Project Milestone 3

Golden Experience

Mo Chen

1.1 **Introduction**

The classifiers I used in the project are: J48 decision tree, JRip rule learner and Dagging Meta classifier. J48 is a version of an earlier algorithm developed by J.Ross Quinlan, the very popular C4.5. Decision trees are a classic way to represent information from a machine learning algorithm. JRip is a propositional rule learner, repeated incremental pruning to produce error reduction (RIPPER), which was proposed by William W. Cohen as an optimized version of IREP. Dagging is a Meta classifier that creats a number of disjoint, stratified folds out of the data and feeds each chunk of data to a copy of the supplied base classifier.

1.2 **Parameter Variation**

For J48 decision tree, there are several parameters and options that I can vary to examine the robustness of the model: (1) confidence factor C, (2) minimum number of objects M, (3) the option of using binary splits or not and (4) reduced error pruning option. These four parameters would affect the final robustness of the model after testing. By using CVParameter function in Weka, we can test the optimal parameter for both C and M, and after that we can set the Boolean number for the options (3) and (4). By varying the parameter C and M, the learning time would increase because we are doing a cross-validation for the parameter, and sometimes the optimal C and M tends to overfit the model (the optimal parameter would give worse results than the default setting). For some testing dataset, changing the (3) and (4) options directly would be very effective and reduced the error rate by 20-30%. And for some cases, changing parameter C and M would not render better results than the default settings.

For JRip decision tree, the parameter we can adjust from the model seems to be more than J48 tree. I have (1) number of folds F, (2) minimum number of object N, (3) optimization O and (4) seed S. However, there are no additional Boolean options in the model that we can vary. And by adopting CVParameter function again, we can see that JRip is very insensitive to the parameter change on test datasets. For most of the cases, the variation in parameters didn't change the final results and even if it did, the change never exceeds 2-3%. From the experiment, I observed that the learning time for this model is the longest among the all three because if you increase the number of folds and optimization O and N to a fairly large number, the learning time would increase exponentially.

For dagging, since it is a Meta classifier with a base classifier, there are a lot of parameters that we can vary: (1) number of folds F, (2) seed S and (3) the base classifier. Actually we can even change the parameter for the base classifier, but after the experiment I think that the change in the parameter of base classifier would give insensitive results to the testing dataset. On the

other hand, if we change the base classifier directly, the results would be very sensitive. Since it is a Meta classifier, dagging is not very sensitive to the parameter change of number of folds, but very sensitive the change in base classifier models.

1.3 **Final settings**

For J48 trees, I tune the parameter C from 0.01 to 0.5 with 50 steps and M from 1 to 10 in 10 steps, and I'll test the final results by turning the Boolean options on and off.

For JRip trees, all parameter are set to range from 1 to 10 with 10 steps and search for optimal parameters

For Dagging, I'll switch the base classifier in JAVA manually, and use CVParameter function to search for other parameters (F and S).

The tunings for all three classifiers are recorded in the experiment log.

1.4 **Summary**

Among the classifier I used, the best robustness was obtained by dagging, which is not the most robust model without tuning (J48 is the most robust model without tuning). However, by changing the base classifier in dagging, the results can be very flexible and this Meta classifier can handle different kinds of dataset very well. The average robustness from milestone 2 was 0.71 (best classifier for the dataset without tuning), and the average robustness from dagging with tuning was 0.64, which showed huge improvement. The single Meta classifier can achieve greater robustness than three classifiers combined. Therefore, I think dagging could have great potential in dealing with different kinds of dataset.

Experiment Log

J48

Default setting C=0.25, M=2, reduced error prune=false, binary splits=false, Laplace=false

1. J48 on anneal, slightly performance increase c=0.087 and M=1 with other options set to be true
2. Audiology performance can be improved by 0.06 with c=0.2 and M=1,
3. I cannot improve the accuracy with tuning the parameters using CVParameter function, but we can change the other options to improve accuracy (binary split, reduced error pruning)
4. 0.3446601941747573 cannot improve the accuracy no matter how
5. Breast cancer: this can be improved by setting the parameter C to 0.01 and M to 7, the overall accuracy improved 0.03
6. Colic: No improvement by tuning parameters, the accuracy remain the same
7. Credit-a: The accuracy can be improved by setting C to 0.14 and M to 10
8. C=0.23, M=7, 0.02 improvements, it can be further improve by setting the options, binary splits = true and reduced error prune=true
9. Glass: No improvements by tuning parameters
10. Can be improved by setting M to 1, further improvements can be made by setting the binary split option true.
11. Same improvements can be obtained by tuning M or C individually, By setting M to 10 and C remain unchanged, the accuracy improves by 0.02. Further improvements can be made by setting the binary split and reduced error prune option true.
12. Original Accuracy is already very high, tuning parameters will only get worse results

JRip

Default setting F=3, N=2, O=2, S=1, Default pruning=true

1. Anneal: the accuracy can be improved by tuning parameter to Folds=6, F=5, N=1, O=2 and S=3. Accuracy improved by 0.01
2. Tuning parameter will make the accuracy worse than default settings

3. Tuning all parameters would be very expensive (setting all four parameter from 1-10, the runtime will be ~10 minutes) and time consuming, and the tuning will actually lower the accuracy
4. Cannot be improved
5. The accuracy can be improved by 0.04 by setting Folds=6, F=5, N=4, O=1, and S=3.
6. Cannot be improved
7. Cannot be improved
8. Can be improved by either tuning F to 1 or N to 1, the accuracy improved by 0.02
9. Cannot be improved
10. Cannot be improved
11. Original accuracy very high, cannot be improved
12. The accuracy can be slightly by 0.001 by the following settings: Folds=6, F=5, N=3, O=4 and S=1

Dagging

Default setting number of folds F=10, seed S=1, default classifier SMO

1. Accuracy can be improved by 0.05 by setting F=1
2. Accuracy can be greatly improved by ~0.3 by setting F=1
3. Can be improved by changing the base classifier for dagging to JRip and set the F=5
4. Can be improved by setting F=5, S=10
5. Can be improved by changing the base classifier for dagging to JRip and set F=6, S=7
6. Can be improved by changing the base classifier to NBTree and set F=4 and S=9
7. Can be improved by setting F=5 and S=7
8. Can be improved by setting F=3 and S=4
9. Can be improved greatly by changing the base classifier to J48 Tree and set F=5, S=10
10. Can be improved by changing the base classifier to NBTree and set F=4, S=6
11. Can be improved by changing the base classifier to NBTree and set F=1, S=1
12. Can be improved by changing the base classifier to NBTree and set F=1, S=1