Phase 3 of Project for Machine Learning 10601

Name: Zhiyu Li, Andrew ID: zhiyul
Name: TianminZou, Andrew ID: tzou

Section 1 Tuning the FT classifier learner.

Section 1.1

Normal Name: Functional trees
Weka Class Name: FT
What it does: (from the weka website) Function Trees are classification trees that could have logistic regression functions at the inner nodes and/or leaves.
Where it is described:
Joao Gama (2004). Functional Trees.
NielsLandwehr, Mark Hall, Eibe Frank (2005). Logistic Model Trees.

Section 1.2 Parameters we varied

The parameters we tuned are shown below. Following information are from the weka website:

-B Binary splits (convert nominal attributes to binary ones)

-P Use error on probabilities instead of misclassification error for stopping criterion of LogitBoost.

-I <numIterations> Set fixed number of iterations for LogitBoost (instead of using cross-validation)Larger value will increase the learning time.

-F <modelType> SetFuntional Tree type to be generate: 0 for FT, 1 for FTLeaves and 2 for FTInner

-M <numInstances> Set minimum number of instances at which a node can be split (default 15)This will affect the over-fitting feature

-A The AIC is used to choose the best iteration.

We did not tune the –W option because it is not suitable in the datasets.

When trying to find the best parameter settings for a given dataset, we use a greedy algorithm. That is, we try to find only one parameter setting at a time. Once we get that option's best parameter, we fix that and find next option's best parameter.

For each of the 12 dataset, we choose those parameters dynamically so that we get different best parameter settings for different dataset.This hopefully would do better than using fixed values for all the datasets as Professor Cohen said in piazza @541. And the reason is simply that we choose the best parameters for each dataset.

Section 1.3 Final Settings.

As discussed in the previous subsection, there is not final settings for this classifier as a whole, but we will show different settings for each dataset here:

Dataset	Settings	E(c)/E(NB)	Time (s)
anneal	[-B, -P, -I, 5, -F, 0, -M, 10, -A]	0.35294117647058826	0.221255369
audiology	[-B, -P, -I, 13, -F, 0, -M, 10, -A]	0.6363636363636364	0.302834919
autos	[-B, -P, -I, 14, -F, 1, -M, 10, -A]	0.7894736842105263	0.126959202
balance-scale	[-B, -P, -I, 5, -F, 0, -M, 11, -A]	0.3787878787878788	0.026897839
breast-cancer	[-P, -I, 9, -F, 0, -M, 14, -A]	1.0	0.026536926
colic	[-B, -P, -I, 11, -F, 1, -M, 13, -A]	0.782608695652174	0.059477916
credit-a	[-B, -P, -I, 6, -F, 0, -M, 14, -A]	0.6428571428571429	0.074299716
diabetes	[-B, -P, -I, 11, -F, 1, -M, 15, -A]	1.1129032258064515	0.035022226
glass	[-B, -P, -I, 10, -F, 0, -M, 10, -A]	0.588235294117647	0.033246312
heart-c	[-B, -P, -I, 9, -F, 0, -M, 11, -A]	1.0555555555555556	0.046142531
hepatitis	[-B, -P, -I, 7, -F, 0, -M, 13, -A]	0.8333333333333334	0.007760908
hypothyroid	[-B, -P, -I, 7, -F, 1, -M, 10, -A]	0.08219178082191782	0.536046166

Avg E(c)/E(NB): 0.687937616998071
Max E(c)/E(NB): 1.1129032258064515

Section 2 Tuning the Dagging classifier learner.

Section 2.1

Normal Name: Dagged Models
Weka Class Name: Dagging
What it does: (from the weka website) This meta classifier creates a number of disjoint, stratified folds out of the data and feeds each chunk of data to a copy of the supplied base classifier. Predictions are made via majority vote, since all the generated base classifiers are put into the Vote meta classifier.
Where it is described:
Ting, K. M., Witten, I. H.: Stacking Bagged and Dagged Models. In: Fourteenth international Conference on Machine Learning, San Francisco, CA, 367-375, 1997.

Section 2.2 Parameter we varied

There is only one parameter we tuned for this classifier. Following information is from the weka website:

-F <folds> The number of folds for splitting the training set into smaller chunks for the base classifier. (default 10)Larger value will increase the learning time.

We did not use the –S option because we do not think tuning the classifier by a random seed is a good choice. And we kept the base classifier as SMO because each base classifier has its own parameters and it is too expensive to tune all of them.

Also, as discussed before, we use different settings for each dataset.

Section 2.3 Final settings

As we did before, we will show the settings for different dataset:

Dataset	Settings	E(c)/E(NB)	Time (s)
anneal	-F 1	1.5294117647058822	0.115807681
audiology	-F 1	1.5454545454545454	0.506149415
autos	-F 1	0.8684210526315789	0.135783179
balance-scale	-F 5	0.48484848484848475	0.035171466
breast-cancer	-F 8	0.9310344827586208	0.018053764
colic	-F 7	1.0	0.019880565
credit-a	-F 7	0.6666666666666666	0.024809334
diabetes	-F 1	1.1290322580645162	0.014331745
glass	-F 6	0.7450980392156862	0.133600597
heart-c	-F 2	1.0	0.027925414
hepatitis	-F 1	1.0	0.013234895
hypothyroid	-F 1	1.3561643835616437	0.146148731

Avg E(c)/E(NB): 1.021344306492302
Max E(c)/E(NB): 1.5454545454545454

Section 3 Tuning the Multi Layer Perceptron.

Section 3.1

Normal Name: Multilayer Perceptron (MLP)
Weka Class Name: MultilayerPerceptron
What it does: (from the weka website) A Classifier that uses backpropagation to classify instances.
This classifier is a artificial neural network with multiple layers of perceptrons and we can adjust the parameters of the classifier.

Section 3.2 Parameters we varied

The parameters we tuned are shown below. Following information are from the weka website:

-L <learning rate>
Learning Rate for the backpropagation algorithm.

(Value should be between 0 - 1, Default = 0.3).

-M <momentum>

Momentum Rate for the backpropagation algorithm.

(Value should be between 0 - 1, Default = 0.2).

-V <percentage size of validation set>

Percentage size of validation set to use to terminate

training (if this is non zero it can pre-empt num of epochs.

(Value should be between 0 - 100, Default = 0).

-C

Normalizing a numeric class will NOT be done.

(Set this to not normalize the class if it's numeric).

-I

Normalizing the attributes will NOT be done.

(Set this to not normalize the attributes)

-D

Learning rate decay will occur.

(Set this to cause the learning rate to decay).

In order to test the results of these parameters, we played with these parameters and the results are shown below.

Dataset	L	M	V	C	I	D	E(c)/E(NB)	Time (s)
anneal	6	6	0	C	NO	NO	0.411765	19.666516571
audiology	5	1	0	C	I	NO	0.545455	7.505280915
autos	4	4	0	C	NO	NO	0.815789	3.081130546
balance-scale	3	2	0	C	NO	NO	0.318182	0.212497675
breast-cancer	1	5	2	C	NO	NO	0.862069	0.106782296
colic	5	7	3	C	NO	D	0.956522	0.158490324
credit-a	2	9	3	C	NO	D	0.678571	0.171744462
diabetes	3	1	7	C	NO	D	1.548387	0.284559218
glass	3	3	0	C	NO	NO	0.568627	0.245072621
heart-c	1	5	4	C	NO	D	0.944444	0.602113734
hepatitis	8	4	5	C	NO	D	0.833333	0.012941974
hypothyroid	1	1	1	C	NO	NO	1.09589	13.487080904

Avg E(c)/E(NB): 0.798252951076108

Max E(c)/E(NB): 1.54838709677419

In this part we also use greedy algorithm to find the best parameters for each of the options we are to test. And after getting all of these parameters optimized, we will test the results using this result.

From the results, we can see that the parameter C should always be set. Also relatively smaller values for L and V seems to improve the result accuracy. However, smaller values for L and M may have negative influence on the efficiency of this program, but generally the performance does not vary a lot.

For each of the 12 dataset, we choose a different set of parameters, as is said in former sections. In this way we will be able to use the best classifier for all of the datasets.

Section 3.3 Final Settings.

As discussed in the previous subsection, there is not final settings for this classifier.

Section 4 Summary.

From the results above, we can see that FT is the best classifier because both the average and maximal scores for E(c)/E(NB) are the lowest than the other two classifier. Also this algorithm is very efficient because the time needed for training and evaluating is shorter than the other two classifiers. Although for some of the dataset, the score for FT is a little lower than another classifier, we still think it is the best and the most robust classifier.