Machine Learning - Lab 3 (Baysian Learning and Boosting)

hfag@kth.se

March 11, 2022

Assignment 1

Returned in our mean function is a C $\times$ d matrix. C is the number of classes, and d the chunks of datasets. Returned for the variance is a C $\times$ d $\times$ d tensor to compute for each class and data chunks the co-variance between the labeled data. Seen below is a plot of the boundary's for the ML-estimates of mu and sigma.

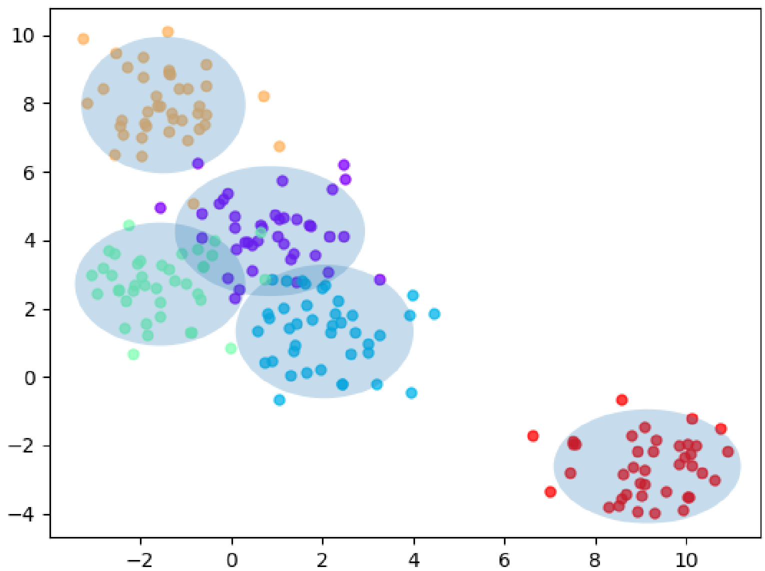

Figure 1: Plot of the 95% interval of the ML-estimates.

Assignment 2

(1)

Compute prior is done by implementing,

$$p(k) = \frac{N_k}{N},$$

i.e compute the frequency of the class divided by total amount. That is how common the class is in percentage.

(2)

Implemented in classifyBayes is the discriminant function, equation 11. The covariance and mean is used to compute the discriminant function for each class and point. argmax() is then used to find the max a-posteriori and the information is then used to assign the point to that class.

Returned is a vector with length N, i.e amount of data points, containing the predicted class value for each point.

Assignment 3

Accuracy test for **Iris** dataset:

Trials	**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

For the Iris dataset I obtained a mean classification accuracy of 89 with standard deviation of 4.16.

Accuracy test for **Vowel** dataset:

Trials	**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

For the Vowel dataset I obtained a mean classification accuracy of 64.7 with standard deviation of 4.03.

(1)

For the conditional Independence assumption to be true, one would have to have either small dependence or none at all. I.e. non-diagonal terms close to zero in covariance matrix.

(2)

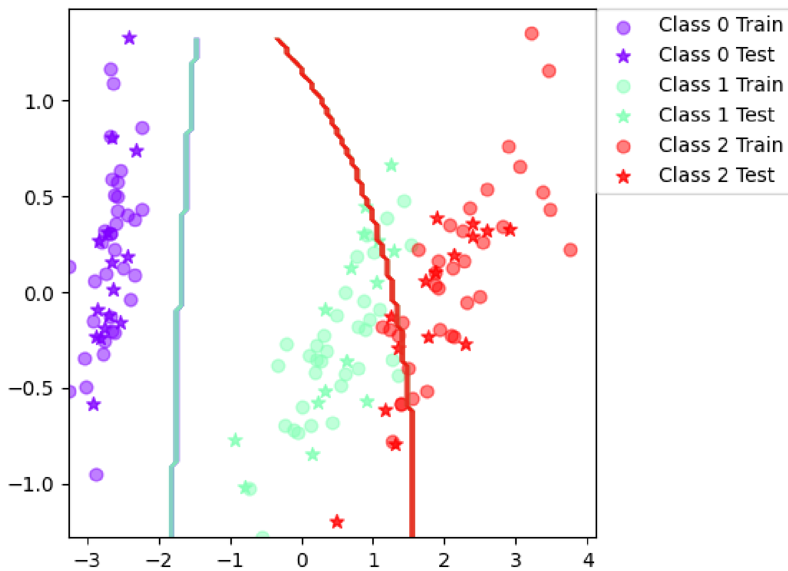

Figure 2: Decision boundary for the Iris dataset.

By manipulating the data one could possibly remove the overlapping datapoints, this could make the boundary easier to identify for the method. One could possibly use some kind of SVM's with slack as an alterative method.

Assignment 4

In assignment 4 the programming that was done was implementing the input W into the function sigma and mu. W is a vector of size 1 $\times$ N that is simply just multiplied on all of the entries. The weights is then updated over iteration by looking at the error. The error is computed by looking at the wrongly/correctly classified labeled data points. It then adjust the values and the error gets smaller and smaller.

Assignment 5

Accuracy test for **Iris** dataset using boosted classifier:

Trials	**Accuracy**
0	95.6
10	100
20	93.3
30	91.1
40	97.8
50	93.3
60	93.3
70	97.8
80	95.6
90	93.3

For the Iris dataset I obtained a mean classification accuracy of 94.7 with standard deviation of 2.82.

Accuracy test for **Vowel** dataset using boosted classifier:

Trials	**Accuracy**
0	76.6
10	86.4
20	83.1
30	80.5
40	72.7
50	76
60	81.8
70	82.5
80	79.9
90	83.1

For the Vowel dataset I obtained a mean classification accuracy of 80.2 with standard deviation of 3.52.

(1)

Mean classification accuracy:

	Iris	Vowel
Basic	89	64.7
Boosted	94.7	80.2
Difference	5.7	15.5

Standard deviation:

	Iris	Vowel
Basic	4.16	4.03
Boosted	2.82	3.52
Difference	1.34	0.51

(2)

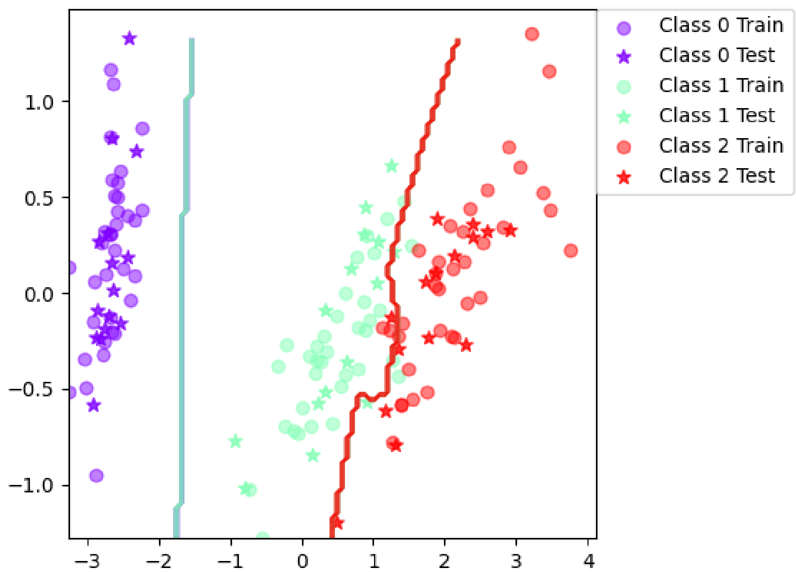

Figure 3: Decision boundary for the Iris dataset using boosted classifier.

Instead of having a curved line, the boundary now have imposed a combination of linear lines which could be considered more complex. It follows the pattern clearly loads better than the earlier model.

(3)

Yes, the boosting algorithm allows for a more complex model which can better fit the data. As seen in the plots.

Assignment 6

Accuracy test for **Iris** dataset using decision tree classifier:

Trials	**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

For the Iris dataset I obtained a mean classification accuracy of 92.4 with standard deviation of 3.71.

Accuracy test for **Iris** dataset using decision tree classifier with boosted classifier:

Trials	**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

For the Iris dataset I obtained a mean classification accuracy of 94.6 with standard deviation of 3.65.

Accuracy test for **Vowel** dataset using decision tree classifier:

Trials	**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

For the Vowel dataset I obtained a mean classification accuracy of 64.1 with standard deviation of 4.

Accuracy test for **Vowel** dataset using decision tree classifier with boosted classifier:

Trials	**Accuracy**
0	85.7
10	85.7
20	85.7
30	91.6
40	84.4
50	79.2
60	89
70	86.4
80	88.3
90	89

For the Vowel dataset I obtained a mean classification accuracy of 86.5 with standard deviation of 3.15.

(1)

Mean classification accuracy:

	Iris	**Vowel**
Basic	89	64.7
Boosted	94.7	80.2
Decision	92.4	64.1
Decision boost	**94.6**	**86.5**

Standard deviation:

	Iris	**Vowel**
Basic	4.16	4.03
Boosted	**2.82**	3.52
Decision	3.71	4
Decision boost	3.65	**3.15**

(2)

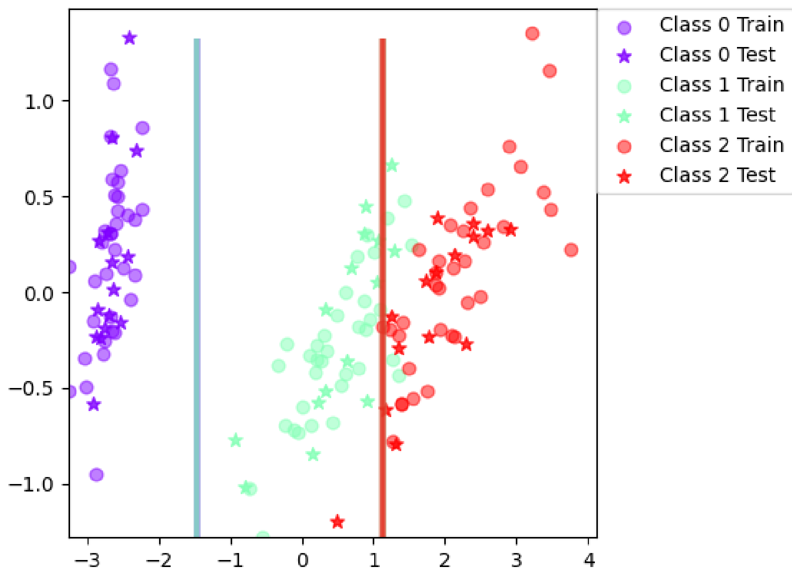

Figure 4: Decision boundary for the Iris dataset using decision tree classifier.

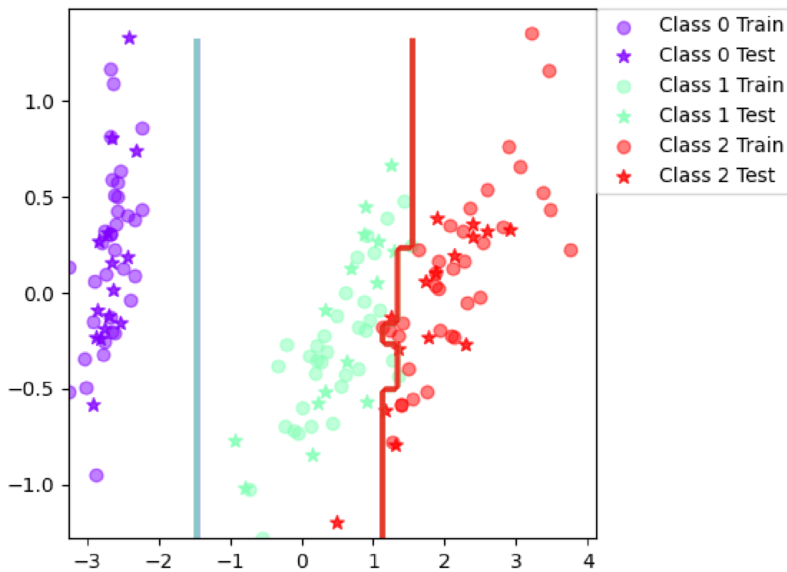

Figure 5: Decision boundary for the Iris dataset using decision tree classifier with boosted classifier.

(3)

Yes, the boosting algorithm allows for a more complex model which can better fit the data. As seen in the plots.

Assignment 7

- **Outliers:** Naive bayes, without boosting. If we were to include boosting it would overcompensate and try to adjust for outliers, which will make the classifier worse.
- **Irrelevant inputs:** Decision tree, since it is based on the information gained and can ignore irrelevant inputs. By using pruning, the irrelevant inputs could be removed.
- **Predictive power:** Naive bayes with boosting would be best, given the dataset is not to highly correlated.

- **Mixed types of data:** Decision tree is better for mixed data types. Seeing as non-continuous data type is problematic in the bayes classifier.
- **Scalability:** Decision trees, as it quite quickly can categorize the data and make the problem smaller. Decision trees would also be preferred when taking computational time in to consideration.