# CE4708 – Artificial Intelligence Project Report

Paul Lynch (16123778) Ciaran Carroll (13113259) Kevin Burke (14155893)

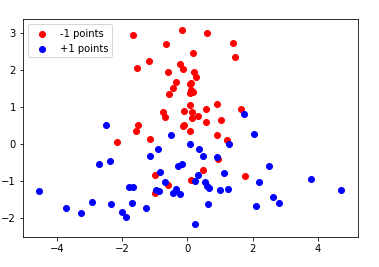
**Project Description:**

Develop and test a soft-margin kernel based Support Vector Machine (SVM) for a 2-d classification problem. For this project, the classification problem is not linearly separable, so a Radial Basis Function (RBF) is needed with σ2 = 0.25.

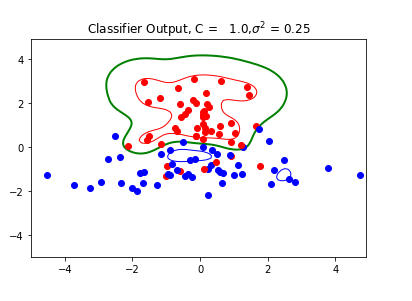
Our classifier is built around a Convex-Optimisation Program Solver which is used to make Lagrange Multipliers (lambdas) for the classification problem. The Lagrange multipliers are used to select the classifier’s Support Vectors and set its bias.

To change the hard-margin SVM provided to a soft-margin SVM I had to change the G and h matrices in order solve the Wolfe Dual with CVXOPT. The G matrix needs to be converted from a n x n matrix with -1 on the main diagonal and 0’s everywhere else to a 2n x n matrix with -1 on the top diagonal and 1 on the bottom diagonal. The h matrix needs to be converted from n column vector of 0’s to a 2n column vector of n 0’s and n C’s. The C parameter determines used to optimise the SVM machine as it controls the trade-off between a smooth decision boundary and classifying all the data points correctly.

When a large value of C is applied an RBF kernel SVM there were less incorrect misclassifications, but the margins were narrow, and the decision boundary was very complex which differs from lower values of C as the lowering of this parameter will allow more misclassifications with wider margins and a less complicated decision boundary.

**Training the SVM:**

*Fig 1: Training dataset*

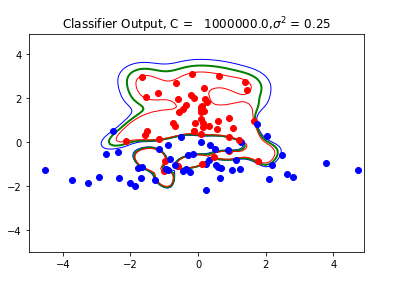


*Fig 2: Classifier Output – C = 1*

-1 Margin Boundary **RED**

-1 Margin Boundary **BLUE**

Classification Boundary **GREEN**



*Fig 3: Classifier Output – C =*

-1 Margin Boundary **RED**

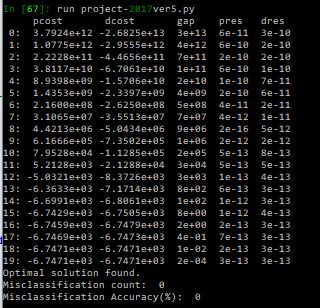
-1 Margin Boundary **BLUE**

Classification Boundary **GREEN**

**Conclusion**

*Training Dataset Classification Accuracy*

The Classification accuracy for the training dataset can be seen in the two figures below. With a C parameter of 1.0 the SVM classifies the data and misclassifies 9% of the data. When the C parameter is changed to , the margins are made significantly smaller and the SVM classifies all points correctly, resulting in a 0% error.



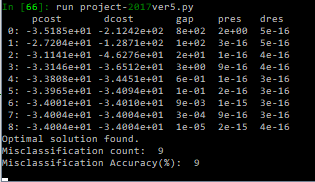
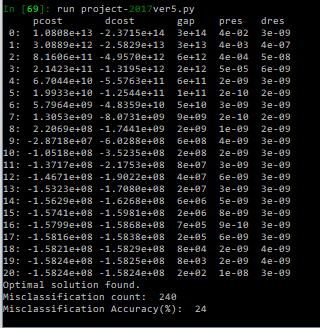


Figure 4 C = 1.0 (Left) and C = (Right) for the Training Dataset

*Testing Dataset Classification Accuracy*

When the previously unseen, larger testing dataset is used the classification accuracy changes. Fig. 5 shows the accuracy for both C parameters. With C = 1.0, the SVM misclassifies 127 points, resulting in approximately a 12% inaccuracy. With the larger C parameter, 240 points are misclassified, leading to a 24% misclassification.

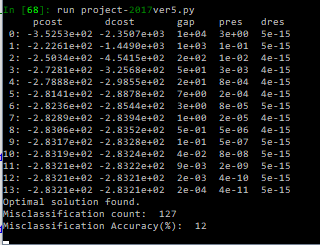


Figure 5 C = 1.0 (Left) and C = (Right) for the Testing Dataset

*Discussion*

It should be noted that the classification results for the training dataset are very often misleading, producing statistics which are very good but not realistic for other datasets.

In terms of overfitting of the data, it is impossible to see this on the training dataset. An independent dataset is needed in order to estimate whether the SVM is overfitting or not.

|  |  |  |
| --- | --- | --- |
| Misclassifications | C = 1 | C = |
| Testing Dataset | 9 | 0 |
| Training Dataset | 127 | 240 |

The above table shows the number of misclassifications for each classifier on both datasets. From this, it can be seen that the classifier is overfitted. With this larger value for the C parameter, the SVM behaves more like a hard-margin classifier. It produces narrower margins and attempts to classify all points correctly, resulting in overfitting. The smaller C parameter allows the SVM to have softer margins, producing some misclassifications.