SVM Digits Classification

Name: Kunal Lad
UT Eid: KL28697
Email: kunal.lad@utexas.edu

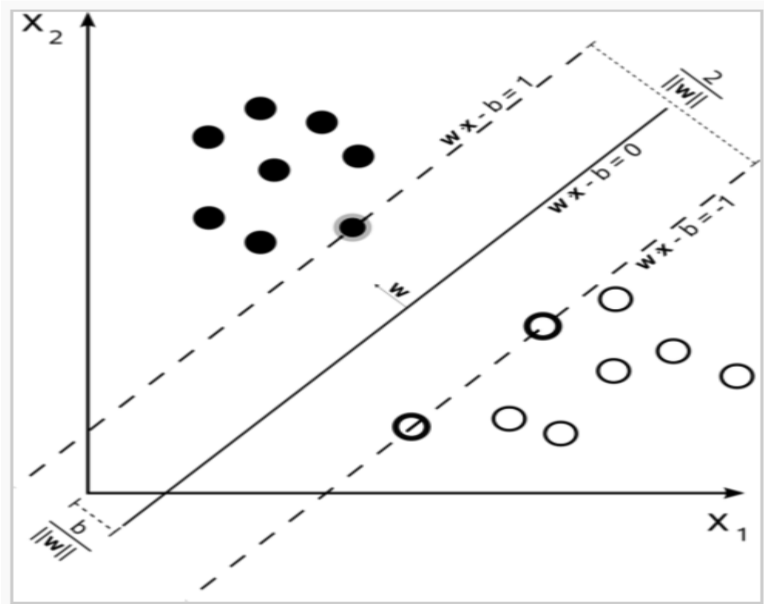
Objective

The goal of this assignment is to use Support Vector Machines for Digits Classification problem on MNIST dataset and investigate how various parameters of SVM models affect the performance.

Introduction

Support Vector Machines are supervised learning algorithms used for classifying linearly separable data. Given training data (xi,yi) for i = 1...N, with xi ∊ Rd and yi ∊ {−1, 1}, learn a classifier f(x) such that yi f(xi) > 0. The goal is to learn maximum margin hyperplane so that the model can generalize well. See fig below (taken from Ref [2])

This is formulated as a quadratic optimization problem with linear constraints. It can be solved more easily by solving its dual problem using standard optimization techniques like Lagrange Multipliers.



Maximum-margin hyperplane and margins for an SVM trained with samples from two classes.

SVM can also be used to classify non-linearly separable data by using kernel trick which projects data to higher dimensional space so that the examples of the separate categories are linearly separable in new space. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

Methodology

Data Details

1. Training Data consists of 60000 images of size 28 x 28
2. Test Data consists of 10000 images (5000 easy, 5000 hard) of size 28 x 28

Features

Feature selection is one of the most important tasks of any Machine Learning based system. In this assignment we investigate SVM models based on 3 different types of features. First we use image pixels as features. Each image is of size 28 x 28, hence we get a 784 dimensional feature vector for each image. We saw in hw1 that most of the information is can be captured by lower dimensional feature space. Thus we use hw1 code to extract 350 dimensional features obtained by PCA. We also use Histogram of Oriented Gradient (HOG) features which capture properties of local neighborhood of pixels. They are much more robust to transformations, illumination etc.

Algorithm

Input: Digit images from MNIST dataset.
Output: Digit Prediction for input image.

1. Extract features from image.
   - Image Features : Use pixels from image as features.
   - PCA : Use code from hw1 to project image features to lower dimensional space.
   - HOG Features: Use computer vision toolbox in matlab to extract HOG features from Image.

2. Train SVM model using features extracted in 1. and labels from MNIST dataset

3. Predict labels on test dataset.

4. Evaluate accuracy on the test set.

Experiments

This section gives details of the experiments performed in order to asses the performance of SVM model with respect various parameters. For all the experiments mentioned in this section we use the following unless specified otherwise:

- Random Sampling using datasample command in MATLAB for sampling training and test data.
- 3000 images for training and 1000 test images (500 hard, 500 easy) for evaluating the accuracy.
- Top 350 eigen vectors for extracting PCA features
- Used **fitcecoc** method in MATLAB for training SVM model with appropriate parameters for each experiment.

E1: Performance variation of models trained with different features with size of training data

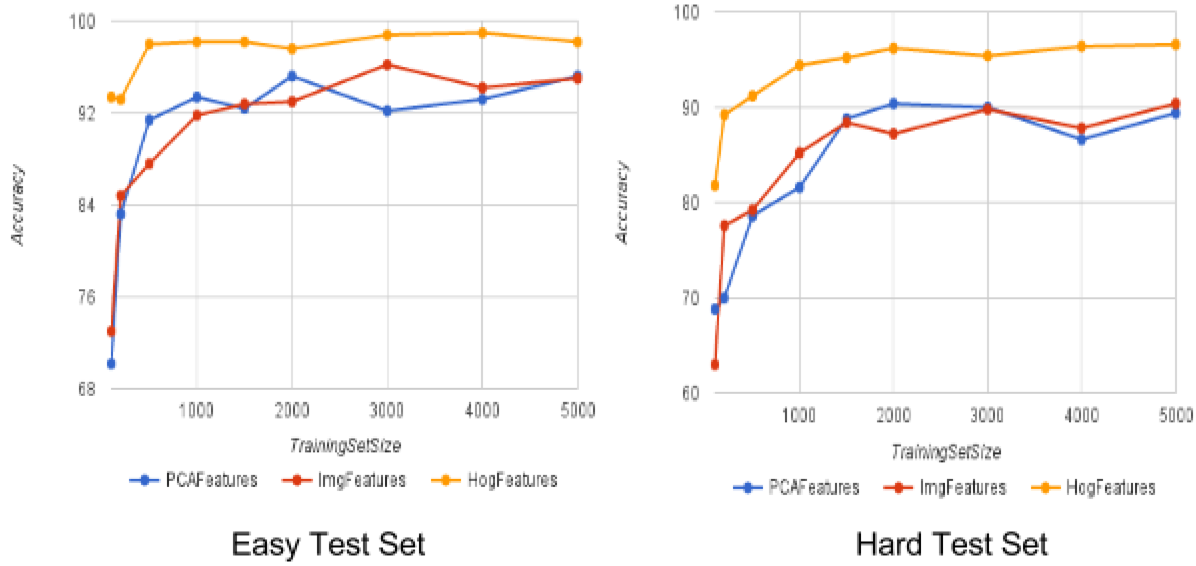


Fig 1 Accuracy vs Training Set Size for PCA, Image and Hog Features on easy test set.

In this experiment we investigate which features are more suitable for this task and how does the model performance vary with size of training data. Plots for both easy and hard test sets in Fig 1 suggest that HOG features are superior achieving ~98% on easy and ~96% on hard test set compared to to PCA and Full Image features which achieve ~93% on easy and ~90% on hard test set. This is expected because HOG features are bigger in size and contain information of local neighborhood of pixels.

As expected we observe that accuracy improves with increase in size of training data upto a point (around 2000) and then stabilizes. We can see that accuracy increases sharply with small increase in data for all feature types. Since HOG features are very informative, they perform well even with small amount of training data. Thus gain in performance is relatively less as compared to other 2 feature types. Finally we can see that performance curve of PCA and full image features is almost similar. This shows that PCA features capture almost all the important information in full image features.

E2: Comparison of different kernels

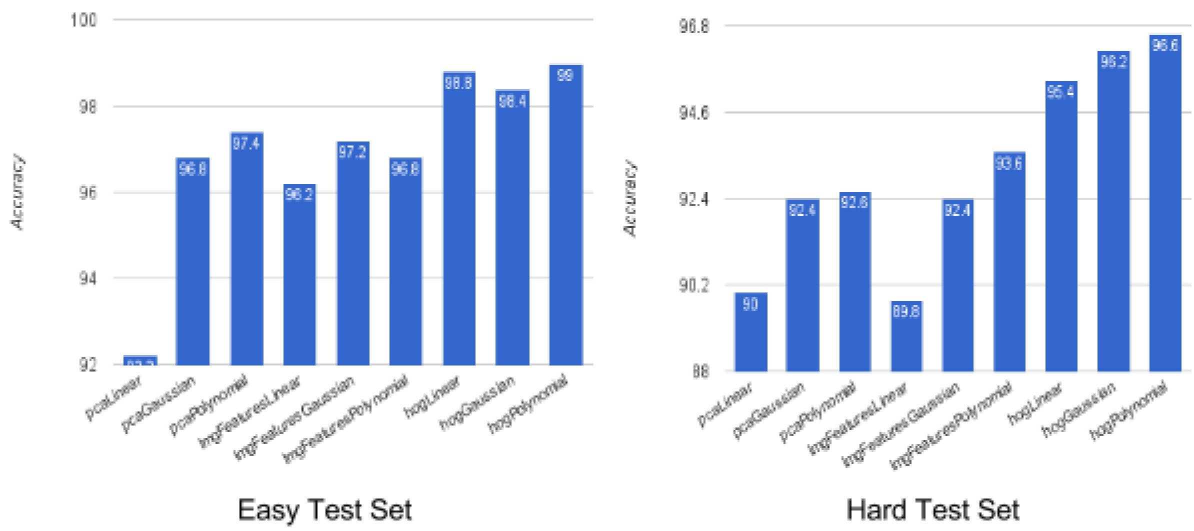


Fig 2 Performance of different kernels with pca, image and hog features

In this experiment we investigate how the kernel affects the performance of SVM model. We experiment with linear, polynomial and gaussian kernels. From plots in Fig 2 we observe that polynomial kernel performs the best with all 3 feature types on both easy and hard test sets. Gaussian kernel performs very close to polynomial kernel whereas linear kernel performs significantly worse than other 2 kernels all feature types. One interesting thing to note is that linear kernel with HOG features outperforms gaussian and polynomial kernel with PCA and image features. This again shows that HOG features are much more powerful than other 2 features.

E3: Comparison of OneVsAll and OneVsOne SVM models

In this experiment we compare the performance of multi class SVM techniques OneVsAll and OneVsOne. Classification of new instances for the one-versus-all case is done by a winner-takes-all strategy, in which the classifier with the highest output function assigns the class. For the one-versus-one approach, classification is done by a max-wins voting strategy, where class votes are aggregated over all classifiers [1]

From plots in fig 3 we can see that OneVsAll performs better than OneVsOne model for all 3 feature types and on both test sets. However the difference is very small for HOG features based model (95.4% vs 95.2% on hard test set) and huge for PCA features based model (90% vs 87.3% on hard test set). Again the reason for this is the fact that HOG features are very informative and hence model parameters don't affect the performance as much as they do for weaker PCA based features. One more thing which we found during experiments was that OneVsOne model trains much larger number of classifiers ((n-1) * n / 2) than OneVsAll (n) for n classes and hence take more time.

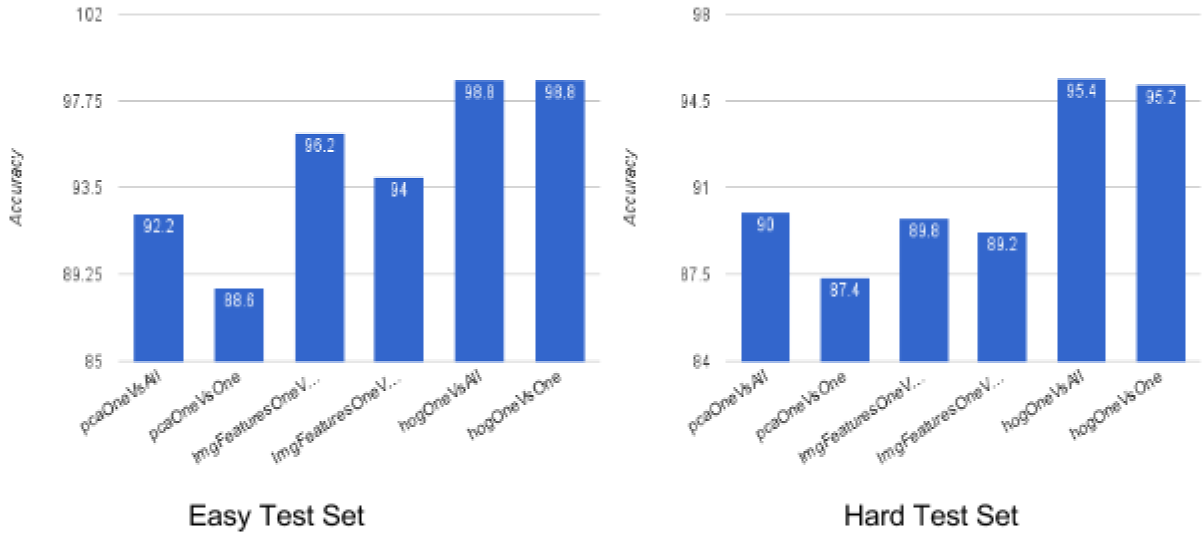


Fig 3 Performance of OneVsAll and OneVsOne models with different features

E4: Performance variation with box constraint parameter

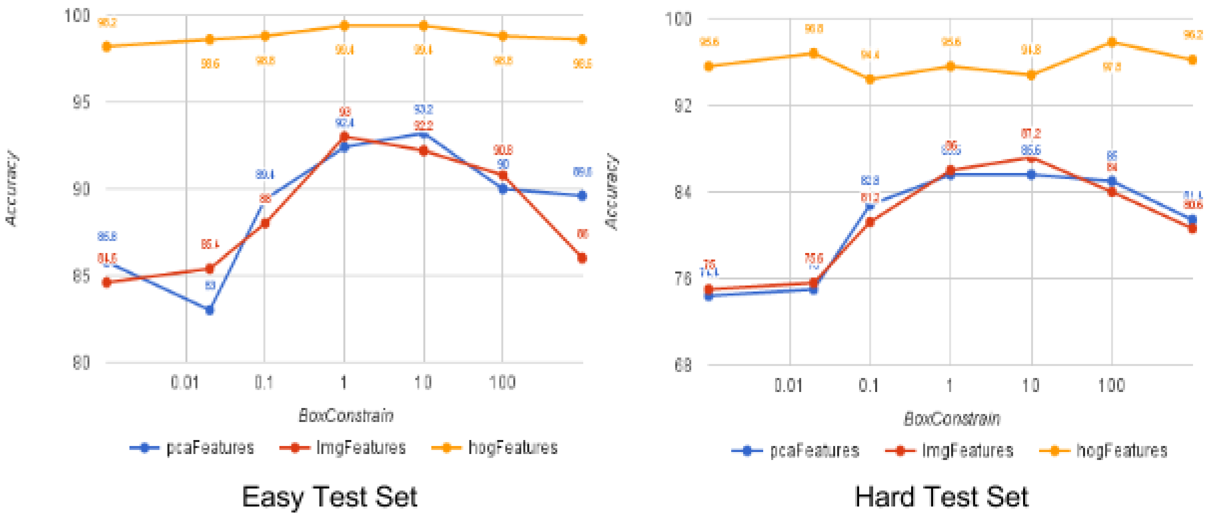


Fig 4 Performance variation with box constraint

In this experiment we investigate the performance variation with respect to box-constraint parameter which controls the penalty for margin-violations. This parameter allows the model to have a soft margin so that it can handle small number of noisy data points which might lie on other side of hyperplane. If the value of box constraint is set to infinity then it becomes a hard margin model. From plots in Fig 4 we can see that we get best performance around the value of 1 which is the default value in MATLAB. The performance is low for small values, increases gradually till box constraint = 1 and then starts decreasing again. As with other experiments the variation is less for HOG features model as compared to the other 2 models.

NOTE : Detailed stats of all the experiments can be found here

Conclusions

Performance of multiclass SVM model depends on the number of parameters like number of training examples, selection of kernel, box constraint, model type (OneVsAll or OneVsOne) etc. However as all experiments suggest effect of these parameters is larger for models based on weaker features like PCA or image features but very less for HOG feature based model. This shows that identifying correct features for given problem is the most important thing for training good models.

References

[1] Pattern Recognition And Machine Learning by Christopher Bishop
[2] Wikipedia