DD2421 - Lab 3

Pierre Rudin

August 13, 2018

1 Introduction

In this lab we will look at Bayesian learning and boosting.

2 Assignments

2.1 Assignment 1

Write mlParams(X, labels). Plot ML-estimates with 95 % confidence interval

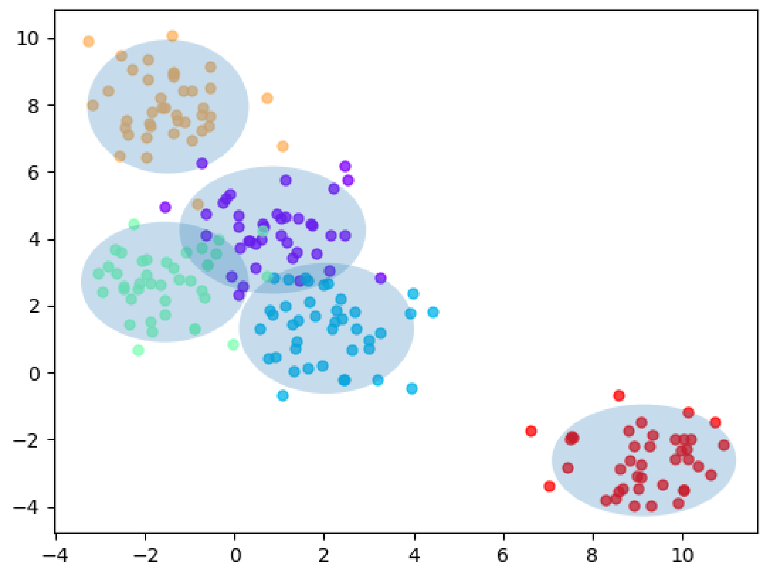

Figure 1: ML-estimates with 95 % confidence interval

2.2 Assignment 2

Write function computePrior(labels), Write function classifyBayes(X,prior,mu,sigma).

2.3 Assignment 3

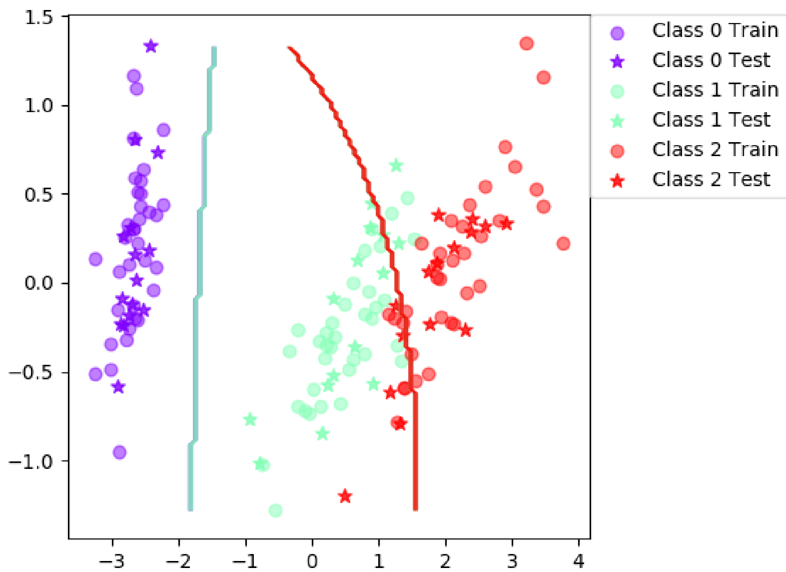

Figure 2: Classified Iris dataset

(1) When can a feature independence assumption be reasonable and when not?
A easy way to find out is to try :), but more serious, it is hard to tell. If we plot our datasets and they are well seperated, it would be fair to guess that the features are independent. However, if they are scattered in a big mess. We cant really tell that much and maybe we need to do some mathematical operations on some samples features to tell which class it belongs to, and in that case I think they are dependent of each other

(2) How does the decision boundary look for the Iris dataset?
See image.

(3) How would one improve the classification results for this scenario by changing classifier or, alternatively, manipulate the Iris dataset?
To improve, I would continue through this lab and hope the results gets better :) otherwise I would maybe us a svm with some slack. Because it looks kinda like a problem where that

model would be useful. Some perceptron-model should also be able to solve this problem quite well.

2.4 Assignment 4

Update mlParams.

2.5 Assignment 5

Trial	Accuracy
0	84.4
10	95.6
20	93.3
30	86.7
40	88.9
50	91.1
60	86.7
70	91.1
80	86.7
90	91.1

Table 1: Iris w/o boosting, Final mean classification 89 with standard deviation 4.16

Trial	Accuracy
0	95.6
10	97.8
20	93.3
30	93.3
40	97.8
50	93.3
60	95.6
70	95.6
80	93.3
90	93.3

Table 2: iris w boosting, Final mean classification 94.5 with standard deviation 2.92.

Trial	Accuracy
0	61
10	66.2
20	74
30	66.9
40	59.7
50	64.3
60	66.9
70	63.6
80	62.3
90	70.8

Table 3: vowel w/o boosting, Final mean classification 64.7 with standard deviation 4.03

Trial	Accuracy
0	68.2
10	68.8
20	74
30	70.1
40	67.5
50	69.5
60	72.1
70	68.8
80	73.4
90	70.8

Table 4: vowel w boosting, Final mean classification 71.3 with standard deviation 4.62.

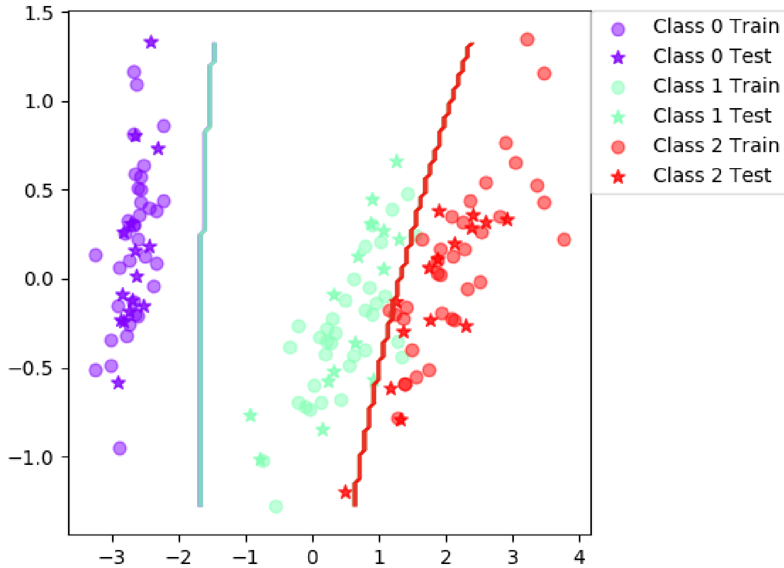

Figure 3: Iris dataset, classified with naive Bayes and boosting.

(1) Is there any improvement in classification accuracy? Why/why not?
Yes, the results are better with 5-6 points. On the iris set that almost removes half the errors existing before, and it is a good improvement on the vowel set as well. This is probably because the datasets are to complex to classify with only the naives bayes, this more complex solution provides a good complement to the existing. If we only look at the line in between class 2 and 3 in the iris data set, the weights in the boost probably makes the classifiers that want to draw the line with a slight curve to the left less important. It also helps that the classifiers train on multiple variants of the datasets.

(2) Plot the decision boundary of the boosted classifier on iris and compare it with that of the basic. What diferences do you notice? Is the boundary of the boosted version more complex?
I would not say more complex, how ever it seems to fit the dataset better.

(3) Can we make up for not using a more advanced model in the basic classifier (e.g. independent features) by using boosting?
Well yes, the results are improved. So, yes at least in the scenarios given in this lab.

2.6 Assignment 6

	Accuracy
0	95.6
10	100
20	91.1
30	91.1
40	93.3
50	91.1
60	88.9
70	88.9
80	93.3
90	88.9

Table 5: Iris w/o boosting, Final mean classification accuracy 92.4 with standard deviation 3.71

	Accuracy
0	95.6
10	100
20	95.6
30	93.3
40	93.3
50	95.6
60	88.9
70	93.3
80	93.3
90	93.3

Table 6: Iris w boosting, Final mean classification accuracy 94.6 with standard deviation 3.65

	Accuracy
0	63.6
10	68.8
20	63.6
30	66.9
40	59.7
50	63
60	59.7
70	68.8
80	59.7
90	68.2

Table 7: Vowel w/o boosting, Final mean classification accuracy 64.1 with standard deviation 4

	Accuracy
0	85.7
10	89.6
20	86.4
30	93.5
40	83.8
50	79.2
60	90.3
70	85.7
80	85.1
90	86.4

Table 8: Vowel w boosting, Final mean classification accuracy 86.9 with standard deviation 3.08

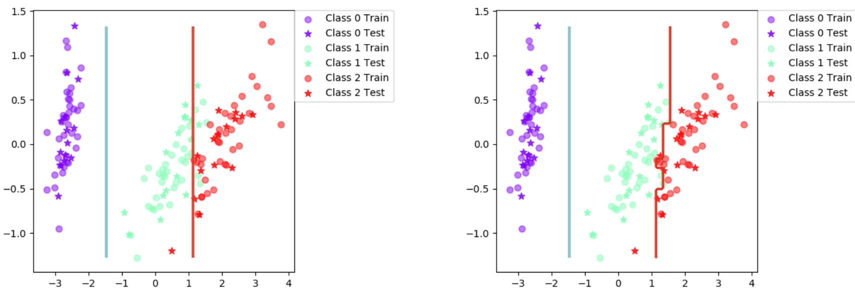

Figure 4: Iris dataset, classified with decision tree. Left w/o boosting, right w boosting.

(1) Is there any improvement in classification accuracy? Why/why not?
Yes, significant improvements on the Vowel dataset. Barely noticeable on Iris dataset. Why? I don't know...

(2) Plot the decision boundary of the boosted classifier on iris and compare it with that of the basic. What diferences do you notice?
The boundary of the boosted it looks a bit more complex this time. I would almost like to suggest that it looks a bit over-fitted, mainly because the separating line doesn't look as natural anymore.

(3) Can we make up for not using a more advanced model in the basic classifier (e.g. independent features) by using boosting?
In the Vowel case most definitely. The results are much better with boosting. On the Iris dataset I'm not equally sure. It might start to becoming overfitted. And the extra computation required might not be worth it.

2.7 Assignment 7

If you had to pick a classifier, naive Bayes or a decision tree or the boosted versions of these, which one would you pick? Motivate from the following criteria:

- Outliers - Here I would say that the naive Bayes with boosting would be to prefer, if were looking at the graphs of Iris datasets and how they are classified, the classification suspiciously follows the contour made by the dots on the graph, creating some weird shapes, while naive Bayes simply draws a best-fit-line through the data set in a place that seems likely.
- Irrelevant inputs: part of the feature space is irrelevant - Same as previous answer.
- Predictive power - A thougher bet, While the decision tree had a lower error it didn't look that generalized. But I think however that the performance on the vowel set makes it a winner, especially when boosting is used.

- Mixed types of data: binary, categorical or continuous features, etc - I would guess that naive Bayes does this a little bit better, mainly because it would be able to handle continous data easier.
- Scalability: the dimension of the data, D, is large or the number of instances, N, is large, or both - Matrices in general aren't that funny to calculate as they grow bigger, so I think the decision trees are a winner here.