Hand written digits recognition

EE5907R – Pattern Recognition Project 1

VIGNESH PRAKASAM | A0120541Y | AY13-14- Semester 2

Abstract:

In this project two classification algorithms has been implemented to classify the hand written digits. MNIST database (0 to 9 digits) has been provided and it is used for training the classifier and testing set from the same database is used for testing the classifier designed. PCA (Principal component analysis) is used for the feature extraction and dimensionality reduction. Two Classifiers are (i) Bayesian classification – Self implemented Bayes Classifier produces an error rate of 14.09% for given 10 classes (ii) K-NN (k-Nearest neighbour) Classification produces an error rate of 5%. Better classification was possible with K-NN classification, best choice of K being three. Confusion matrix (10x10) has also been implemented for both the classifiers for greater understanding of the classification. A simple study on how the accuracy is increased and the error rate is decreased when the training samples increases is also made to scope.

Introduction:

The MNIST database of handwritten digits has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centred in a fixed-size image. It is a good database for learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting. These images are available in Matlab format as 'mnist_sub.mat'. 800 images of each digit is provided as the training set and 200 images of each digit is provided as the test set.

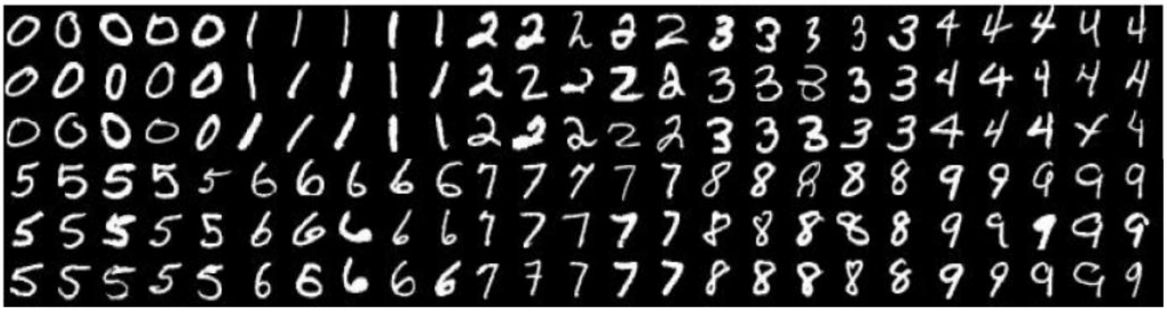

Fig 1.

For further increasing the classification accuracy more images are provided for training. It was also provided in the Matlab format as 'mnist_sub_more.mat'. In this database extra 200 images has been added to the existing database for reducing the classification error rate. So totally 1000 images for each training data has been given. Based on this increased training set an improved classifier is designed. Feature extraction from the images is performed using the

Principal component analysis algorithm which makes the images to a reduced dimensions. Two classifiers are run over the feature extracted vector or the reduced dimension vector for classification. They are Bayesian classifier and K- Nearest neighbour classifier.

Feature Extraction:

Principal component analysis (PCA) is a statistical procedure that uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to (i.e., uncorrelated with) the preceding components. Principal components are guaranteed to be independent if the data set is jointly normally distributed. PCA is sensitive to the relative scaling of the original variables

In this case, PCA (Principal component analysis) has been used for an effective feature extraction. All the training images vector are taken in a single vector with each 800 vectors of the matrix representing a class. Totally there are 10 classes (0,1,2,3….,9) . PCA algorithm is used on this total description vector to get the necessary coefficient and score for further classification of the data.

For better efficiency first 32 components of PCA score has been selected and the process follows on it. 32 was seemed to be a best fit of principal components that contribute for better classification accuracy compared to other values.

$$y1 = W^T(x - m)$$

y1,W,x,m are score, coefficient, description vector, mean respectively.

With the above coeff(W) we find the score on test vector. Shown as below with 't' representing the test vector.

$$y2 = W^T(t - m)$$

Classification:

Two classification algorithm has been proposed to solve the classification problem. They are Bayesian classification and Nearest neighbour classification.

i. Bayes classification:

Bayes classifier is popular in pattern recognition because it is an optimal classifier. It is possible to show that the resultant classification minimises the average probability of error. Bayes classifier is based on the assumption that information about the classes in the form of prior probabilities and distribution of patterns in class are known. It employs the posterior probabilities to assign the class label to a test pattern; a pattern is assigned the label of the class that has the maximum posterior probability into posterior probability based on the pattern to be classified, using likelihood values.

Classification method:

Once the feature extracted vector is available each images of the training vectors are subjected to the bayes algorithm to the end classification. 32 principal components are obtained so the calculations of mean and covariance of each class is calculated.

In this classification case I consider ‘all the features are independent and have the same variance for all classes’. So the density function reduces to the below form.

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^t \ \epsilon_i^{-1} \ (x-\mu_i) - \frac{1}{2}\ln|\epsilon_i| + \ln P\,(\omega_i)$$

$$\downarrow$$

$$g_i(x) = -\frac{1}{2\sigma^2}(x-\mu_i)^t \ (x-\mu_i) + \ln P\,(\omega_i)$$

Labelling the training and test data is done accordingly in prior before finding the log likelihood with respect to each class. Largest log likelihood is selected and compared with test label to classify which class it belongs to.

Self coded bayes classifier was able to provide, 85.95% accuracy in classification with an error rate of 14.05%. Below (Fig 2.) is the study of accuracy obtained with respect to the reduced dimensions.

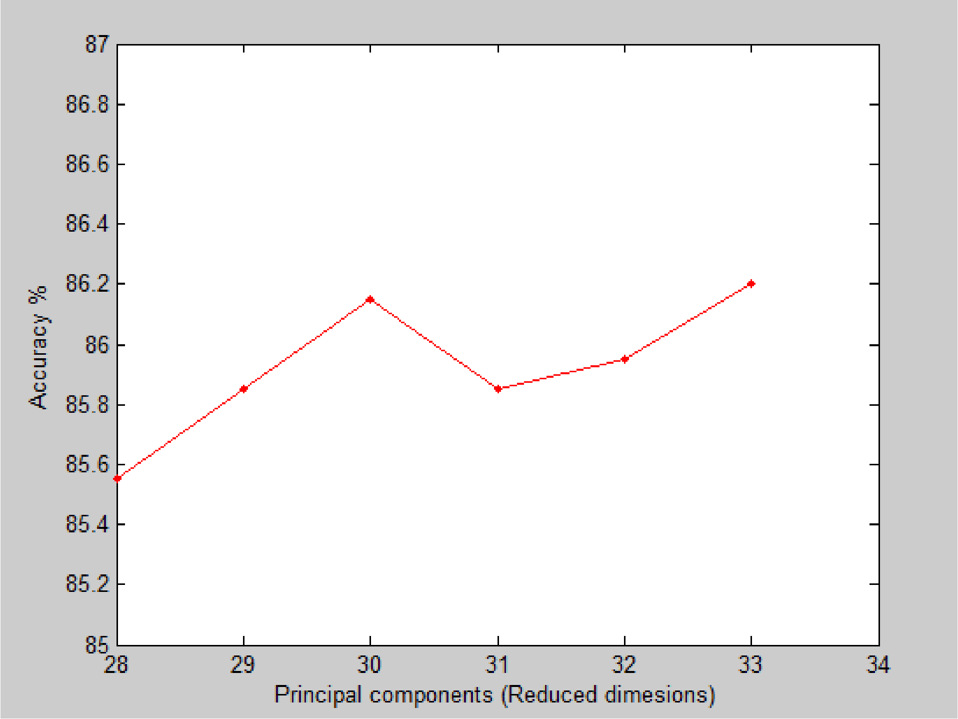

Fig 2.

Confusion matrix:

In the field of machine learning, a confusion matrix, also known as a contingency table or an error matrix , is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mis labeling one as another).

Confusion matrix has been formulated for the Bayes classifier and results of a self coded confusion matrix has been shown below (Table 1.).

	Class0	Class1	Class2	Class3	Class4	Class5	Class6	Class7	Class8	Class9
Class0	193	0	2	0	0	3	2	0	0	0
Class1	0	189	3	2	0	3	0	0	3	0
Class2	3	2	164	6	2	3	5	2	12	1
Class3	1	0	2	169	0	13	1	2	8	4
Class4	1	3	4	0	162	2	3	0	2	23
Class5	1	0	2	9	3	174	2	3	3	3
Class6	0	3	1	0	2	13	181	0	0	0
Class7	0	7	4	1	4	1	3	166	4	10
Class8	2	1	6	8	3	14	2	2	159	3
Class9	2	2	1	0	19	6	0	5	3	162

Table 1.

ii. K-Nearest Neighbourhood classification:

In pattern recognition, the k-Nearest Neighbours algorithm (or k-NN for short) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space.

In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours (k is a positive integer, typically small).

If k = 1, then the object is simply assigned to the class of that single nearest neighbour.

So a self coded K-NN classifier was implemented and verified with the test set. Reduced dimension vector was used to perform the algorithm with different choice of ‘K’. Various study on each K specified the best choice of K would be 3 because it could result to a good accuracy.

Method:

1. Finding the nearest neighbour

2. Finding the K nearest neighbour

3. Get the K least entries

4. Median of all i.e finding the K min

5. Matching with the labels for accuracy calculation.

With the above implementation of K-NN, Error rate was 5.85% and accuracy accounts to 94.15%.

A graphical representation of the reduced dimension and the accuracy obtained by the classifier is shown below.

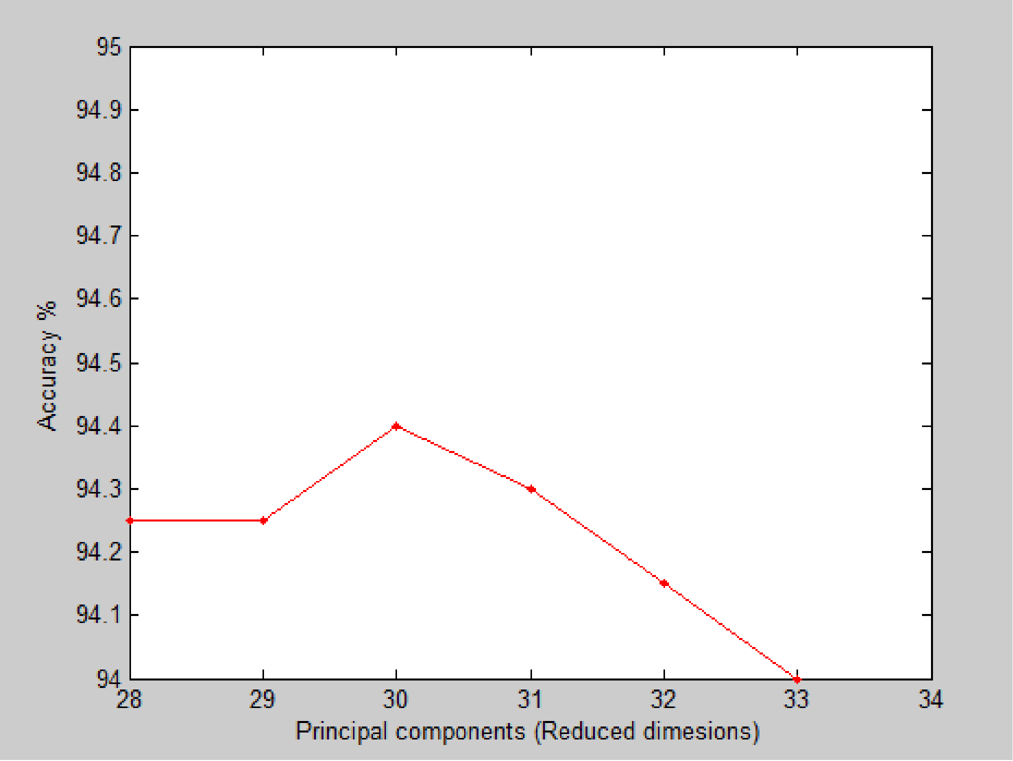

Fig 3.

Confusion Matrix:

Similar to Bayes classifier, K-NN classification error rate is also described with a confusion matrix for an easy representation and understanding which digits are classified as wrong digits most number of times. K-NN classification had better efficiency compared to the Bayes classification.

Handwritten digit '9' is always misinterpreted. So a better efficiency can be achieved by a better feature extraction. Below is the confusion matrix for a reduced dimension of 32. (Table 2.)

	Class0	Class1	Class2	Class3	Class4	Class5	Class6	Class7	Class8	Class9
Class0	200	0	0	0	0	0	0	0	0	0
Class1	0	198	2	0	0	0	0	0	0	0
Class2	0	1	192	0	0	1	2	2	2	0
Class3	0	0	1	189	0	4	0	2	3	1
Class4	1	3	1	0	184	0	2	0	1	8
Class5	0	0	1	2	0	196	0	0	0	1
Class6	0	2	0	0	0	2	196	0	0	0
Class7	0	3	2	0	0	0	0	194	0	1
Class8	1	0	3	2	0	3	2	1	187	1
Class9	2	2	0	0	3	3	0	1	1	188

Table 2.

Increasing the samples:

For better classification purposes additional 200 images have been given for training. Both the classifiers are made to train over these additional images. It is present in the matlab format 'mnist_sub_more.mat' file. Now the error rate of the classifiers seemed to be reduced to an extent over the testing of classifiers.

Bayes classifier had an accuracy of 86.30% and K-NN classifier had an accuracy of 94.60%

For Bayes: Efficiency increased by 0.35%

For KNN: Efficiency increased by 0.45%

Concise classification tabulation is mentioned below (Table 3.).

Experimental Results:

	Initial Training set		Additional Training set		Performance improvement	
	Error rate	Efficiency	Error rate	Efficiency	Error rate ↓	Efficiency ↑
Bayes Classification	14.05%	85.95%	13.70%	86.30%	0.35%	0.35%
K-NN Classification	5.85%	94.15%	5.40%	94.60%	0.45%	0.45%

Table 3.

Conclusion:

Based on the intensity and shape of the images the feature extraction algorithm (PCA) gets the principal components for the classification based on them. Best accuracy of 94.60% was available from the K-NN classifier and a better classification accuracy of 85.95% from Bayes classifier. A confusion matrix was generated for better error rate understanding. When the training data set was increased, the efficiency of both the classifiers (Bayes and K-NN) increased by 0.35% and 0.45% respectively. So from this study it is evident that the accuracy of classification is proportional to number of samples available for training.

References:

1. Y. LeCun,L. Jackel et al. “Comparison of Learning algorithms for Hand written digit recognition”.
2. “Pattern classification” by Duda, Hart, Stork,2001.
3. Bottou, Corrinna et. Al. “Comparison of classifier methods: A Case study in Handwritten digit recognition”.
4. Kazuki, Hiroshi et al. “Handwritten digit recognition : investigation of normalization and feature extraction techniques”.