GENDER CLASSIFICATION USING SUPPORT VECTOR MACHINES

Raghuraman Gopalan

raghuram@umd.edu

ENEE 633: Statistical and Neural Pattern Recognition

Instructor: Prof. Rama Chellappa

Course Project 2

1. INTRODUCTION

In this project we consider the problem of gender classification using support vector machines. The dataset consists of 270 images which has both male and female faces. So, this is a two class problem and the objective is to find a separating hyperplane that properly classifies the faces as male or female [1], [2]. Some images of the dataset are used to train the SVM. Since any prior information about the dataset is not known, the samples may or may not be separable by a linear classifier. So, the samples are transformed into a higher [or possibly infinite] dimensional space wherein there may exist a hyperplane which correctly classifies both the classes in a linear sense and hence the expected classification error for the unseen test samples is minimized. The end goal in this project is to find the equation of the optimal separating hyperplane such that, the margin of separation is good enough and is robust. This problem can be approached by solving the primal problem or the dual problem. The primal problems aims to minimize at cost function, whereas, the dual problem is expressed in terms of Lagrange multipliers and the corresponding cost function has to be maximized [3], [4]. When the training set is not linearly separable, slack variables can be introduced so that, the width of the margin can be changed to accommodate the training samples. When the data points have to be
transformed to a higher dimension space, kernel functions are used. Most commonly used kernel functions are polynomial kernels, radial basis functions and the two layer perceptron. The functions that can be chosen as kernels should have lowest VC dimensions. The important advantage in using the SVM is that, only the training points that fall on the boundary of the hyperplane [support vectors] determine the nature of the classification problem. All the other training points are not involved in the calculation. So, this can be thought of as dimensionality reduction technique [5].

2. IMPLEMENTATION

Implementation was done using the MATLAB routine of the libSVM toolbox [6]. The training data and its label is given as the input and the libSVM call the ‘svmtrain’ routine, to train the SVM with regard to the training data supplied to it. Then, the test data along with the trained model is given as the input to the ‘svmpredict’ routine which gives the class labels of the test data. Based on the output labels, the accuracy of the SVM is calculated.

Each training image is of size 207*180. It is resized into a single vector of dimension 37260. A matrix containing the features of training images in its column wise is defined and is given as the input to the libSVM routine. The class label for male is +1 and for female is -1.

The other variant attempted was to apply PCA [8] [principal component analysis] to the training data before giving it as the input to the libSVM routine. Since each image has 37260 features, the covariance matrix construction in MATLAB is not possible due to memory limitations. So, the training set is downsized and converted to the dimensions of 3348. The application of PCA has its own advantage and disadvantage. PCA aims to represent the data effectively along the directions of maximum scatter. The good thing
is, the original image will have a lot of redundant information and these points will be discarded by PCA. But, the undesirable thing is that, the two classes may be well separated in their original dimension, but the PCA in the event of projecting the samples in the scatter directions, may make the samples from two classes to overlap and thereby present a difficult mixture of training samples to the libSVM routine.

Another optimization routine [LSVM] was also tried. This Lagrange support vector machine [7] takes the kernel matrix as the input and gives the Lagrange parameters as the output. All the three kernels namely, polynomial, RBF and two layer perceptron were tried and given as the input to the LSVM routine. But, in order to construct the hyperplane, Mercers theorem has to be applied to find the inner product kernel function. Since, an optimization routine that does this part could not be found; this trial could not be completed.

3. RESULTS AND DISCUSSIONS

The training set consisted of 30 faces, 15 each of male and female faces. The test data had 20 faces with random number of male and female faces.

When the training set is directly given to the libSVM [6] routine, depending on the training set and the testing set, the accuracy of the SVM classifier varied. The accuracy for most of the training-testing combinations was around 80 to 85 percent. For two cases, the accuracy shot up to 90 percent, while for some, the accuracy was around 75 percent.

When PCA [8] was done before using the libSVM optimization routine, depending on the number of eigenvectors chosen, the accuracy varied with respect to the case without PCM. The training set had its dimensions reduced to 504 and then, when the top 1000 eigenvectors were chosen, the accuracy of the SVM classifier [around 60%] was below that of the case without PCA. When the 3000 of eigenvectors were chosen, the accuracy was slightly better at 87 % for most of the test data cases. But, when the training data
was constantly changed and top 3000 eigenvectors were chosen all the time, the numbers sometimes dropped to around 70%. This says that the training set chosen holds the key for the proper performance of the optimization routine. When the training set contains the prospective candidates for support vectors [Fig 1 to 6], then the test data will most likely be classified correctly. This is because; these faces are more likely to fall on the wrong side of the hyperplane. Hence, by training the SVM with those confusing faces, better results are achieved in the test dataset as the test faces are far less confusing when compared to the support vectors.

By repeated trials, a conclusion was reached on the properties of the prospective candidates for support vectors. Male faces that resemble a female (and vice-versa), variation in brightness, variation in the view of the pose makes the faces to be wrongly classified by the SVM. When all these kinds of faces were included in the training set, and the test faces are normal male and female faces, the accuracy of the SVM was up at 90 percent. The results obtained from the experiment are given below.

Training set:
Total = 30
Male = 15 [arface001 to arface014]
Female = 15 [arface200 to arface215]

A:- Test: total = 30; first 15 from class 1.
Output class label:
1 1 1 1 1 1 1 1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
Accuracy: 26/30 = 87%

B:- Test: total = 20; first 15 from class 1.
Output class label:
1 1 1 1 1 1 1 1 1 1 -1 1 1 -1 -1 -1 -1 -1
Accuracy: 16/20 = 80%
C: Test: total = 20; all 20 from class 2
Output class label:
-1 -1 -1 -1 1 1 1 -1 -1 1 1 -1 -1 -1 -1 -1 -1 -1
Accuracy: 16/20 = 80%

D: with PCA [3000 eigenvectors] test = same as above
Output class label:
-1 -1 -1 -1 1 1 1 -1 -1 1 1 -1 -1 -1 -1 -1 -1 -1
Accuracy: 17/20 = 85%

E: using prospective support vectors in training set
Test : 10 from class1, 10 from class 2
Output class label
1 1 1 1 1 1 1 1 1 1 -1 -1 1 1 -1 -1 -1 -1 -1
Accuracy: 18/20 = 90%

4. CONCLUSIONS

The gender classification was implemented using Support Vector Machines, by giving the training data directly to the SVM classifier [6], and by doing Principal Component Analysis [8] before giving the input to the classifier. It was shown that doing PCA had its own set of advantages and disadvantages. The accuracy of the classification of these two methods was also discussed – which was found to depend strongly on the nature of the training and testing data pool. The progress made with using another optimization routine, Lagrange Support Vector Machine [7] was also explained.
5. REFERENCES


Figures of some prospective support vectors that were used for training in case E:
Fig 1 to 3: male fig 4 to 6 female
Fig 1: arface066

Fig 2: arface108

Fig 3: arface019
Fig 4: arface194

Fig 5: arface190

Fig 6: arface159