

Assignment 1

Please refer to the code for implementation of `mlParams()`. Figure 1 shows the result.

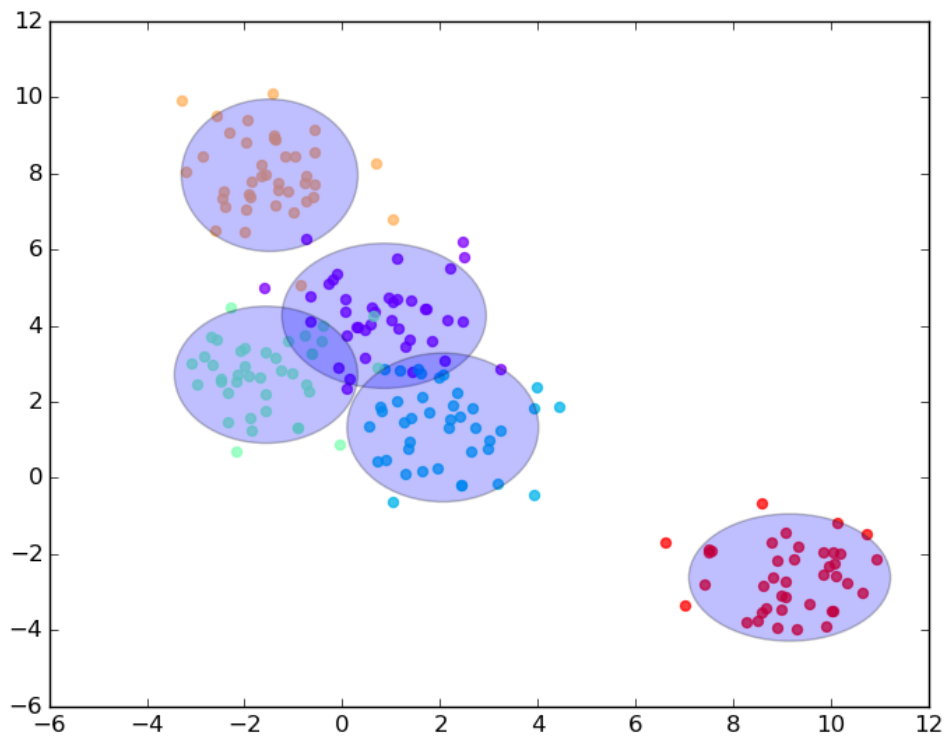


Figure 1: 95 % confidence intervals for data generated by `genBlobs(centers = 5)`

Assignment 2

Refer to code for implementations of `computePrior()` and `classifyBayes()`.

Assignment 3

The result of running `testClassifier(BayesClassifier(), dataset='iris', split=0.7)` and `testClassifier(BayesClassifier(), dataset='vowel', split=0.7)` is given:

```

Trial: 0 Accuracy 84.4
Trial: 10 Accuracy 95.6
Trial: 20 Accuracy 93.3
Trial: 30 Accuracy 86.7
Trial: 40 Accuracy 88.9
Trial: 50 Accuracy 91.1
Trial: 60 Accuracy 86.7
Trial: 70 Accuracy 91.1
Trial: 80 Accuracy 86.7
Trial: 90 Accuracy 91.1
Final mean classification accuracy 89 with standard deviation 4.16
Trial: 0 Accuracy 61
Trial: 10 Accuracy 66.2
Trial: 20 Accuracy 74
Trial: 30 Accuracy 66.9
Trial: 40 Accuracy 59.7
Trial: 50 Accuracy 64.3
Trial: 60 Accuracy 66.9
Trial: 70 Accuracy 63.6
Trial: 80 Accuracy 62.3
Trial: 90 Accuracy 70.8
Final mean classification accuracy 64.7 with standard deviation 4.03

```

The decision boundary for the 2D-iris data is depicted in Figure 2.

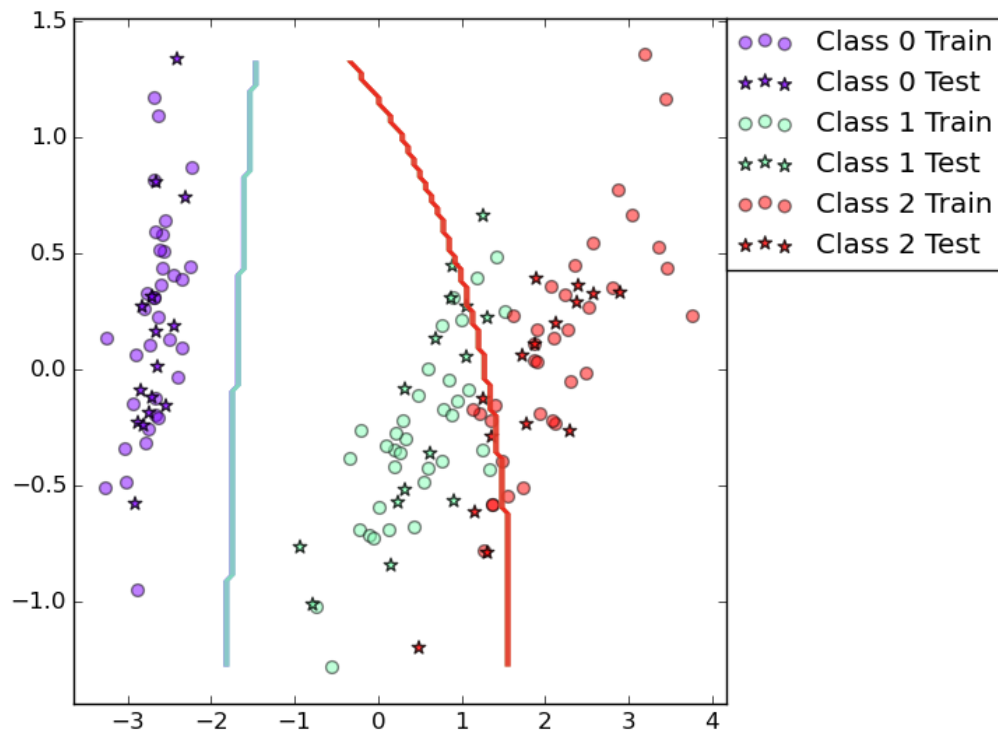


Figure 2: Boundary for the iris data.

Questions:

1) When can feature independence assumption be reasonable?

A: When one may assume that there is no or little correlation between the features. In the Iris-set one may assume that there is positive correlation between petal width and petal length (or sepal width & length), c.f. <https://archive.ics.uci.edu/ml/datasets/Iris>.

2) How does the decision boundary look for the iris dataset? How could one improve the results for this scenario by changing the classifier, or by manipulating the data?

A: It looks like class 0 is has fairly uncorrelated features while class 1 and class 2 has not. One could use a “non-naive” Bayes classifier, i.e. where it is not necessarily the case that $\Sigma(m, n) = 0, \quad n \neq m$. One could possibly make non linear transformation of the data.

Assignment 4

Refer to code for the augmented functions `mlParams()`

Assignment 5

Refer to code for augmented function `computePrior()`. Also for implementations of `trainBoost()` and `classifyBoost()`. The results of running

```
testClassifier(BoostClassifier(BayesClassifier(), T=10), dataset='iris', split=0.7)
testClassifier(BoostClassifier(BayesClassifier(), T=10), dataset='vowel', split=0.7)
```

are given below. Decision boundary for the iris data using boosting is given in Figure 5.

```

Trial: 0 Accuracy 95.6
Trial: 10 Accuracy 100
Trial: 20 Accuracy 93.3
Trial: 30 Accuracy 91.1
Trial: 40 Accuracy 97.8
Trial: 50 Accuracy 93.3
Trial: 60 Accuracy 93.3
Trial: 70 Accuracy 97.8
Trial: 80 Accuracy 95.6
Trial: 90 Accuracy 93.3
Final mean classification accuracy 94.7 with standard deviation 2.82
Trial: 0 Accuracy 76.6
Trial: 10 Accuracy 86.4
Trial: 20 Accuracy 83.1
Trial: 30 Accuracy 80.5
Trial: 40 Accuracy 72.7
Trial: 50 Accuracy 76
Trial: 60 Accuracy 81.8
Trial: 70 Accuracy 82.5
Trial: 80 Accuracy 79.9
Trial: 90 Accuracy 83.1
Final mean classification accuracy 80.2 with standard deviation 3.52

```

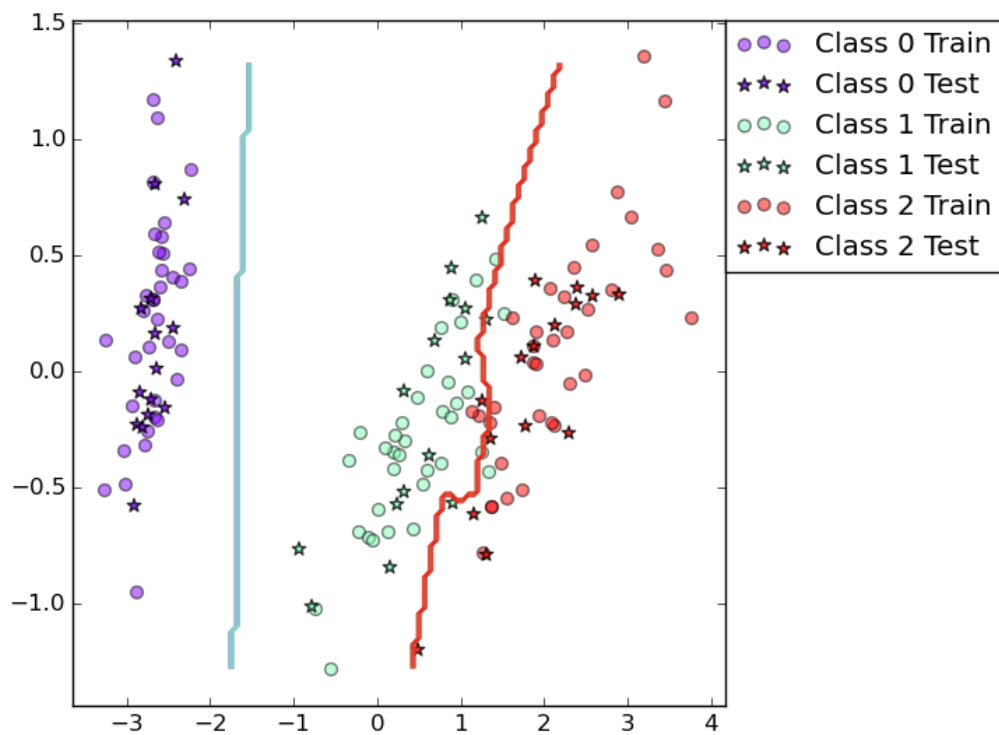


Figure 3: Boundary for the iris data using boosting.

Questions:

1) Is there any improvement in classification accuracy? Why/why not?

A: Yes, this is because the boosting puts more weight on the missclassified points

TODO, har
vi koll på
detta?

2) Plot the decision boundary of the boosted classifier for the iris set. What differences do you notice? Is the boundary of the boosted version more complex?

A: See Figures 2 and 5. Yes, more complex. Also copes better with the correlation

3) Can we make up for not using a more complex model by using boosting? E.g. not using independent features.

A: Yes, at least to some extent.

Assignment 6

Result of running

```
testClassifier(DecisionTreeClassifier(), dataset='iris', split=0.7)
testClassifier(BoostClassifier(DecisionTreeClassifier(), T=10), dataset='iris',split=0.7)
testClassifier(DecisionTreeClassifier(), dataset='vowel',split=0.7)
testClassifier(BoostClassifier(DecisionTreeClassifier(), T=10), dataset='vowel',split=0.7)
```

is given:

```
Trial: 0 Accuracy 95.6
Trial: 10 Accuracy 100
Trial: 20 Accuracy 91.1
Trial: 30 Accuracy 91.1
Trial: 40 Accuracy 93.3
Trial: 50 Accuracy 91.1
Trial: 60 Accuracy 88.9
Trial: 70 Accuracy 88.9
Trial: 80 Accuracy 93.3
Trial: 90 Accuracy 88.9
Final mean classification accuracy 92.4 with standard deviation 3.71
Trial: 0 Accuracy 95.6
Trial: 10 Accuracy 100
Trial: 20 Accuracy 95.6
Trial: 30 Accuracy 93.3
Trial: 40 Accuracy 93.3
Trial: 50 Accuracy 95.6
Trial: 60 Accuracy 88.9
Trial: 70 Accuracy 93.3
Trial: 80 Accuracy 93.3
Trial: 90 Accuracy 93.3
Final mean classification accuracy 94.6 with standard deviation 3.65
Trial: 0 Accuracy 63.6
Trial: 10 Accuracy 68.8
Trial: 20 Accuracy 63.6
Trial: 30 Accuracy 66.9
Trial: 40 Accuracy 59.7
Trial: 50 Accuracy 63
Trial: 60 Accuracy 59.7
Trial: 70 Accuracy 68.8
Trial: 80 Accuracy 59.7
Trial: 90 Accuracy 68.2
Final mean classification accuracy 64.1 with standard deviation 4
Trial: 0 Accuracy 85.7
Trial: 10 Accuracy 90.3
Trial: 20 Accuracy 88.3
Trial: 30 Accuracy 90.9
Trial: 40 Accuracy 84.4
Trial: 50 Accuracy 81.2
Trial: 60 Accuracy 87.7
Trial: 70 Accuracy 86.4
Trial: 80 Accuracy 87
Trial: 90 Accuracy 90.3
Final mean classification accuracy 86.7 with standard deviation 2.7
```

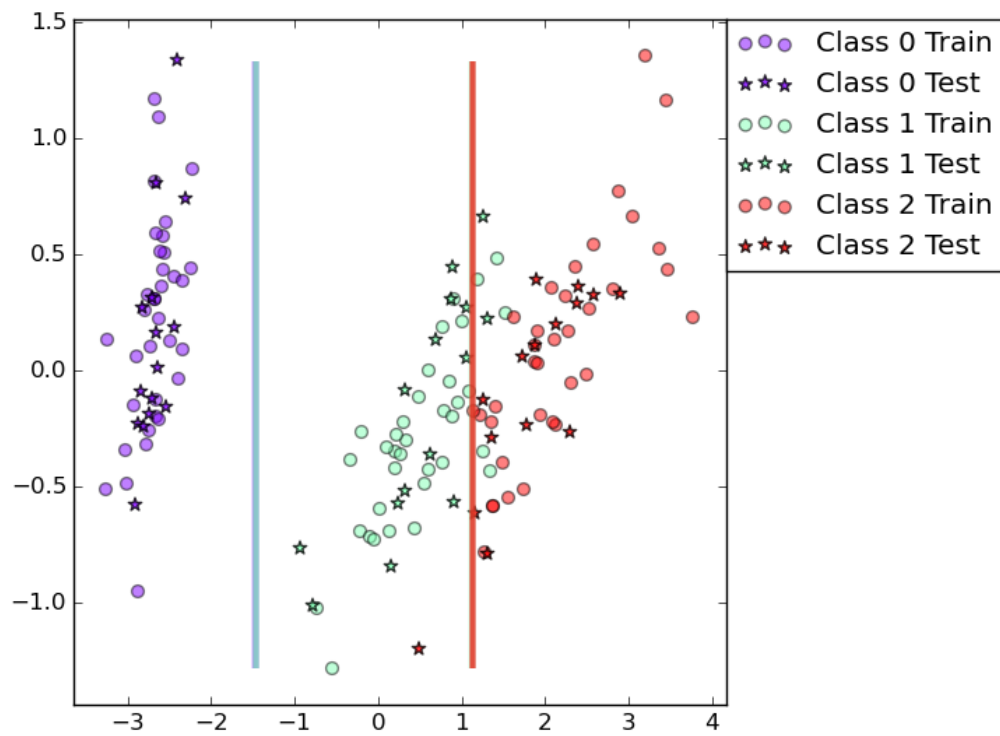


Figure 4: Decision tree.

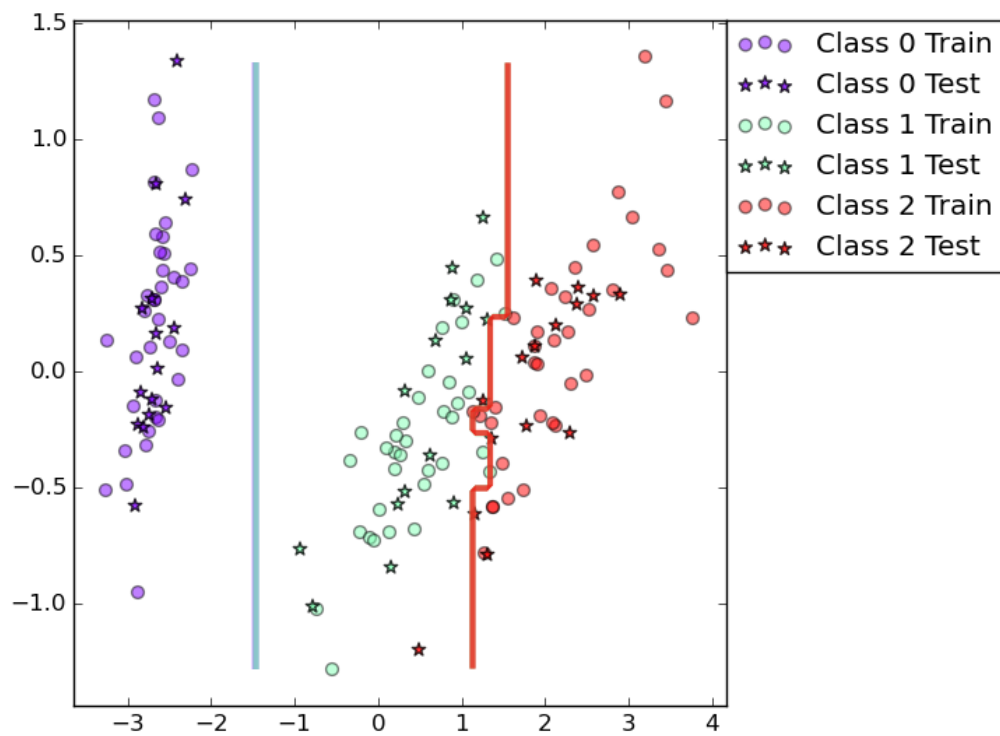


Figure 5: Boosted decision tree.

Questions:

1) Is there any improvement in classification accuracy? Why/why not?

A: Yes, in particular for the vowel data set. The mean accuracy increases from 64.1 % to 86.7 %. Not so much for the iris data set.

2) Plot the decision boundary of the boosted classifier for the iris set. What differences do you notice? Is the boundary of the boosted version more complex?

A: Boosted version is more complex. Both seems to find a boundary that runs in parallel to the axes.

3) Can we make up for not using a more complex model by using boosting?

A: Yes

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
- Irrelevant inputs: part of the feature space is irrelevant
- Predictive power
- Mixed types of data: binary, categorical or continuous features, etc.
- Scalability: the dimension of the data, D , is large or the number of instances, N , is large, or both.

We would pick boosted decision tree. It seems to take less computing resources as N and/or D grows. It has only slightly lower classification accuracy than the boosted naive Bayes classifier for the iris set. It should also work well for binary or categorical data.