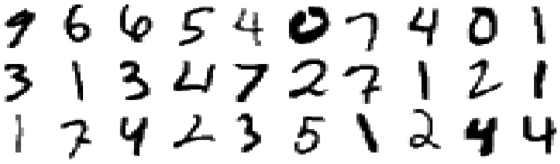
Handwritten Digit Recognition



Project Milestone

Team Members

Akshara Boppidi

Anirudh Nagulapalli

Pooja Jadhav Eshwarlal

December 12, 2016

I. Abstract

The goal of the project is to take an image of a handwritten digit and determine what the digit is. This project is chosen from Kaggle competitions. The data for this competition is taken from the MNIST dataset (Modified National Institute of Standards and Technology) which consists of training set of 60,000 examples and a test set of 10,000 examples. This data is provided by Kaggle as train.csv and test.csv.

These data files contain grayscale images of hand-drawn digits starting from zero to nine. There are 10 digits to predict. Each image contains 784 pixels in total. It is scaled into 28*28 pixels height by width representing the pixels brightness or darkness with a number rating. The rating given to the pixel is between 0 and 255, inclusive. The train.csv contains 785 columns. The first one is the 'label' and is the digit drawn by the user. The rest of the column contains the pixel value of the surrounding pixels.

II. Introduction

Digit recognition is a good problem to learn about machine learning. Some applications for digit recognition are online handwriting recognition on devices and numeric entries in forms filled by hand, investigation, recognition of zip code by postal services. There are different approaches to achieve higher performance for digit recognition, based on support vector machines, neural networks and nearest neighbor methods.

While performing handwritten digit recognition we come across many challenges, one being the digits written are not always the same, they may vary in size and thickness. The variety in handwriting influences the appearance and formation of digits. Our goal is to recognize the handwritten digits (0-9) from the dataset of images.

III. BACKGROUND

Pattern recognition is another well-established area of study that is known through years especially in the field of digit recognition which is considered as an obvious challenge and significant contributors to digit recognition. Recognition of handwritten digits is a big challenge since 1980's especially in the old manuscripts and documents [1].

Digit recognition with use of classifier gives a greater performance and use such as recognition of zip codes. The most general problem in digit recognition is to predict the similar digits like (1, 7), (5, 6), (3, 8). Also, users write same digits in different ways. This influences the appearance and shaping of digits. We can perform digit recognition by implementing different kernel functions such as polynomial kernel function, radial basis function and sigmoid function. Numerous experiments are also being conducted in high dimensionality and small sample size to analyze the superior classification performance of SVM.

IV. METHODOLOGY

4.1-Support Vector Machines:

Support Vector Machines (SVM) is supervised learning method used for analysis of data either by classification or regression. SVM's are widely being used in image detection and novelty detection. SVM's generate input-output mapping functions from labeled training dataset. In general kernel functions are used to transform the input data to a higher dimensional space so that the original data is easily separable as shown in figure 1. The main aim of SVM is to determine a maximum margin hyperplane.

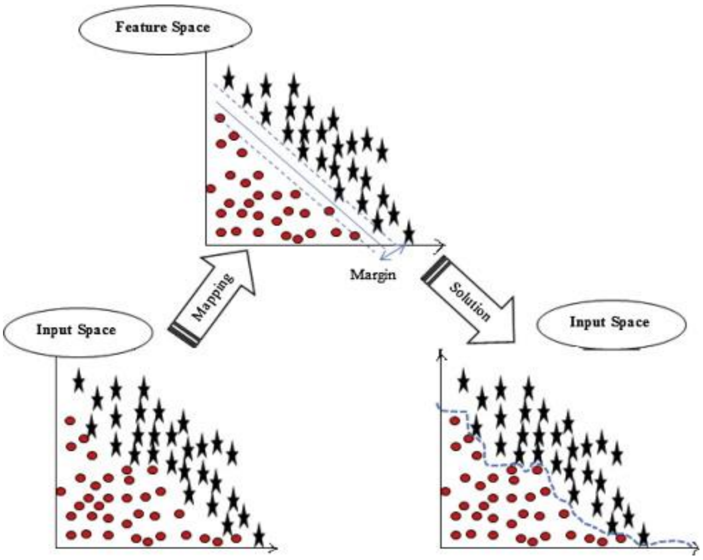


Figure1: Flow of Data in SVM

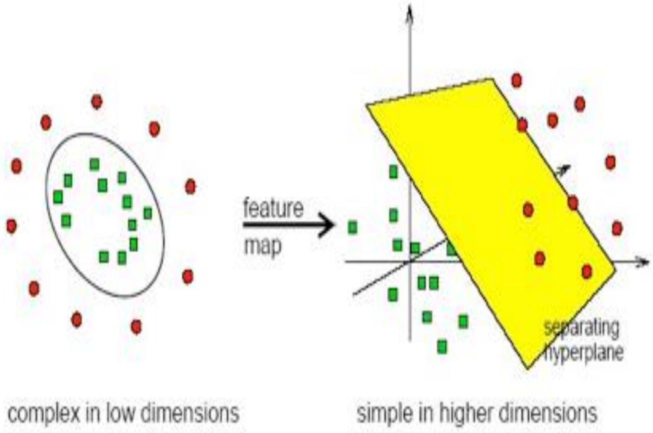


Figure 2: Mapping input data to a higher dimension

4.2-Random Forest:

Random forest algorithm has been extremely useful as a general-purpose classification and regression method [3]. This algorithm uses an ensemble of large number of decision trees whose output is equal to the mode of singular trees decision. It runs effectively on large databases. Collections of decision trees which all together produce predictions are called Random Forests. We can also use this method for estimating any missing data and maintain accuracy when large proportion of data is missing. Ensembles are basically a divide and conquer approach which is used to obtain higher performance.

4.2.1-Random Forest Algorithm:

Each node of a tree is associated with a hyper rectangular cell. The root of the tree is called ‘X’ and at each step of tree construction, the node is divided into two terminal nodes (or leaves). Decision trees are a popular method for solving many machine learning problems. The main function of Random forests are a way of averaging multiple deep decision trees which are trained on different parts of same training set, with the ultimate goal of reducing the variance. This results in some increase in bias and some loss of interpretability but greatly boosts the performance.

4.3-Principal Component Analysis:

We use PCA to reduce the dimensionality of data, based on the idea that data may lie close to a low dimensional hyperplane. Regression finds a line that minimizes the vertical distance between a data point and the line.

Since PCA uses the eigenvectors of the covariance matrix, it is able to find the independent axes of the data under the unimodal Gaussian assumption.

Based on the information in the image PCA extracts Eigen based digits. The main limitation of PCA is that it does not consider class separability since it does not take into account the class label of the feature vector. The features such as grayscale value and thickness of the digits provide more information.

V. EXPERIMENTS

5.1-Load the data:

We are getting data from Kaggle. Training set has 42000 examples and 785 features. Test set has 28000 and 784 features.

5.2-The database:

The train.csv file has 42k examples, where the first column is digit labels and remaining 784 columns are pixel intensity values that are from 0 to 255 and the test.csv contains 28k rows of pixel color values which are classified as digits. The training set has shape of (42k, 784), each row is 28*28 pixels image. Which will be reshaped to have the training set of each row 28*28 matrix of pixel values and same is done with testing dataset.

5.3-Train SVM classifier:

Here we perform ‘fit’ operation between training set of images and test set of what those images mean. We now try to predict the validation data using score, which tells how accurate our model is, like what percentage of getting our handwriting correct. For example, if score = 0.8312, our algorithm will get 83 images right for every 100 images in the dataset.

We have applied cross validation for the data, as the dataset is large we were obtaining a memory error after running the code for a long time. So, we performed Support vector classification for given dataset, we obtained an accuracy of 1. Which, can be an over fit.

5.4-With Random Forest classifier:

In random forest classifier, we fit the data and predict class for training set. We have given different number of trees in the forest to obtain a predictive accuracy. For n_estimators = 2, we obtain an accuracy rate of 0.926 and if we increase n_estimators = 10, we get an accuracy on training set as 0.96 and for n_estimators = 100, accuracy is 1 and then we save the predicted results into a .csv file which consists of imageid and label. Here, label is the predicted digit by the random forest.

5.5-Classification in PCA:

The unknown digit image is projected on principal components. Then the image is compared to all the images in the database. After comparison, the closest match is taken as the identified digit. To improve the accuracy of PCA by implementing the k nearest neighbors approach and perform this identification process for "N" number of matches and identify the majority of the these matches to be the recognized digit.

VI. Analysis

We have loaded the csv files as input data and obtained the number of rows and columns from the files. We are reshaping the pixel size of the images and predicting the accuracy by training the dataset using cross-validation by learning techniques like Support vector machines, random forest generator and principal component analysis. And we save the results to .csv file.

VII. Conclusion:

We have reached the computer to the human's brain by the importance of isolated digit recognition for different applications. This project deals with recognition of handwritten digits by applying supervised learning techniques such as Support Vector Machine, Random Forest and Principal Component Analysis. This recognition starts with acquiring the image, reshaping the dataset and we obtain predictive accuracy by choosing an appropriate classifier technique.

At the end of this experiment we will determine the accuracy rate that the current feature extraction. We found that SVM was proving to be an overfit with an accuracy of 1. With Random Forest we obtained an accuracy of 0.926. The time taken by random classifier was better compared to SVM. We are trying to reduce the dimensionality to observe the performance of given dataset using PCA.

VIII.Acknowledgement:

Special thanks and recognition goes to our advisor, Professor. Dr. Pablo Rivas who guided us through this study, inspired and motivated us. Also, we would like to thank the Marist College for giving us an opportunity to get introduced to this course work.

IX. References

1. International Conference on Advances in Pattern Recognition, 2009. ICAPR'09., 2009, pp. 391–394.
2. G. Biau and E. Scornet, "A random forest guided tour," *TEST*, vol. 25, no. 2, pp. 197–227, Apr. 2016.
3. X. Jiang, H. Pham, and Q. Xu, "Handwritten Digit Recognition via Unsupervised Learning," 2015. [Online]. Available: http://cs229.stanford.edu/proj2015/165_report.pdf.
4. A. Juan, V. Romero, J. A. Sanchez, N. Serrano, A. Toselli, and E. Vidal,

"Handwritten Text Recognition for Ancient Documents," in *JMLR: Workshop on Applications of Pattern Analysis*, Spain, 2010. [Online]. Available: http://jmlr.org/proceedings/papers/v11/juan10a/juan10a.pdf.

5. A. Alaei, U. Pal, and P. Nagabhushan, “Using modified contour features and svm based classifier for the recognition of persian/arabic handwritten numerals,” in IEEE Seventh