10601 Class Project: Milestone 3

Qichen Pan (AndrewId: pqichen)

Haodong Liu (AndrewId: Haodong Liu)

Did you receive any help whatsoever from anyone in solving this assignment?

No.

Did you give any help whatsoever to anyone in solving this assignment?

No.

Background

For classifier LMT:

Logistic model tree (LMT)[1] is a classification model with an associated supervised training algorithm that combines logistic regression (LR) and decision tree learning. The algorithm can deal with binary and multi-class target variables, numeric and nominal attributes and missing values. The parameters we used are –I[2] (set fixed number of iterations for LogitBoost) and –W (set beta for weight trimming for LogitBoost). We used CVParameterSelection to select the best parameters automatically. The parameter of CVParameterSelection was set as "-I -1 14 4" and "–W 0.0 0.5 6".

For classifier ADTree:

An alternating decision tree [3](ADTree) is a machine learning method for classification. It generalizes decision trees and has connections to boosting. The parameters we used are B[4] (the number of boosting iterations) and E (Set the nodes to expand: -3(all), -2(weight), -1(z_pure), >=0 seed for random walk). We used CVParameterSelection to select the best parameters automatically. The parameter of CVParameterSelection was set as "B 10 20 3" and "E -3 -1 3".

For classifier Dagging:

This meta-classifier creates a number of disjoint[5], 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. The parameters we used are C (the size of the cache, 0 for full cache and -1 to turn it off) and N (whether to 0=normalize/ 1=standardize / 2=neither). We used CVParameterSelection to select the best parameters automatically. The parameter of CVParameterSelection was set as “C 1.0 5.0 5” and “N 0.0 2.0 3”

Optimization Details

We chose LMT as the final classifier for that it can give a most robust model among all 3 classifiers. The options we tuned for each case is shown below:

Dataset	Parameters	Time consumption (Seconds)
anneal	"-I -1 -M 15 -W 0.0"	24.09
audiology	"-I 15 -M 15 -W 0.1"	1.61
autos	"-I 30 -M 15 -W 0.0"	3.79
balance-scale	"-I 30 -M 15 -W 0.2"	0.87
breast-cancer	"-I -1 -M 15 -W 0.2"	0.42
colic	"-I 15 -M 15 -W 0.0"	0.98
credit-a	"-I -1 -M 15 -W 0.2"	1.48
diabetes	"-I 15 -M 15 -W 0.2"	0.66
glass	"-I -1 -M 15 -W 0.0"	0.58
heart-c	"-I -1 -M 15 -W 0.0"	1.51
Hepatitis	"-I -1 -M 15 -W 0.2"	0.25
Hypothyroid	"-I -4 -M 15 -W 0.0"	101.79

Table 2.1 Parameters and Time consumptions for LMT

For LMT, we mainly tried to tune two parameters: I and W. ‘I’ stands for iteration times of LogitBoost and ‘W’ stands for trimming weight. From table 2.1, we can observe that the time consumption on building a model is mainly decided by the size of dataset. The largest training dataset ‘hypothyroid’ has the longest training time.

We used the following metric for performance.

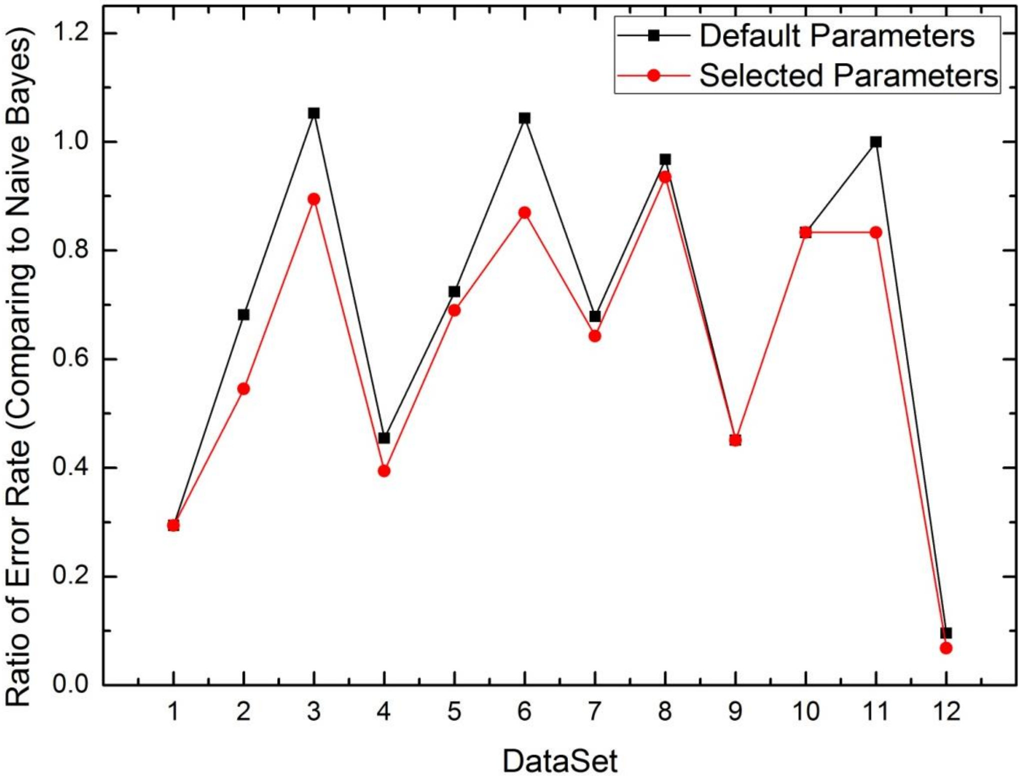
$$\text{Performance} = \frac{\text{ErrorRate[LMT]}}{\text{ErrorRate[Naive Bayes]}}$$

Here is another table showing the improvement of performance by tuning the parameters:

Dataset	Performance with Default Parameters	Performance with Tuned Parameters	Improvement of Performance
anneal	0.29411765	0.294119529	0.00%
audiology	0.68181818	0.545454545	20.00%
autos	1.05263158	0.894736842	15.00%
balance-scale	0.45454545	0.393940667	13.33%
breast-cancer	0.72413793	0.68965531	4.76%
colic	1.04347826	0.86956387	16.67%
credit-a	0.67857143	0.642856857	5.26%
diabetes	0.96774194	0.935483823	3.33%
glass	0.45098039	0.450979471	0.00%
heart-c	0.83333333	0.833333333	0.00%
hepatitis	1	0.8333315	16.67%
hypothyroid	0.09589041	0.068492055	28.57%
Average	0.68977055	0.62099565	10.30%

Table 2.2 Improvement of Performance on 12 datasets

Here is a graph showing the improvement in a more intuitive way:



Graph 2.1 Improvement of Performance on 12 datasets

We can observe that in some datasets such as “anneal”, “glass” and “heart-c”, the default parameters turn out to be the best parameters. So in these cases, there are no improvements are 0.0%. In some datasets such as “hepatitis”, we increase trimming weight from 0.0 to 0.2 to overcome over-fitting problem, so the performance gains a boost. In some case such as “hyphyroid”, we reduce the iteration times to 15 or less to reduce training time. In this way we can gain the golden point on this trade-off.

[1]. Logistic Modeling Tree. http://en.wikipedia.org/wiki/Logistic_model_tree

[2]. LMT in Weka. http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/LMT.html

[3]. Alternating decision tree. http://en.wikipedia.org/wiki/Alternating_decision_tree

[4]. ADTree in Weka. http://bio.informatics.indiana.edu/ml_docs/weka/weka.classifiers.adtree.ADTree.html

[5]. Dagging in Weka. http://weka.sourceforge.net/doc.packages/dagging/weka/classifiers/meta/Dagging.html