Peng Ji & Zibo Meng

Oct 12, 2015

Report to assignment 3 for CS570

1. Basic requirement

Implementation: C4.5 and Naïve Bayes classifier

JDK: java version "1.8.0\_60"

Java(TM) SE Runtime Environment (build 1.8.0\_60-b27)

Java HotSpot(TM) 64-Bit Server VM (build 25.60-b23, mixed mode)

Training Dataset: mushroom.training

Test Dataset: mushroom.test

Result comparation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Training set | | Test set | |
| Our results | Weka | Our results | Weka |
| C4.5 | 100%  (cf = 0, M = 1) | 100%  (cf = 0.25, M = 2) | 100%  (cf = 0, M = 1) | 100%  (cf = 0.25, M = 2) |
| Naïve Bayes | 96.2% | 96.2% | 95.7% | 95.7% |

The C4.5 tree shows consistently higher accuracy than Naïve Bayesian. The reason of this difference is that the decision tree maximize the purity of every leaf node while Naïve Bayesian classifier allow certain impurity. Although C4.5 tree can provides higher accuracy than Naïve Bayesian classifier, the implementation of the tree is more complicated and the speed is slow.

1. Exploring Weka

To explore the classifiers in Weka, two datasets are used, IRIS and MUSHROOM

* 10-fold validation is used for accuracy testing
* With SVM,
  + the accuracy for MUSHROOM is 99.83%,
  + the accuracy for IRIS is 96.67%
* With Random Forest,
  + the accuracy for MUSHROOM is 100%,
  + the accuracy for IRIS is 95.33%
* Since both datasets are small, there is no significant difference of the speed.
* Comparing to SVM, Random Forest shows higher accuracy for MUSHROOM and lower accuracy for IRIS. The reason may be that Random Forest needs relatively more data to facilitate its power of drawing the boundary, while IRIS has only 150 records and MUSHROOM has 7423 records.
* In our case, data preprocessing might not help much, since our data are pretty clean. However, raw data in the real world can be really dirty. Noise and Outliers will largely reduce the performance and accuracy of the classifier. Comparing to SVM, Random forest may perform better considering the outliers. Therefore, data preprocessing methods, such as removing the outlier, averaging out the noise and normalization, will help classifiers improve their performance as well as accuracy.

From the implementation, it will consume more time to build a decision tree than calculating the possibility for each attribute. So C4.5 will cost much more time to build the tree. However, during the prediction, C4.5 may only need a small part of the whole set of attributes to classify the instance, while Naïve Bayesian always needs to calculate the possibility of all the attributes. In such case that there are not enough records but tons of features, C4.5 may have advantage in both performance and accuracy.

1. Exploring and implementing further

Because the mushroom data set is too clean, we tried another dataset – connect4.dat, which contains 42 features, 3 labels and 67557 instances. Training data set and test data set are separated randomly as 4:1. As this dataset is extremely ugly and difficult, we induced confidence factor in C4.5 classifier and bootstrap in those 2 classifiers. The results are as follows:

a)Classifier: C4.5

Training Set: mushroom.trn

Test Set: mushroom.test

Bootstrap: false

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CF | 0 | 0.1 | 0.15 | 0.18 | 0.2 | 0.4 |
| Accuracy | 100% | 100% | 100% | 98% | 98% | 98% |

vs. Weka

Accuracy is 100% when cf = 0.25 and M = 2

b) Classifier: C4.5

Training Set: mushroom.trn

Test Set: mushroom.test

Bootstrap: true

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CF | 0 | 0.1 | 0.15 | 0.18 | 0.2 | 0.4 |
| Accuracy | 100% | 100% | 100% | 100% | 98% | 98% |

vs. Weka

Accuracy is 100% when cf = 0.25 and M = 2

c)Classifier: Naïve Bayes

Training Set: mushroom.trn

Test Set: mushroom.test

Sampling rate: 100%

|  |  |  |
| --- | --- | --- |
| Bootstrap | false | true |
| Accuracy | 95.72% | 96.86% |

vs. Weka

|  |  |  |
| --- | --- | --- |
| Bootstrap | false | true |
| Accuracy | 95.72% | 96.43 % |

d) Classifier: C4.5

Training Set: connect4.trn

Test Set: connect4.tst

Bootstrap: false

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CF | 0 | 0.2 | 0.4 | 0.6 | 0.8 | 0.99 |
| Accuracy | 73.98% | 73.98% | 74.02% | 74.16% | 74.27% | 74.34% |

vs. Weka

Accuracy is 80.52% when cf = 0.25 and M = 2

d) Classifier: C4.5

Training Set: connect4.trn

Test Set: connect4.tst

Bootstrap: true

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CF | 0 | 0.2 | 0.4 | 0.6 | 0.8 | 0.99 |
| Accuracy | 70.78% | 70.91% | 70.85% | 71.57% | 71.97% | 71.88% |

vs. Weka

Accuracy is 77.21% when cf = 0.25 and M = 2

c)Classifier: Naïve Bayes

Training Set: connect4.trn

Test Set: connect4.tst

|  |  |  |
| --- | --- | --- |
| Bootstrap | false | true |
| Accuracy | 71.59% | 71.72% |

vs. Weka

|  |  |  |
| --- | --- | --- |
| Bootstrap | false | true |
| Accuracy | 71.59% | 71.51 % |

According to these results we list above, it’s not very helpful to induce the confidence factor and bootstrap on the decision tree of mushroom dataset, because this data is easy and clean enough. However, for the mushroom dataset, we can see the bootstrap will bring us an improvement from 95% to 96%, which is what we want to reduce the overfitting in the classifier training.

For the connect4 dataset, the results are not very good not only for the original C4.5 and Naïve Bayes classifier but also for the bootstrap and confidence factor induce. From the results, we can see very clear overfitting situation in our classifier training procedure. In the C4.5 classifier, increasing the confidence factor will cause a little bit increase of accuracy. Thus, I think bringing in the confidence factor will help to reduce the overfitting more or less. However, there is no feature in this data set, which can bring a great information gain. As a result, the decision tree is not a good solution for this dataset. In weka, its accuracy is higher about 6% than our results when its confidence factor is 0.25 and minimum number of each leaf node is 2, so I think the minimum number of each leaf node requirement might be an important method to reduce the overfitting further, which I haven’t implemented yet.

For the Naïve Bayes classifier of connect4 dataset, there is almost no difference between our results and weka’s, both the original algorithm and bootstrap one.