Comparison of Two Applied Linear Algebra Techniques for Classification of Hand Written Digits: Centroids and Singular Value Decomposition

Kevin Chu

December 9, 2019

Contents

1. Abstract

This paper works to explore a data set of hand written digits using two classification techniques. Two methods of classification known as centroids and singular value decomposition will be used to explore this data set. The goal of this project is to apply these two methods of classification to the dataset to compare the accuracy between each method.

2. Introduction

The process taken for this project will be divided into four steps. The first step is to first visualize the data. This process involves viewing the first 16 images in the training patterns matrix. The next part of the visualization process is to view what the average hand written digit looks like in the training patterns matrix. The second step is to utilize the topic of centroids to try and predict the labels of the hand written digits in the testing patterns matrix. This third step is to utilize the topic of singular value decomposition to try and predict the labels of the hand written digits in the testing patterns matrix. The last step will be to summarize and analyze the differences between the results of each classification model.

3. Data Set

The file being worked with is the USPS hand written digits dataset. The file contains four different matrices. The four matrices are denoted as train_patterns, test_patterns, train_labels, test_labels.

In total, there are 9298 total patterns and labels in the dataset. For the purpose of testing the accuracy of classification algorithms, the dataset has been split into two. The matrices denoted as train are used to build models to predict the labels. The matrices denoted as test are used to check the accuracy of the models built.

The train_patterns and test_patterns matrices are of size 256x4649. Each row in these matrices represent a pixel of an image of a hand written digit. Each column in these matrices represents a different image of a hand written digit. The columns can be individually re-scaled to be a 16x16 matrix to plot and view each image.

The train_labels and test_labels are matrices of size 10x4649. Each column in these matrices corresponds to a column in the previous two matrices. Each row represents a number 0, ..., 9 and the value of each index of the row is either -1 or 1. A value of 1 indicates that the label is the row position. The value of each index in these two matrices represent the actual label of the pattern shown in the previous two matrices.

4. Classification Methods

4.1 Centroids

This algorithm works by pooling together the patterns of each digit D_i, i = 0, 1, ..., 9 from the training pattern data set. After each digit's patterns are pooled, the average pattern is then calculated for each digit as M_i, i = 0, 1,

..., 9. Next, the algorithm calculates the squared euclidean distance between all of the testing digits and the average pattern for each digit. The algorithm then computes the minimum value of the squared euclidean distances and then refers to the column index of the smallest one to predict the label of the test patterns. In essence, this algorithm attempts to classify digits based on the differences of testing patterns to the average training patterns.

$$\text{Euclidean Distance} = d(\vec{p}, \vec{q}) = \sqrt{(p_1 - q_1)^2 + ... + (p_n - q_n)^2} \tag{1}$$

4.2 Singular Value Decomposition

In singular value decomposition, any matrix A can be factored into three components: a matrix of orthogonal eigenvectors of AA^T, a matrix of orthogonal eigenvectors of A^TA, and a diagonal matrix of the singular values (the square roots of the aigenvalues of A^TA). Singular value decomposition reduces the dimension of the original matrix A by ordering the eigenvectors and singular values by size and then extracting the largest ones. Then the most important vectors and eigenvalues can be used to construct an approximation of the original matrix.

$$\text{SVD}(A) = U\Sigma V^T \tag{1}$$

4.3 Confusion Matrix

A confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm. Each row of the matrix represents the actual labels while each column represents the predicted actual labels (It can also be the other way around depending on how it is set up). It is a special kind of contingency table, with two dimensions (actual, predicted), and identical sets of classes in both dimensions. Confusion matrices are typically used in supervised learning scenarios which means that the labels are known prior to prediction. The confusion matrix will be used after applying each method of classification. Every index along the top left to bottom right diagonals are the correctly predicted labels and everything else are incorrectly predicted labels.

5. Process and Results

5.1 Visualization

The first step taken will be to visualize what the data actually looks like. Since the data set contains so many images, it will suffice to view only the first 16 images of the training pattern data set. To do this, first extract the first 16 columns from the training pattern matrix. After doing so, rescale each of these 16 columns to become 16x16 matrices and transpose them. Next, display each of the 16 images in one figure using the imagesc() and subplot() functions from MATLAB. Figure 1 displays the first 16 images from the training pattern data set.

The next step in the visualization process will be to view what the average digit looks like in the training pattern data set. To do this, first pool together each unique digit D_i, i = 0, 1, ..., 9 into 10 matrices. Then the average pattern for each D_i is computed by summing all the patterns and then dividing by the number of patterns in each unique digit's matrix. The result will be a 256x10 matrix where each column represents the corresponding average digit's pattern M_i, i = 0, 1, ..., 9. Lastly, each M_i is plotted into one figure using the imagesc() and subplot() functions from MATLAB. Figure 2 displays what the average training digits look like.

5.2 Prediction Using Centroids

Prediction using centroids involves comparing the testing patterns to the patterns of M_i which has been calculated previously. The comparison is done by subtracting each M_i from each testing digit pattern. After this is done, the euclidean distance is then calculated. The result of the prior steps will be a 4649x10 matrix where each row index represents a testing digit and each column index represents a label. The predicted label is then calculated looking at the minimum value of each row. Once the minimum value of each row is found, its predicted label is the value of the column index from which the minimum value is found. The values predicted are shown in table 1.

The next step after predicting each label using centroids is to calculate the accuracy of the model. To do this, a confusion matrix is formed for the predicted and actual values. The confusion matrix is calculated by counting the number of predictions for each digit and referencing it to the actual test digit labels. The rows of the confusion matrix are the actual values and the columns are the predicted values. Table 2 displays the confusion matrix for the centroid classification method.

The accuracy of the centroid algorithm can be calculated with the confusion. The total number of accurately predicted labels can be obtained by summing the values from the top left diagonal to bottom right diagonal elements of the confusion matrix. (Note that confusion matrix is denoted as CM).

$$\text{Accurately Predicted Labels} = \sum_{i=1}^{10} CM_{i,i} \tag{1}$$

$$= 656 + 644 + ... + 314$$

$$= 3936$$

The result is 3936 accurately predicted labels. The accuracy rate can then be obtained by dividing the number of accurately predicted labels by the total number of labels.

$$\text{Accuracy Rate} = \frac{3936}{4649} = 0.8466 \tag{2}$$

The accuracy rate of the centroid classification algorithm is 84.66%.

5.3 Prediction Using Singular Value Decomposition

Similar to the centroids classification method, singular value decomposition also involves pooling together all the unique digits D_i, i = 0, 1, ..., 9. For this particular problem, it was decided that the first 17 left singular vectors will be used (Varying the number of left singular vectors will also vary the accuracy of the model). Therefore for each pooled D_i, the first 17 left singular vectors are computed using the svds() function. Afterwards, the expansion coefficients of each testing digit pattern is calculated using the 17 singular vectors of each training digit pattern. In other words, an approximation of the test digits patterns is calculated using the 17 left singular vectors. Next, the 2-norm of the residual error is calculated for each test digit and the results are recorded into a 10x4649 matrix. Then, for each column, the row index of the lowest value is the predicted label. The values predicted are shown in table 3.

The steps to calculating the accuracy of the singular value decomposition model is the same as the steps to calculating the accuracy of the centroids model. Similar to the prior confusion matrix, the rows are the actual labels and the columns are the predicted labels. A confusion matrix is formed by comparing the predicted values to the actual values. The confusion matrix results are shown in table 4.

The total number of accurately predicted labels and accuracy rate are calculated using the same logic as before.

$$\text{Accurately Predicted Labels} = \sum_{i=1}^{10} CM_{i,i} \tag{1}$$
$$= 772 + 646 + ... + 388$$
$$= 4492$$

$$\text{Accuracy Rate} = \frac{4492}{4649} = 0.9662 \tag{2}$$

The accuracy rate of the singular value decomposition algorithm is 96.62%.

6. Summary and Conclusion

One of the main goals of this project is to compare the two classification models. An intuitive way to compare the two is to look at the overall classification accuracy. Based on the results as described previously, the singular value decomposition method is clearly better because it is almost 12% more accurate than the centroid method.

Another way to compare the two models is to look at how accurately the model predicted each digit's label based on the pattern of the image. This accuracy can be calculated by dividing the i^{th} diagonal element of the confusion matrix by the sum of the i^{th} row. The results for each model are displayed in tables 5 and 6. From the results shown in table 5, the centroid method predicted digits 1 the best with an accuracy rate of 99.54 and it predicted digits 5 the worst with an accuracy rate of 76.35. From the results shown in table 6, the singular value decomposition method predicted digits 1 the best with an accuracy rate of 99.85 and digits 8 the worst with an accuracy rate of 93.35. In addition, the singular value decomposition method classifies every digit with an accuracy of at least 90% while the centroid method accuracy rates vary from 70% to 90%.

The differences in the accuracy rates of the models can be explained by the process each model takes. The centroid model looks at the average image for each pooled digit. One problem with this is that since it pools together all the images and finds the average, there may be some overlap between images that look similar to each other. For example, look at the average images for digits 5 and 6 in figure 2. The bright pixels are generally located in similar areas. This problem could later cause errors because this classification algorithm does not take into account which pixels are more important for each digit. The singular value decomposition method on the other hand creates approximations of images based on a specified amount of singular values. In other words, the singular value decomposition method looks for the most common pixels used in each digit and uses that information to predict the label of the pattern.

From the results, it can be concluded that the singular value decomposition model is strictly better than the centroid model but this is not always the case. As mentioned before, it was decided to use the first 17 singular values for classification. Varying this number can also vary the results of the accuracy rate for the model. Generally using less values will result in lower accuracy rates but using too many values can overfit the model to predict the training images rather than the testing images.

7. Tables

Table 1: Centroid Prediction Results

Label	Counts	Percentage of Total
0	720	15.4872
1	721	15.5087
2	397	8.5395
3	419	9.0127
4	474	10.1957
5	325	6.9908
6	449	9.6580
7	391	8.4104
8	331	7.1198
9	422	9.0772

Table 2: Centroid Prediction Confusion Matrix

	0	1	2	3	4	5	6	7	8	9
0	656	1	3	4	10	19	73	2	17	1
1	0	644	0	1	0	0	1	0	1	0
2	14	4	362	13	25	5	4	9	18	0
3	1	3	4	368	1	17	0	3	14	7
4	3	16	6	0	363	1	8	1	5	40
5	13	3	3	20	14	271	9	0	16	6
6	23	11	13	0	9	3	354	0	1	0
7	0	5	1	0	7	1	0	351	3	34
8	9	19	5	12	6	6	0	1	253	20
9	1	15	0	1	39	2	0	24	3	314

Table 3: SVD Prediction Results

Label	Counts	Percentage of Total
0	788	16.9499
1	683	14.6913
2	438	9.4214
3	421	9.0557
4	432	9.2923
5	351	7.5500
6	410	8.8191
7	400	8.6040
8	320	6.8832
9	406	8.7331

Table 4: SVD Prediction Confusion Matrix

	0	1	2	3	4	5	6	7	8	9
0	772	2	1	3	1	1	2	1	3	0
1	0	646	0	0	0	0	0	0	0	1
2	3	6	431	6	0	3	1	2	2	0
3	1	1	4	401	0	7	0	0	4	0
4	2	8	1	0	424	1	1	5	0	1
5	2	0	0	5	2	335	7	1	1	2
6	6	4	0	0	2	3	399	0	0	0
7	0	2	0	0	2	0	0	387	0	11
8	2	9	1	5	1	1	0	0	309	3
9	0	5	0	1	0	0	0	4	1	388

Table 5: Centroid Accuracy for Each Digit

0	1	2	3	4	5	6	7	8	9
83.46	99.54	79.74	88.04	81.94	76.34	85.51	87.31	76.44	78.70

Table 6: SVD Accuracy for Each Digit

0	1	2	3	4	5	6	7	8	9
98.22	99.85	94.93	95.93	95.71	94.37	96.38	96.27	93.35	97.24

8. Figures

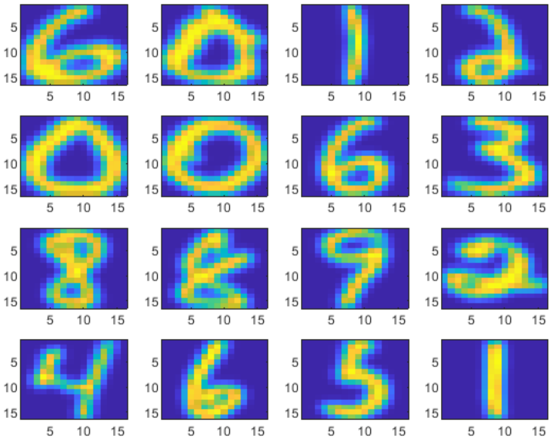

Figure 1: First 16 Training Images

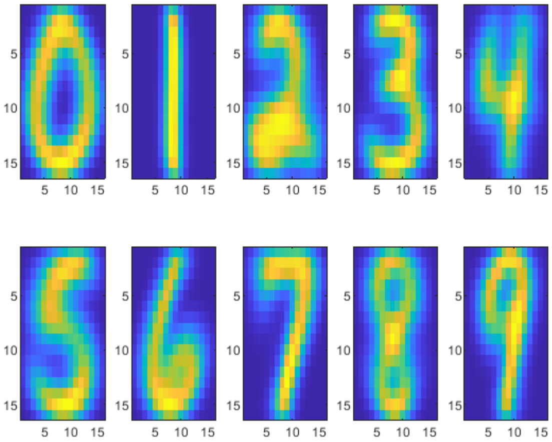

Figure 2: Average 10 Training Images