Lab 9: Machine Learning

Jorge García Ferreiro Pedro García Castillo

25/April/2016

Exercise 1: J48

Objetive Executes J48: (Weka version of C4.5) with drug1n.arff. Once built the decision tree, I evaluate it using 3 different methods: "training set", "cross-validation" and "supplied test set" (using as drug2n.arff set of tests). Analyzes the results. Text is coming soon.

The first step we have taken is to load the training data file "drug1n.arff". That said, we open the Classify tab and choose the classifier "J48".

Then, once generated the decision tree C4.5 we discuss how the algorithm behaves according to the evaluation methods we are using ("training set", "cross-validation" and "supplied test set").

Training set

In this example we obtain a high percentage of success. Around 97%! That's huge and shows that our system is overfitted. Because we're training and testing with the same data.

Date	Instances	Success
Correctly Classified Instances	194	97%
Incorrectly Classified Instances	6	3%

So our system train and test with same data… So, obviously it will obtain a good score because they are almost the same (so our system was modeled according to this data).

Cross-validation

In this method we train and test with same data but dividing our data in different chunks. So the system will obtain the mediam of all the obtained result. In this case, the result is less than the Training set but is also overfitted.

Date	Instances	Success
Correctly Classified Instances	185	92.5 %
Incorrectly Classified Instances	15	7.5%

Supplied test set

In this case we feed our system with a custom data and then test with another different dataset. We obtain a great mark! 90.5% of accuracy! That's a very good result with a different dataset. This means that the algorithm made a good model.

Despite this mark is lower than the previous ones, is a very good result (taking into account the dataset is different)

Date	Instances	Success
Correctly Classified Instances	362	90.5 %
Incorrectly Classified Instances	38	9.5%

Exercise 1: ZeroR

ZeroR is the simplest classification method which relies on the target and ignores all predictors. ZeroR classifier simply predicts the majority category (class). So this algorithm is not good, but is useful to compare with others algorithm

Training set

Really bad result. Just 45% of success. And 54 of failure. That means this algorithm is really bad and don't predict the correct data.

So the model it's creating is really bad. That's because it constructs a frequency table for the target and select its most frequent value.

Date	Instances	Success
Correctly Classified Instances	91	45.5%
Incorrectly Classified Instances	109	54.4%

Cross-validation

Also cross validation performs really bad. That's because the data is still the same. And the model has not changed. So we obtain the same results as with the training set (nothing changes).

Date	Instances	Success
Correctly Classified Instances	91	45.5%
Incorrectly Classified Instances	109	54.4%

Supplied test set

So in the supplied test set things get worse! And the accuracy now is lower than previous execution. This make sense, because when we use another data set in every type of program we normally obtain lower results (because sometimes is difficult to not overfitting our algorithm). In this case we got 3% worse than when cross validation.

Date	Instances	Success
Correctly Classified Instances	170	42.5%
Incorrectly Classified Instances	230	57.5%

Exercise 2: Hierarchical clustering

Training set

In this first experiment we are splitting using the file drug1n both to train and test the results so this experiment is quite prone to overfit.

We can see that by asking Weka to split it into 5 clusters we get 3 of the 5 clusters that contain most of the examples and the rest that are almost empty. Those 3 dominant clusters are the first the third and the fifth, that contain all 3 together a total of 91% of the examples.

So as we can see two of the drugs are not very needed and only a very reduced group of people uses them.

Cluster number	Instances	Success
0	91	46%
1	9	5%
2	54	27%
3	7	4%
4	39	20%

Supplied test set

This second experiment is made by using drug1n as training set and then applying that cluster model to the drug2n dataset as test set, so this experiment is much less prone to overfit than the previous one both because it doesn't use the same dataset both for train and test and because it has a bigger test set.

But the percentages obtained when splitting the drug2n in 5 clusters is quite similar to the previous one because the first, third and fifth cluster contain together a total of 88% of the examples. This shows once more that those are the most common drugs to treat the patients in most of the cases.

Cluster number	Instances	Success
0	170	43%
1	22	6%
2	117	29%
3	25	6%
4	66	17%

Percentage split

This third experiment is made by using 66% of the examples in the file drug1n as training set and the other 33% percent of the examples as test set. This method is quite usefull when we don't have 2 different files to train and test which is quite common because is a way to simulate it.

In this example we can see that the first cluster contains more than half of the examples in the test set (57%) and the second one a 29% so we can see that most of the examples in the test set could be classified in those two clusters, the last 3 clusters contain respectively 7%, 5% and 1% of the examples so we can tell that does 3 drugs are not very used in the examples in this test set.

At the same time we can see that the percentages are very different to the previous 2 experiments so we can think that this may be caused because the test set is way too small, just 68 examples and this may cause that we don't have the same distribution as in the 2 previous ones.

Cluster number	Instances	Success
0	39	57%
1	20	29%
2	5	7%
3	3	4%
4	1	1%

Exercise 3: J48

Training set

This example suffer from overfitting as we are using the same dataset for train and test and we can see it because we are getting a 100% correct classification and has 1,0 both precission and recall which is really strange and usually tells us that our model is overfitting.

Date	Instances	Success
Correctly Classified Instances	200	100%
Incorrectly Classified Instances	0	0%

Cross-validation

This new example using the fold 10 over the training set shows seems also to be overfitting as the previous one because from the 200 examples it classifies correctly 198 of them which means 99% and as can be seen in the table it also has a 0,99 both for precision and recall which means that the classifier is classifying almost perfectly all the examples. The precision means of all the example classified in a class how many of them do really belong to that class. The recall measures of all the example that belongs to a class how many of them are correctly classified as belonging to it.

Date	Instances	Success
Correctly Classified Instances	198	99%
Incorrectly Classified Instances	2	1%

Supplied test set

This method can not be done because our data set contains 8 features and the train set 7. So there is no possible

Exercise 3: ZeroR

Training set

We can observe in this case that the ZeroR algorithm works quite badly when using the same data set both for training and testing, it only classifies a 45.5% of the examples correctly. The bad rate of classification is also shown on the precision and recall parameters. We get a 0.207 and 0.455 which means that most of the examples are badly classified in the class that it's not their class.

Date	Instances	Success
Correctly Classified Instances	91	45.5%
Incorrectly Classified Instances	109	54.5%

Cross-validation

The result in this case is just about the same as in the previous one.

Date	Instances	Success
Correctly Classified Instances	91	45.5%
Incorrectly Classified Instances	109	54.5%

Exercise 3: Hierarchical clustering

Training set

Thanks to the new grouping of some features we have improve our model to obtain a better prediction. So in the first cluster (corresponding to the new group) we group the 73% of the samples. So that means our assumption about the correlation with K/Na is simetric to Na/K. However in this method there is an obvious overfitting.

Cluster number	Instances	Success
0	146	73%
1	54	27%

If we re-run the same code but with 3 clusters. We see that the result has not changed a lot. So the 0 cluster only decrease in 8%. So that means, there is a group well defined thanks to the pre-filter we did.

Cluster number	Instances	Success
0	130	65%
1	16	8%
2	54	27%

Supplied test set

This method can not be done because our data set contains 8 features and the train set 7. So there is no possible

Percentage split

Here. The more we split, the worse result we obtain. When we split less percentage we obtain better results. That's obvious, because we're training/testing with few data, so there are more probabilities to obtain a good mark. However, if we split 60% of our data, we obtain some good results (less overfitted than with the own data set testing).

60% split:

Cluster number	Instances	Success
0	55	69%
1	25	31%

Dividing into 3 clusters and with this method. We obtain a worse Percentage than the training set. This is normal, because in the training set we're overfitting. In this case, we

Cluster number	Instances	Success
0	49	61%
1	25	31%
2	6	8%