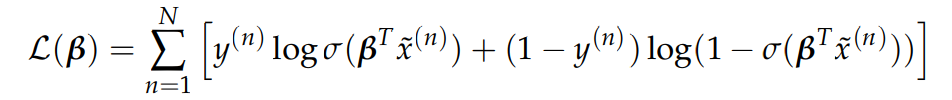
MODULE 3F8: INFERENCE

LAB REPORT

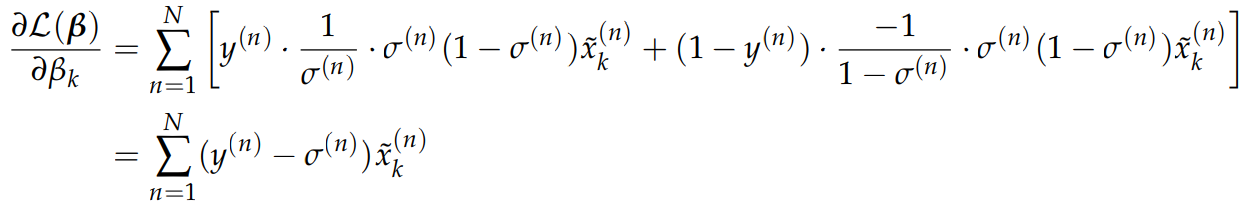
Yufeng Zhao [yz496@cam.ac.uk](mailto:yz496@cam.ac.uk) Magdalene College

(a). Gradient of the Log-likelihood of the parameters

By definition, the log-likelihood function is given by:

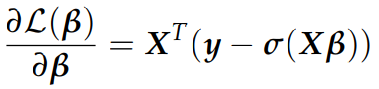


Evaluate the derivative of the log-likelihood function:



In which

By vectorization:



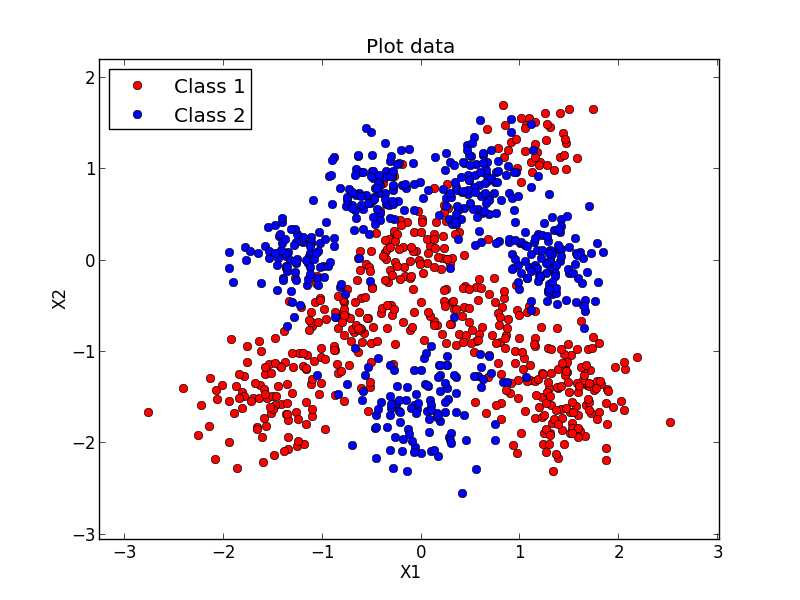
(b). Pseudocode to estimate the parameters

*for i in range(train\_steps):*



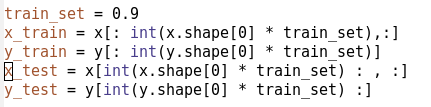
Learning rate η is chosen to be higher if the learning process is slow, i.e. the log-likelihood is not converging fast enough, and is chosen to be lower if the learning process becomes oscillatory.

(c). Visualisation of the dataset



It is clearly observed from the plot that the boundaries of two data type are not straight lines. Therefore, a linear classifier will not perform well on this data.

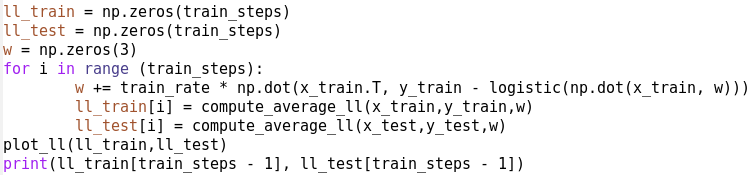
(d). Split data into training and test sets



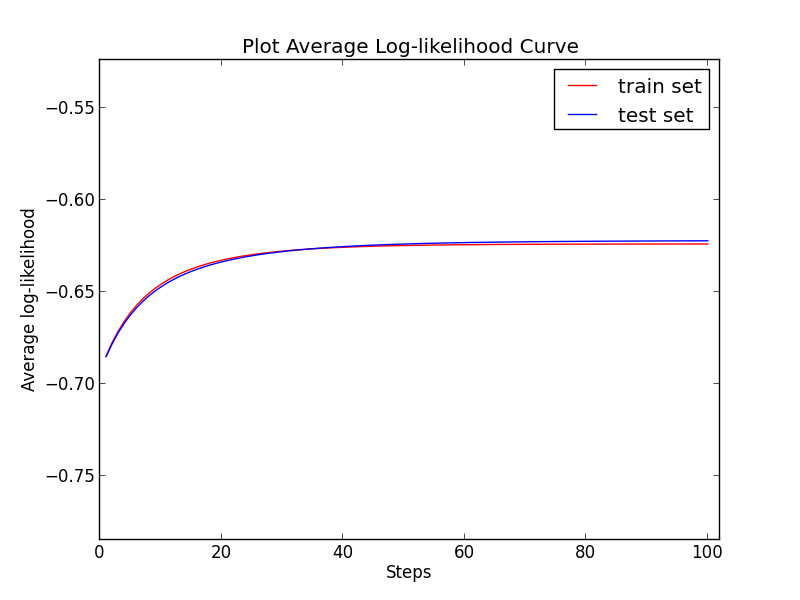
The data is split into training and test sets by the ratio of 90:10. In this case we set the first 900 data points to be training set and the last 100 test set. The ratio selected is calibrated to avoid overfitting and to make sure there is still enough data for test.

(e). Linear Classification

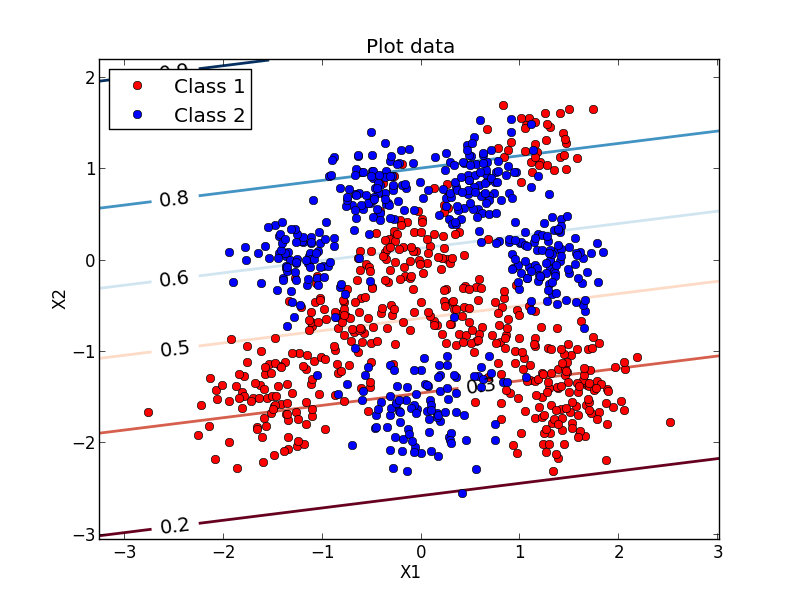
Python code for training and log-likelihood calculation:



Training Log-likelihood Curves:



Probability contour of the predictive distribution against original data:



(f). Final log-likelihood and confusion matrix

Final log-likelihood of training: -0.6237

Final log-likelihood of test: -0.6219

Confusion matrix:

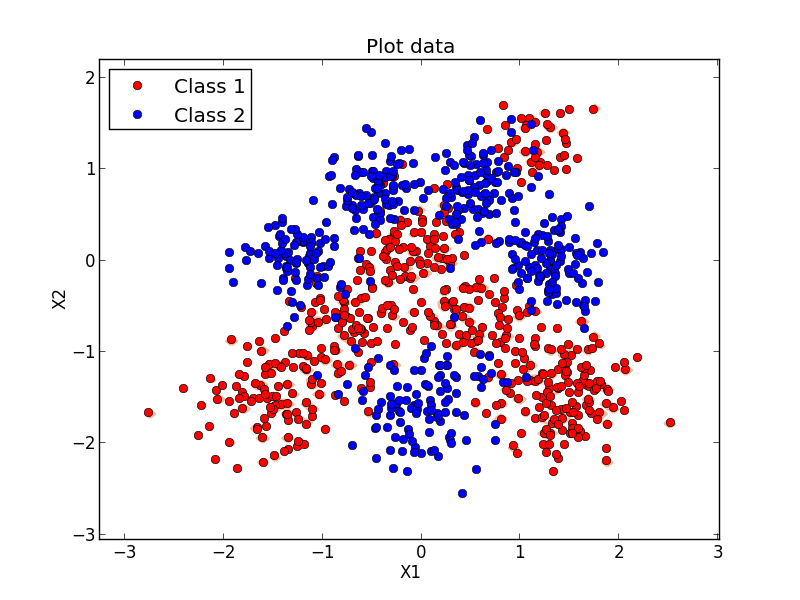
|  |  |  |  |
| --- | --- | --- | --- |
|  | | Predicted Label | |
| 0 | 1 |
| True Label | 0 | 0.7255 | 0.2745 |
| 1 | 0.3469 | 0.6531 |

Training accuracy: 68.89%

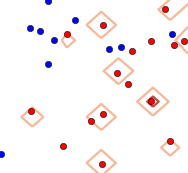
Test accuracy: 68.00%

(g). Predictions with Radial Basis Functions (RBF)

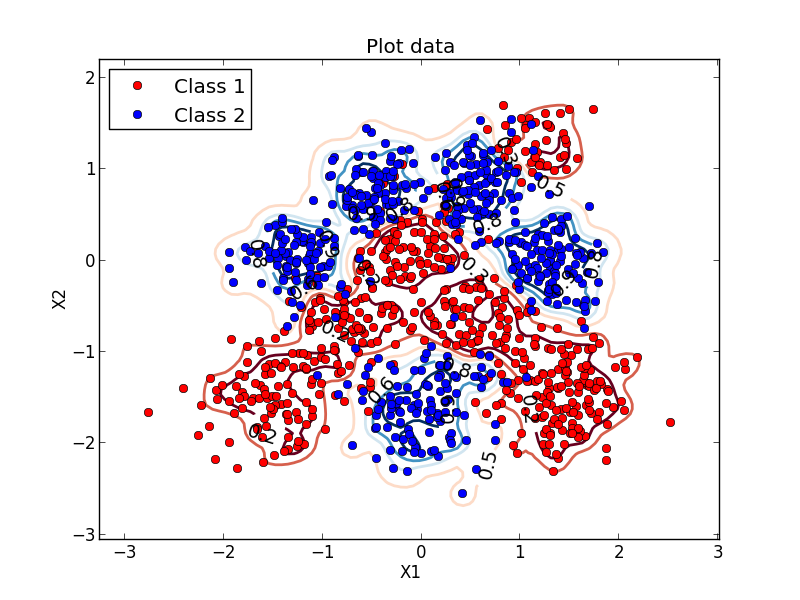
L = 0.01:



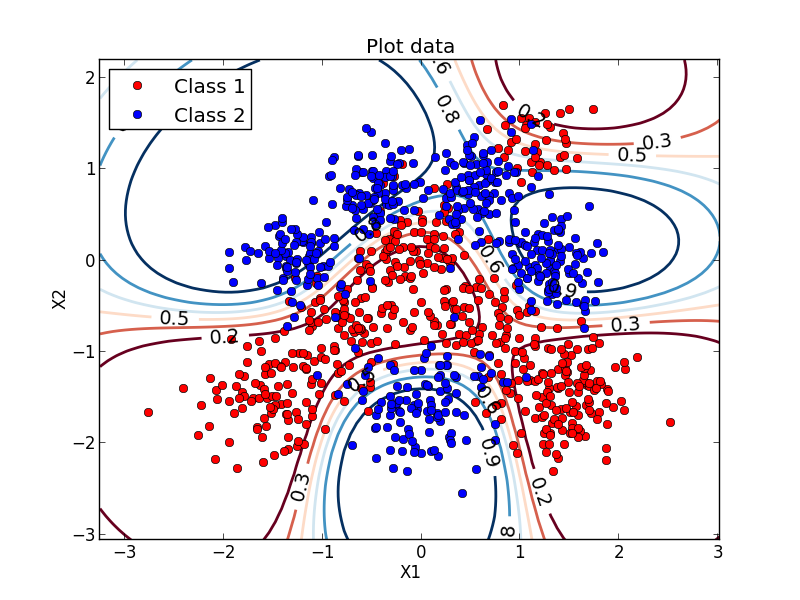
L = 0.01 (zoom in):



L = 0.1:



L = 1:



(h). Final log-likelihood and confusion matrix

L = 0.01:

Final log-likelihood of training: -0.4827 Final log-likelihood of test: -0.6842

Confusion matrix:

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Predicted Label | |
| 0 | 1 |
| True Label | 0 | 0.9804 | 0.0196 |
| 1 | 0.8980 | 0.1020 |

Train accuracy: 99.89% Test accuracy: 55.00%

L = 0.1:

Final log-likelihood of training: -0.2126 Final log-likelihood of test: -0.3213

Confusion matrix:

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Predicted Label | |
| 0 | 1 |
| True Label | 0 | 0.9412 | 0.0588 |
| 1 | 0.2041 | 0.7959 |

Train accuracy: 93.22% Test accuracy: 87.00%

L = 1:

Final log-likelihood of training: -0.2363 Final log-likelihood of test: -0.2729

Confusion matrix:

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Predicted Label | |
| 0 | 1 |
| True Label | 0 | 0.9216 | 0.0784 |
| 1 | 0.1429 | 0.8571 |

Train accuracy: 91.11% Test accuracy: 89.00%

Discussion:

The RBF model with L = 0.01 performs poorly on the test data due to overfitting. The decision boundaries are essentially small encirclements around the source data points, which can be observed from the zoomed in plot. This results in bad test performance.

For properly selected width (e.g. L = 0.1, 1), the RBF model is superior to the original source, with ~90% test accuracy compared with ~70%. This is because the radial basis functions applied allow the classifier to achieve more complicated decision boundaries rather than simple linear model. Also, applying RBF provides information on relative distance from training data points which helps in the classification of test data points.