ROB311 - Apprentissage pour la robotique

TP4 - SVM : Digit Recognition

Mateus Lopes Ricci
Matheus Melo Monteverde

8 Octobre 2020

1 Introduction

A support vector machine (SVM) is a supervised learning method that attempts to take input data and classify it into one of two categories. For a SVM to be effective, it is first necessary to use a training input and output data set to build the SVM model that can be used for new data classification. This model works by taking the training inputs, mapping them into multidimensional space and using regression to find a hyper-plane that best separates two classes of inputs (as shown in the figure 1).

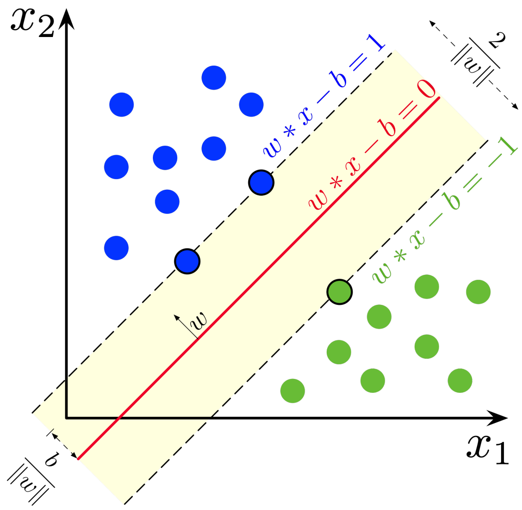

Figure 1 – Support Vector Machine

The SVM method finds the optimal separation hyperplane which divides the classes maximizing the margin of separation between them. Figure 1 shows a hyperplane of maximum margin. The red line represents the hyperplane and the dotted one, the margins.

This approach can be very powerful as it can be used alongside with the kernel-trick, making possible to define non-linear boundaries at a low computational cost, by implicitly mapping the data to a high dimensional feature space.

An example of non linear boundary utilizing kernel is shown in figure 2.

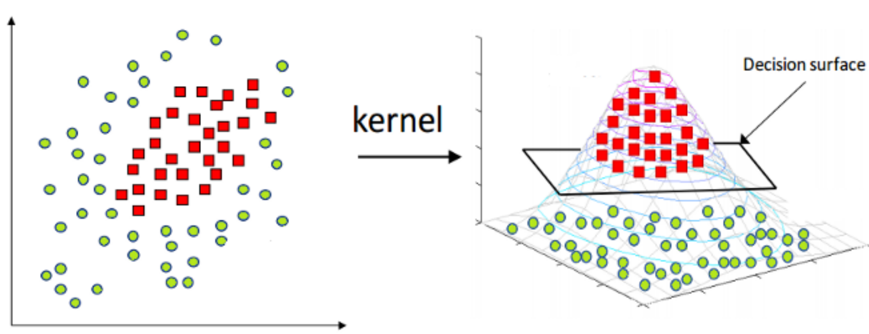

Figure 2 – Support Vector Machine

This SVM method can also be generalized to finding multiple classes by separating one class at a time from the others.

1.1 MNIST Dataset

In this project it was desired to implement the SVM in order to implement a digit recognition algorithm. The train and test database were the MNIST dataset, containing grayscale (8-bits), 28x28 images of hand written digits (figure 3).

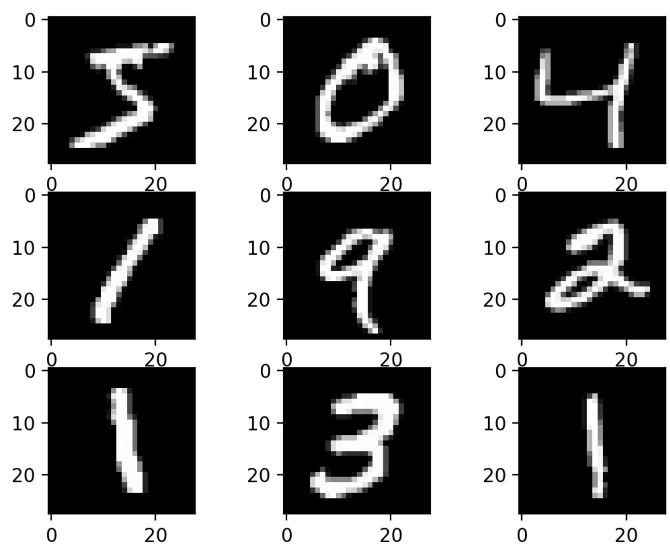

Figure 3 – MNIST Dataset examples

Since each image contains 784 pixels and the training and test dataset have 60,000 and 10,000 images respectively, it would take a long time for the code to run and the results to be obtained.

2 Principal Component Analysis

To facilitate the algorithm execution it was decided to use the Principal Component Analysis (PCA) together with the SVM. It a is the process of computing the principal components and using them to perform a change of basis on the data. In this project it was used for dimensionality reduction by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much of the data variation as possible. This way it was possible to reduce the size of features without losing its main features.

For the current analysis, we utilized the number of principal components equal to 10 and 40, weighing the processing time taken by the algorithm and precision of the results. This drastic reduction in the components can be justified by the study of the influence of multiple parameters in the SVM model, such as the kernel utilized, C and γ parameters.

3 SVM Parameters

In this section we present the different SVM models analyzed in this report. The models parameters are show in the table below :

Parameters	Values
Kernel	Linear Kernel, Gaussian Kernel
C	1, 10, 100
γ	0.002, 0.001, 0.0005

4 Results

To obtain the results for the different models, we have utilized the function of "GridSearchCV" that automatizes the process of modification of the parameters.

The accuracy results of the 5 best models and their respective parameters is shown in the table below :

Principal Components	C	γ	Kernel	Train prediction	Std Train Score
40	100	0.002	Gaussian Kernel	0.9913	$5.33 * 10^{-4}$
40	100	0.001	Gaussian Kernel	0.9783	$2.73 * 10^{-4}$
40	10	0.002	Gaussian Kernel	0.9668	$7.33 * 10-4$
40	100	0.0005	Gaussian Kernel	0.9580	$5.90 * 10-4$
40	10	0.001	Gaussian Kernel	0.9471	$9.39 * 10-4$

The best accuracy is obtained utilizing the following parameters :

- **Number of principal components** : 40
- **Kernel** : Gaussian Kernel
- **C** : 100
- γ : 0.002

The results of the accuracy for the training cross validation, training prediction and testing prediction of the best model is show in the table below :

Accuracy	Percentage
Training cross validation	0.9751
Train prediction	0.9913
Test prediction	0.9803

The accuracy results per class is shown in the table below :

Accuracy	Percentage
Digit 0	0.9949
Digit 1	0.9947
Digit 2	0.9690
Digit 3	0.9822
Digit 4	0.9776
Digit 5	0.9798
Digit 6	0.9812
Digit 7	0.9737
Digit 8	0.9805
Digit 9	0.9683

5 Confusion Matrix

With all the previous results it was possible to plot the confusion matrix (figure 4) and check the errors and hits of all classes.

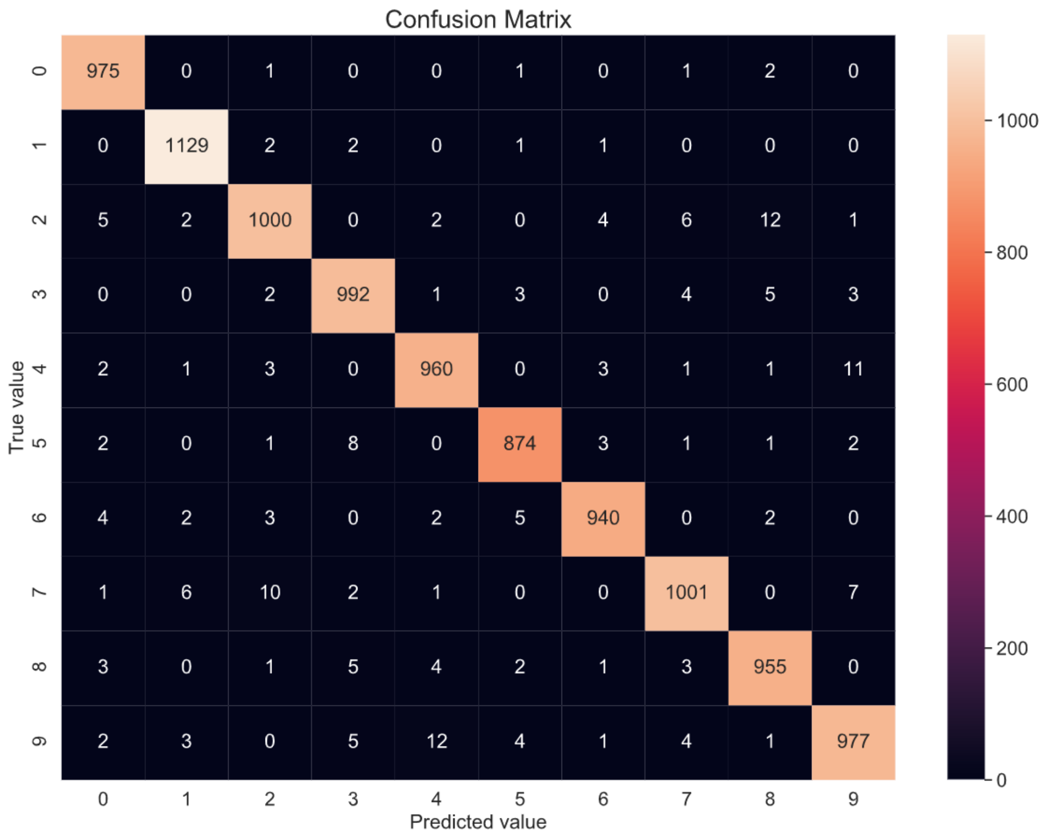

FIGURE 4 – Confusion Matrix

6 Conclusion

Analyzing the results obtained, it is reasonable to conclude that the SVM method alongside the PCA is capable to classify the handwritten digits with a very high success rate. Although the success rate is high, so is the training processing time of this algorithm, which makes the SVM a not very suitable algorithm for this application. Once the model is trained, its application for testing and predicting classes is executed very quickly.

From the confusion matrix, it is also possible to see that some classes are mistaken for others more frequently. This is the case for the classes 4-9, 2-8 and 7-1 that have some similar handwritten style.