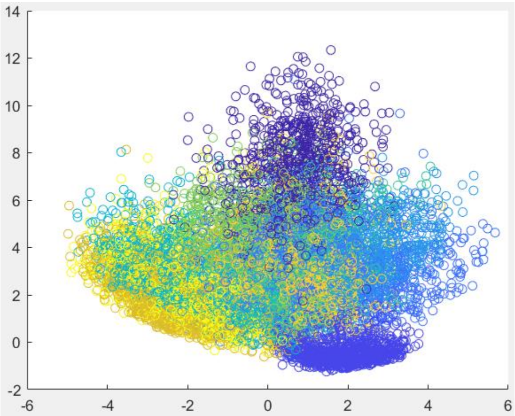

Figure 3 Scatter plot of the MNIST PCA data

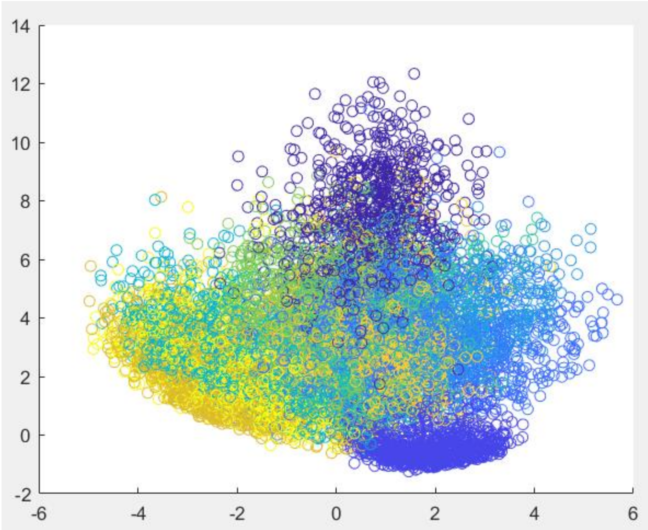

Figure 4 Scatter plot of MNIST PCA data

As it can be seen in Figure 3 Scatter plot of the MNIST PCA data and Figure 4 Scatter plot of MNIST the data is tired very close together. With some of the data appearing on the edges of the of the plot.

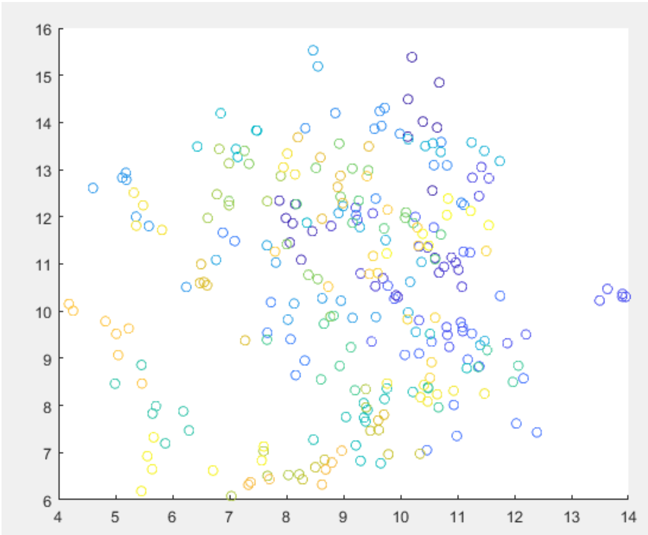

Figure 5 Scatter plot of ORL train PCA

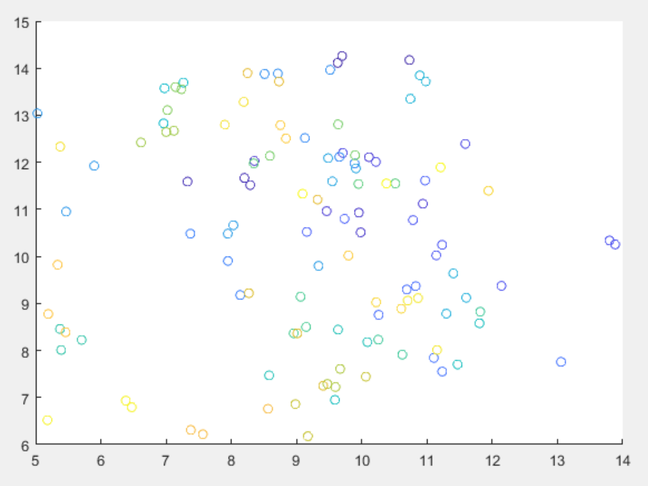

Figure 6 Scatter plot of ORL test PCA data

As it can be seen in Figure 5 Scatter plot of ORL train PCA and Figure 6 Scatter plot of ORL test PCA data. The data is scattered all over with no clear clusters.

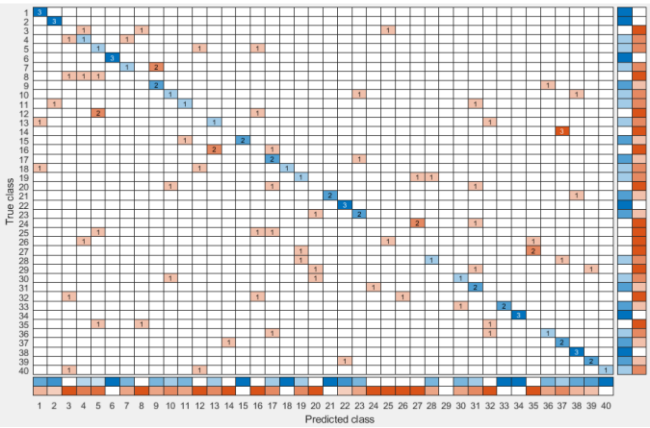

Figure 7 confusion matrix Nearest Neighbor classifier PCA

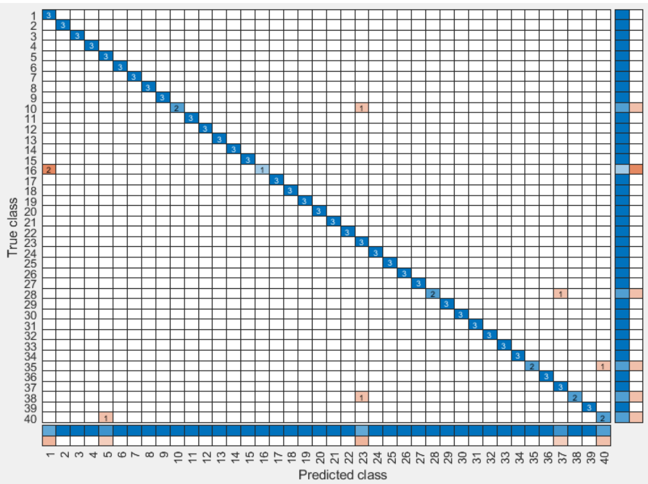

Figure 8 confusion matrix Nearest Neighbor classifier raw

As it can be seen in Figure 7 confusion matrix Nearest Neighbor classifier the prediction is low on the PCA data due to it's low dimensions. On Figure 8 confusion matrix Nearest Neighbor classifier raw it can be seen how much it matters when all data is included.

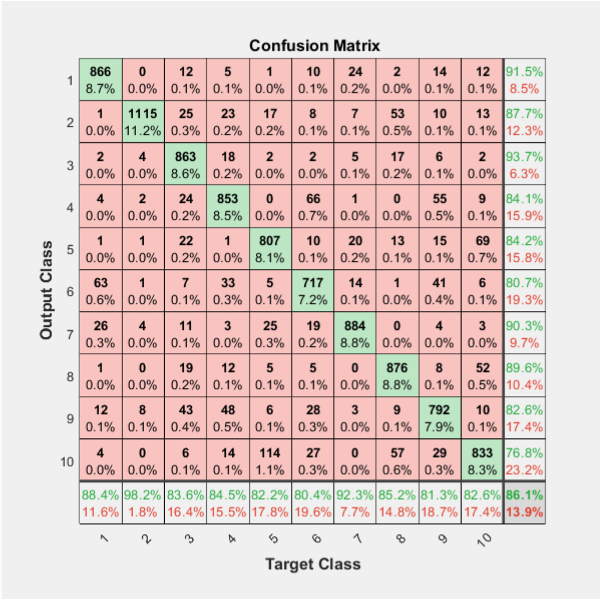

Figure 9 Nearest Sub-Class Centroid Classifier 2 PCA

Figure 10 Nearest Sub-Class Centroid Classifier 2 raw

As it can be seen in Figure 10 Nearest Sub-Class Centroid Classifier 2 raw and Figure 9 Nearest Sub-Class Centroid Classifier 2 PCA which has less dimensions and but much faster processing speed for classification, so this is a accuracy and speed tradeoff. Only processing time for Nearest Sub-Class Centroid Classifier 2 was benchmarked to settle the point.

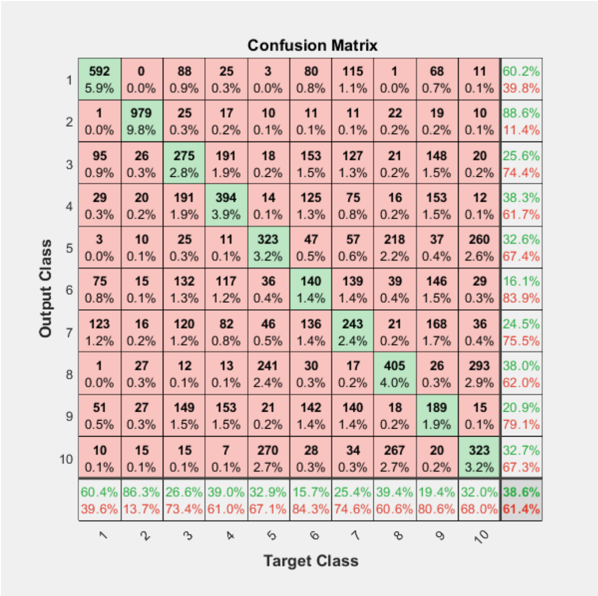

Figure 11 Nearest Neighbor classifier PCA

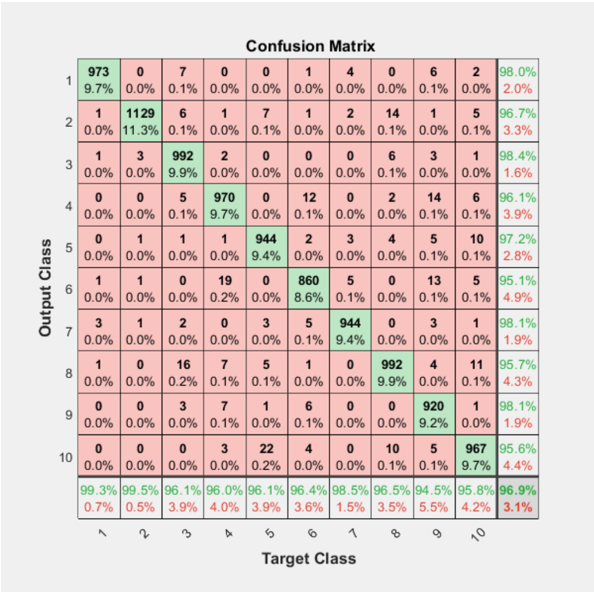

Figure 12 Nearest Neighbor classifier

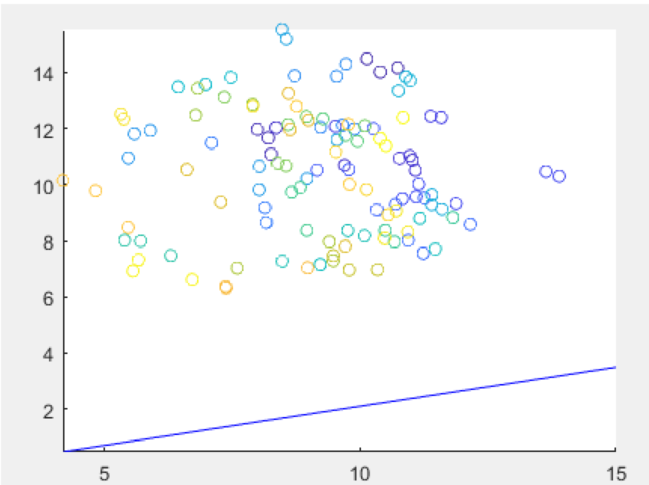
Figure 13 Perceptron LMS

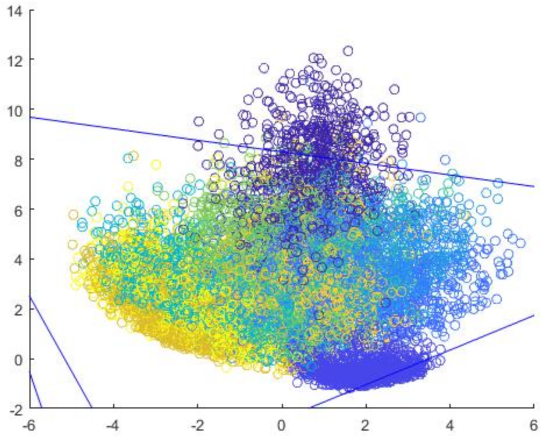
Figure 14 Perceptron LMS MNIST

VII. Discussion

As seen in the previous sections some of the classification methods perform better than others. Some important parameters include not only the different methods used but also how the data is represented. As seen in the results section the Perceptron MSE perform very well and achieve the best overall score on the ORL and MNIST dataset As it is show in the table Experimental protocols and results the different classifiers offer different accuracy, but it also depends on how the data is structured how well a algorithm will work.

VIII. Conclusions

As paper showed the results for the 5 different classification algorithms. Depending on the dataset and the algorithm, they will vary in speed and accuracy. So picking the best one will depend the use case. As seen with nearest sub-class centroid classifier 2 it matters a great deal if one reduced the dimensions both in accuracy and speed.

Acknowledgment

Aacknowledgment goes to the teachers and assistant teachers for always being ready to answer questions.

References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

[1] A. Iosifidis, "Introduction to Machine Learning," 2008.

[2] "Datasets," 2018. [Online]. Available: https://blackboard.au.dk/webapps/blackboard/content/listContent.jsp?course_id=_117051_1&content_id=_1892999_1&mode=reset. [Accessed: 25-Nov-2018].

[3] L. Cambridge, UniversityComputer, “ORL Data - The Database of Faces,” 2002. [Online]. Available: https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[4] MathWorks, “MATLAB,” 2018. [Online]. Available: https://se.mathworks.com/products/matlab.html. [Accessed: 05-Dec-2018].