Academic year: 2005-2006

Project: Construction of a Male/Female classifier for face recognition.

Leandro Cesar Cotta 11380500

Oscar Medina Duarte 11400500

Introduction

We will present the implementation of a Support Vector Machines (SVM) classifier in order to build a model which is able to discriminate the gender of a face. Trough this paper will show the advantages and disadvantages of the SVM classifier and a simple comparison between different kernels. The kernels subject to study will be the Linear Kernel and the Gaussian Kernel.

The SVM can be used for pattern classification and linear regression as well as multilayer perceptron and radial base functions. We can say that the SVM is a linear machine with interesting properties and its implementation is a minimum risk structural approach.

We expect in this kind of classification confirm that SVM has a good answer even without absorve the problem domain knowledge, furthermore two characteristics are attractive in SVM:

- A global limit of the error surface, where the error is the difference of the wished output and the SVM output;
- Its implementation can be efficiently made using numerical approach.

Unfortunately the memory demand grows squarely with the samples of training, therefore in a real application the number of training data requires to be small to able the classification in an acceptable time. This restriction decreases the number of images particularities which can help in classifications.

Several techniques are recommended to decrease the dimension of the input. The Principal Component Analysis (PCA) technique, for example, will reduce the dimension of each image, keeping an acceptable probability of reconstruction. This technique uses the linear correlation between the centered inputs and reorganizes them in a characteristics space, afterward is possible to reconstruct the initial input with a desired error and the memory and time required to load the algorithm are decreased satisfactorily.

In this work the PCA technique chosen was the Singular Value Decomposition, because it is already implemented in Matlab. The function SVD produces a diagonal matrix S with the same dimension as X, with nonnegative diagonal elements in decreasing order, and unitary matrices U and V so that X = U*S*V'. The input dimension was decreased using the number of lines which the sum of the diagonal elements of the matrix S kept the reconstruction probability bigger than 95%. Before its utilization we normalized the data between the average point.

The SVM behavior will be improved with parameters selection. At the Gaussian kernel the parameters were sigma and C and at the linear Kernel the parameter is C. These parameters will be improved using the crossvalidation method, it consists in the pattern sample separation, where 75% will be used for data training and 25% will be used for validation. The algorithm will be loaded several times, with different values and the answers will be plotted to compare which parameters values are good.

Block Diagram

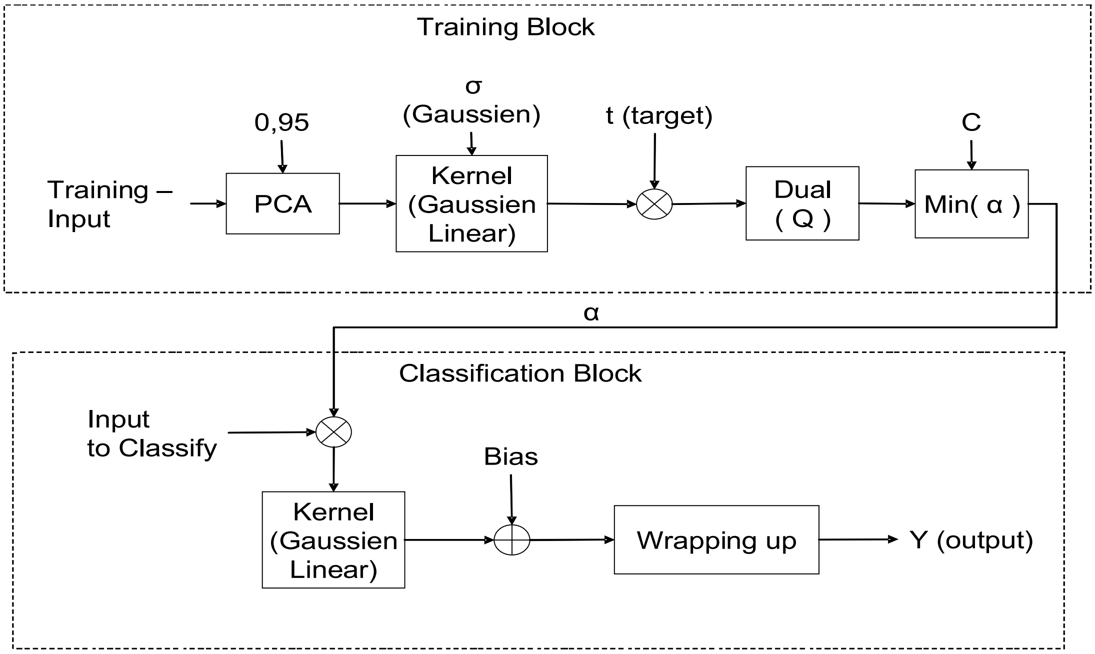

Illustration 1: Training and Classification – Block Diagrams

At illustration 1 it is considered that C and sigma had been already optimized.

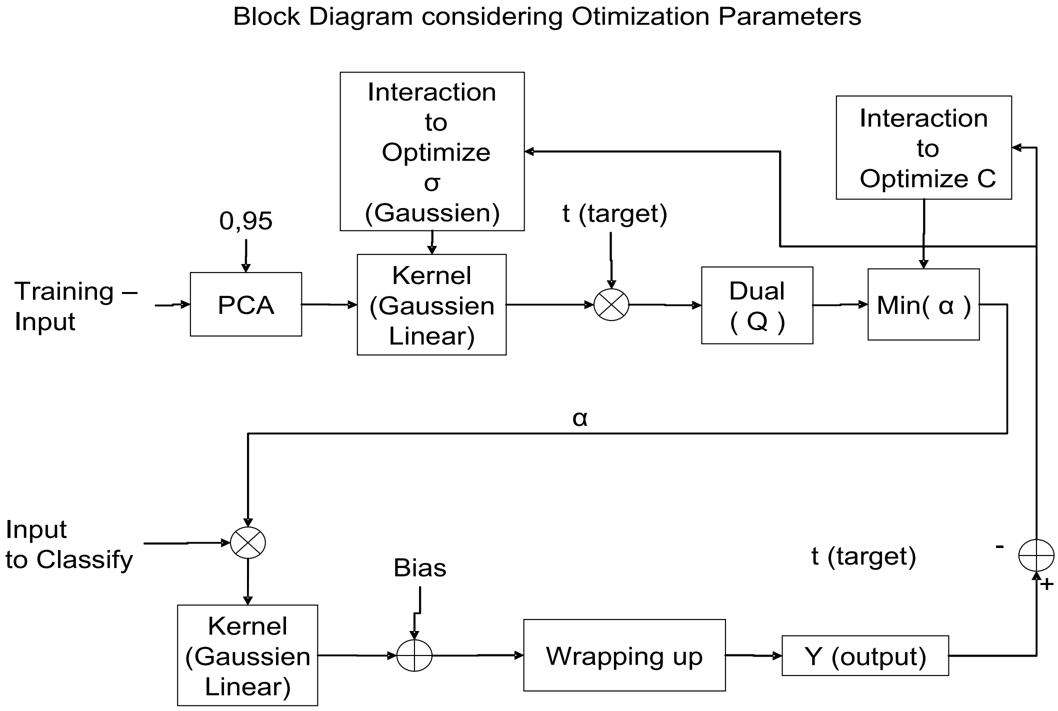

Illustration 2: Otimization Parameters - Block Diagram

Implementation of a Support Vector Machine

To implement the SVM model is necessary to find W and the bias such that the expected risk is minimized however since we can not obtain the expected risk itself we minimize the bounds, which consist of the empirical risk and the complexity term.

We can avoid the explicit usage of W by forming the dual optimization problem of a Lagrange from which we will obtain the α coefficients that minimize the problem and which will be used by the decision function.

We used brute force to find the C and σ values that minimize the MSE between the output and the target.

The results

Linear Kernel

When the Linear Kernel classification was used, we obtained the following results:

1 – Illustration 3 – Classification Output non binary:

2 – Illustration 4 – Expected value X Classified value – binary. The '*' is the classified value and the line is the expected value.

An important characteristic of a linear kernel is that for different values of C the classification error didn't changed as shown by Illustration 5. In order to confirm it was used 300 different values of C.

The answers were very good using linear Kernel, 92.1% of right classification was reached.

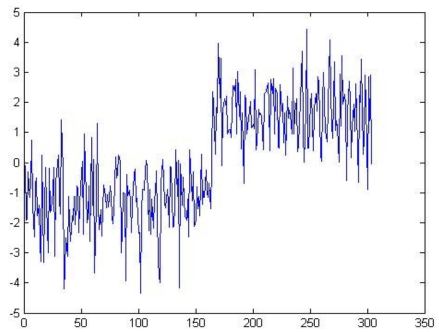

Illustration 3: Classification Output non binary , with Generic value of C.

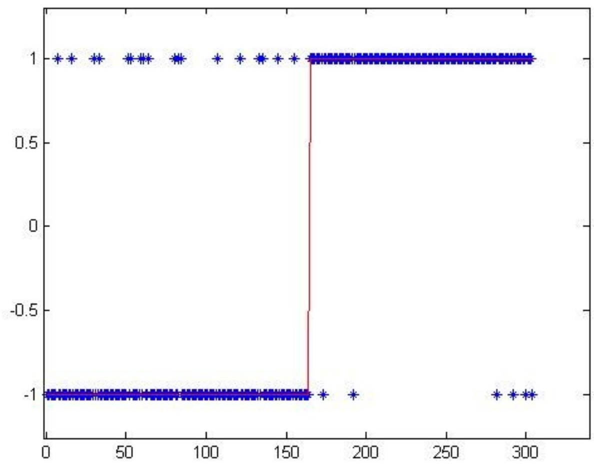

Illustration 4: Expected value X Classified value – binary

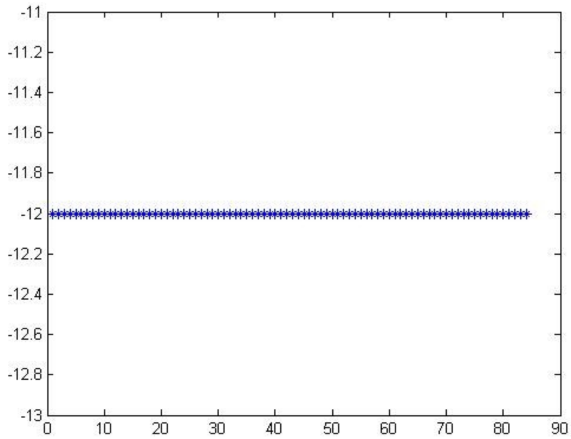

Illustration 5: Number of errors changing C Between 0 - 336

Gaussian Kernel

The Gaussian Kernel function provided the output shown by illustration 6. This graphic consists of all results of the classification using different values of C and Sigma.

1 – Illustration 6 – Gaussian Kernel Classification not binary output . Different values of C and Sigma were used for the classification.

2 – Illustration 7,8,9 – Error X Sigma X C. This illustration shows the classification error in function of variation at Sigma and C.

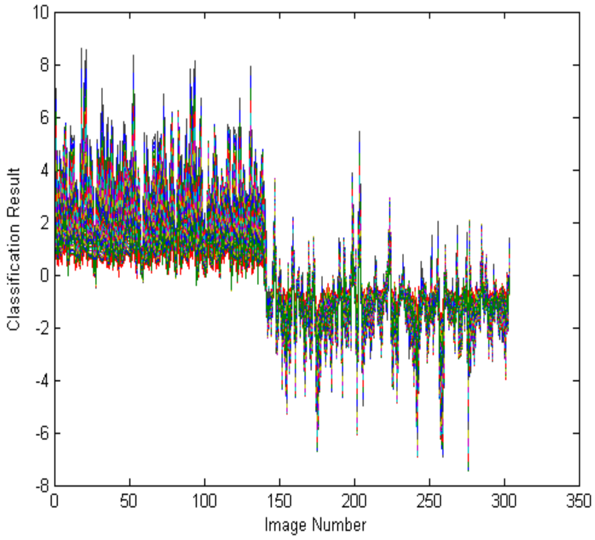

Illustration 6: Gaussian Kernel classification output

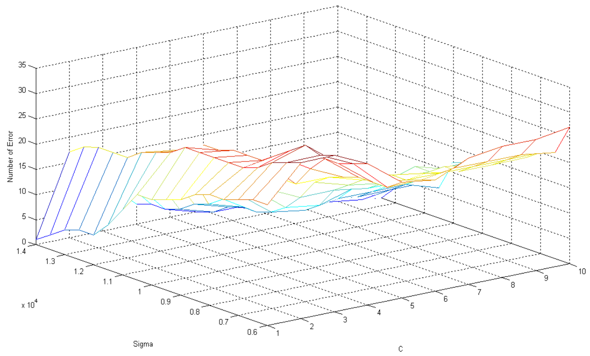

Illustration 7: Error x Sigma x C

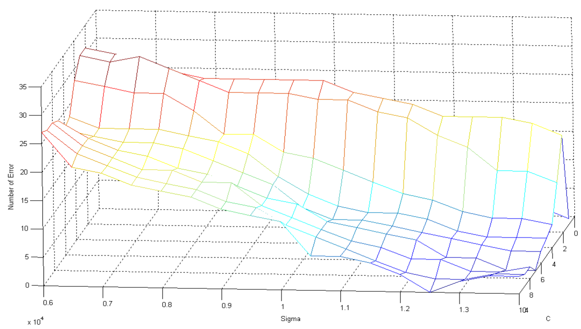

Illustration 8: Error x Sigma x C

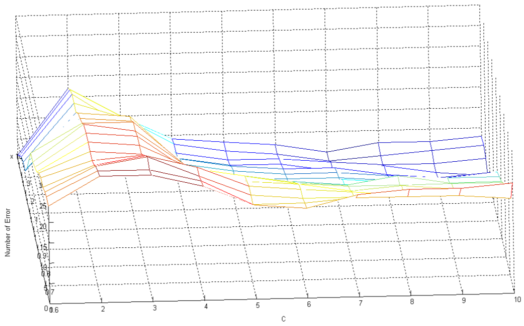

Illustration 9: Error x Sigma x C

A relation was verified between C, Sigma and the number of errors found in the output. The number of errors were changed with the modification in C and Sigma.

When we analyzed C keeping Sigma constant is possible to verify that when we made it big less is the error, however the relationship between them is not linear and when C becomes very big the error starts to increase.

Keeping C constant and changing the value of Sigma was possible to verify that as small is Sigma less is the error, but when Sigma is near to zero the error starts to increase.

While using the Gaussian Kernel with values for sigma between 6000 and 14000 and for C between 1 and 10, we obtained a range of error between 0 and 31 failed classifications. We got 100% of failures for values of C and Sigma far from the above mentioned ranges. This showed that the classification has a clear dependency on those parameters.

Conclusion

The Support Vector Machine classifier's implementation was able to discriminate the gender of a face. This paper showed some advantages and disadvantages of the SVM classifier by comparing 2 different kernels (Linear Kernel and the Gaussian Kernel).

Since memory and time requirements are quite big (polynomial), the usage of a technique to reduce the size of the problem (and therefore its computing time) was employed.

Principal Component Analysis (PCA) was the reduction technique used, keeping an acceptable probability of reconstruction and correct classification.

The SVM behavior was improved by selecting parameters while using a Gaussian Kernel. Those parameters were selected by using the crossvalidation method. The algorithm was loaded several times, with different values and the answers were plotted to find the parameter's values minimizing the error.