Machine Learning Assignment - 2

Darshan Agarwal
201225189

Problem Setting

- Handwritten digit recognition is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch screens and other devices.
- The applications of the handwritten digit recognition include automatically address reading and mail routing, postal mail sorting, bank check processing, etc.

Problem Setting

In this assignment, we will be tweaking the feature representation as well as the classifier to see their impact on performance in solving handwritten digit recognition problems.

Problem

Given an unseen image of a handwritten digit, assign relevant label to the image.

Problem Space

- The problem deals with objects in a **real two-dimensional space** and the mapping from image space to category space has both considerable regularity and considerable complexity.
- This problem is of great interest in the **pattern recognition research community** because of its applicability to many fields towards more convenient input devices and more efficient data organization and processing.
- The problem in the recognition of handwritten digits is that of **within class variance** since a same digit can be written in multiple ways.
- It is seen from the experiments that extraction of features in terms of **direction, local structure and curvature** improves the accuracy of classification.

Dataset

- We have used MNIST and USPS datasets for the assignment.
- MNIST 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 centered in a fixed-size image.
- USPS has a training set of 4649 images and a test set of 4649 examples.

Techniques Used

In this assignment we are doing a comparative study between the following techniques :

1. K Nearest Neighbour
2. SVM
    a. RBF Kernel
    b. Polynomial Kernel
    c. Linear Kernel

K Nearest Neighbor

- In k-Nearest Neighbor classification, the output is a **class membership**. The K is used to signify how many votes are used for decision making.
- An object is classified by a **majority vote of its neighbors**, with the object being assigned to the class most common among its *k* nearest neighbors.
- The optimal value of K to choose is **odd** so that it eliminates a tie between two sets.

SVM

- A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane.
- In other words, given labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples.
- The optimal separating hyperplane ***maximizes*** **the margin** of the training data.
- We have used a multiclass SVM classification (libsvm) in our model with different kernels of
  a. Linear
  b. RBF
  c. Chi Square
  d. Polynomial

SVM

- **RBF Kernel:** The Radial Basis Function kernel on two samples x and x', represented as feature vectors in some *input space*, is defined as

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

- **Linear Kernel:** For all objects *u,v X,* where X is a set of training examples.

$$K(u,v) = f(u).f(v) = f_1(u)f_1(v) + f_2(u)f_2(v)$$

- **Polynomial Kernel:** For degree-*d* polynomials, the polynomial kernel is defined as where *x* and *y* are vectors in the *input space*, i.e. vectors of features computed from training or test samples and $c \geq 0$ is a free parameter trading off the influence of higher-order versus lower-order terms in the polynomial.

$$K(x, y) = (x^\top y + c)^d$$

STAGE 1

- In this stage, the raw data samples in original dimensions were taken from both the datasets and classification performance was obtained for the explained classifiers.
- Confusion Matrices and Accuracy can be seen in the report.

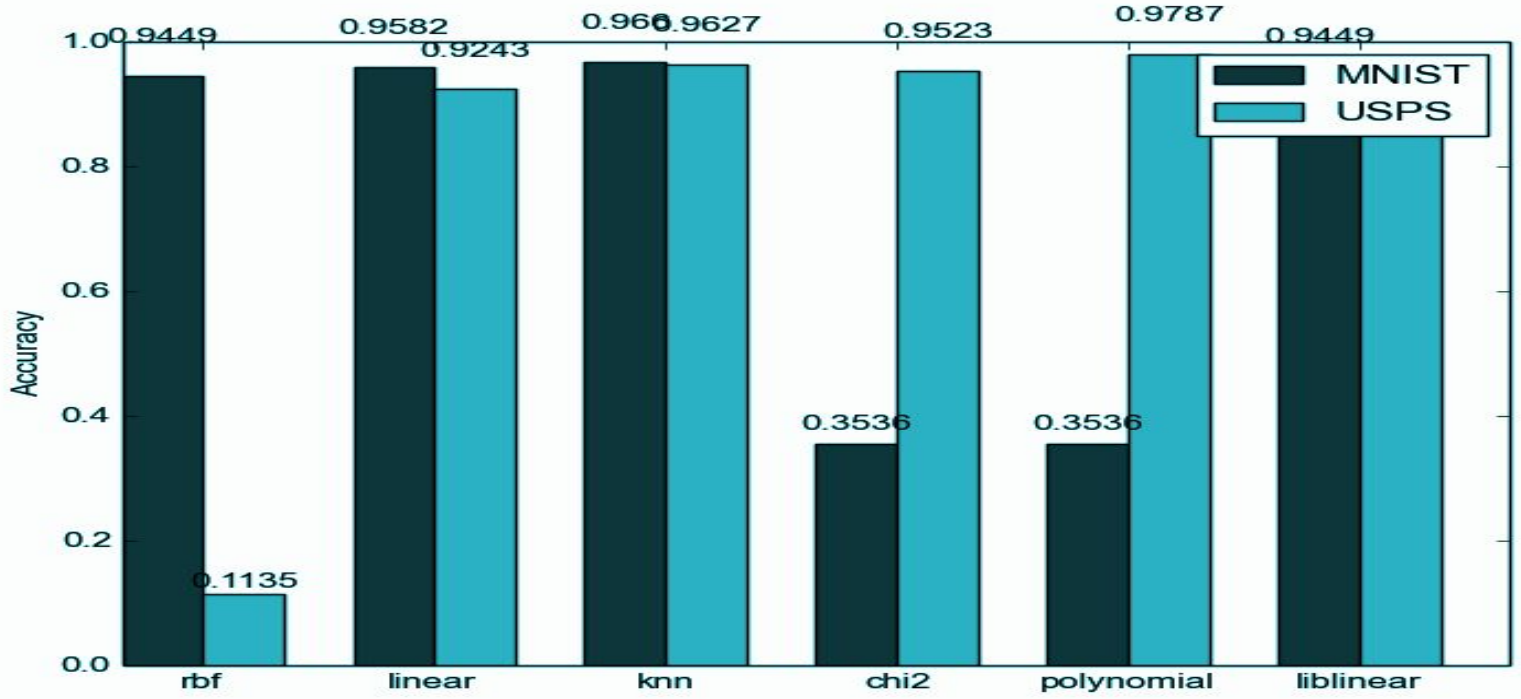
MNIST

- In the case of MNIST data, it is observed that a polynomial kernel gave the best results. It is followed by KNN classifier. SVM (libsvm) using a rbf kernel gave extremely bad results and a linear SVM using liblinear gave slightly lower accuracy. The relatively high accuracy of linear SVMs compared to rbf can say that the classes are more or less linearly separable except for a few outliers in each class, which were taken care of properly by a polynomial kernel.

USPS Data

In the case of USPS data, it is observed that a knn classifier gave the best results. It is followed by SVM (libsvm) using a linear kernel. SVM (libsvm) using rbf kernel and linear SVM using liblinear gave almost the same accuracy and similar number of misclassified samples. Same as USPS it is seen that, linear SVMs has high accuracy, so we can come to a conclusion that the classes are more or less linearly separable except for a few outliers in each class.

MNIST and USPS



Runtimes for Linear SVM

	USPS Training time	MNIST Training time	USPS Testing time	MNIST Testing time
LibLinear	2.25seconds	1min50sec	0.5seconds	2.1seconds
LibSvm	28.0seconds	140min0sec	24seconds	49seconds

- Liblinear was observed to be much faster than libsvm. The main reason behind the different run times for LibLinear and Libsvm is that Libsvm uses a 1 vs 1 strategy while liblinear uses a 1 vs rest strategy.

HyperParameter Tweaking

In both RBF Kernel and Polynomial kernel, the performance was observed to be increasing in case of the c parameter. The performance decreased with gamma in the case of rbf, while there was no such decrease observed in case of polynomial kernel, it had an optimum as gamma=0.5. One thing to note is that, in case of parameter ‘degree’ the performance of polynomial kernel degraded linearly indicating the linear separability of data.

STAGE 2

- Three different representations chosen are : PCA, sphog and FLD.
- Results can be seen in the report.
- The performance of PCA was observed to be very bad with accuracy of only 9.5. The reason for this could be due to the fact that the data was just a collection of pixels and wasn't consisting of any features as such that were extracted from the images. SPHOG gave extremely good results probably because of the fact that it is scale invariant.

STAGE 3

- When tangent distances are used, they lead to more **distinguishable clustering**.
- This could be easily seen in the case of 4's and 9's which were hardly distinguishable with Euclidian distance but formed distinct clusters when tangent distances were used.
- In case of "4" and "9", the sample images with higher x coordinate is clearly different from the image with different label, whereas those having lower or negative x coordinate are more similar with the ones with different labels. Tangent distance maps the digits on a larger scale [-150 150] and the variance among instances of same digits is lesser.
- Although mapping done using **euclidean distance is faster than that using tangential distance.** Isomap is not able to create much clearer regions for all the digits it seems for more clear boundaries our decision region should be of more dimensions.

STAGE 4

The parameters that affect the computational requirements are:

1. Number of samples, their dimensionality and their distribution
2. Number of classes
3. Tolerance (mainly affects memory)
4. Kernel used and its hyperparameters in non linear svm’s like the degree of a polynomial kernel
5. Multiclass strategy (1-v-1 or 1-v-rest)

STAGE 4

- When the following training scheme is used:
  - Train with only 10% of the training data and test on the rest of the training data.
  - If there are any misclassification, add them and retrain.

Result with 10% training scheme

- This training scheme can give a performance close to that training with the full dataset. The support vectors would mostly remain same even if we the samples that aren't support vectors are ignored while training.
- Hence, this training scheme would end up learning the support vectors in a quicker way as they are very few in number and independent of the number of non support vector samples.
- The main advantage is that it saves a lot of time because we are training only on a small portion of the dataset.
- The disadvantage of this scheme is that there may not be enough outlier points to minimize the error in decision boundary margin between classes.
- Also, the distribution of the final data that is considered may be an accurate representation of real world data.

Result with 10% training scheme