Nick Blazier

NBlazier

Nikolai Mushegian

Nmushegi

Milestone 3 report

We did not receive help from anyone on this assignment.

We did not give help to anyone on this assignment.

Section 1.1

Real name: Functional Trees

WEKA name: FT

Functional Trees are classification trees that have logistic regression functions at their inner nodes and leaves. Functional Trees were introduced by João Gama and can be found in the 2004 Springer publication of "Machine Learning". They are also discussed in the 2001 Springer publication of "Advances in Intelligent Data Analysis".

Section 1.2

Changing the –F option changes how the algorithm uses logistic regression. Setting a value of 0 Sets it to the default Functional Trees algorithm. Using a value of 1 uses logistic regression only on the leaves. Using a value of 2 uses logistic regression only on the inner nodes. Setting the parameter to 1 (using logistic regression only on the leaves) provides the best results, improving both the average and maximum error rates. Changing this can affect the learning time of the algorithm by adding complexity.

Changing the –M option changes the minimum number of instances at which a node can be split. This parameter can make the classifier much larger by increasing the number of splits. It could also lead to over-fitting if you split perfectly onto your training data. Having increased splits can also increase increase learning time if the size of the tree increases. When tested from 1 to 30 the best value is 12, although results change little between 10-15.

The option –W changes the beta used for trimming. This value will be used to trim the tree down so it doesn't get too large. It can affect the size of the classifier which in turn effects the learning time as well.

We tested from .01 to 1 and found the best value to be .06. When set to this value both the average and maximum error rates decreased.

Section 1.3

By combining all of these optimal parameters: using linear regression on only the leaf nodes, a minimum of 12 instances for splitting nodes, and a trimming beta of .06 is the optimal version, you obtain an optimal Functional Tree. The total training time for all 12 datasets was 1244 milliseconds, with an average of 112 milliseconds per dataset.

Section 2.1

Real name: Dagging, Sequential Minimal Optimization

Weka name: Dagging, SMO

This is a higher-level classifier which works by splitting the training data into disjoint stratified folds, training a collection of internal classifiers, and then using a vote from each of the sub-classifiers. The internal classifier is SMO, for which we kept the default parameters. The Dagging classifier. For a more thorough description of this classifier, see: 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

Changing the -N option allows us to choose whether data is normalized, standardized, or neither. The best option ended up being standardization, which means that the data was preprocessed by transforming it to have mean 0 and variance 1 (which works well if you assume the data is more normally distributed). The other option, normalization, just normalizes all the points so they fit into the same unit interval.

Changing the -F option changes the number of folds the data is split into before being fed to the base classifier. Lowering this option helped with our error rates, which suggests that our data sets simply weren't big enough to train the sub-classifiers well when only 10% of the data was used.

Changing the -C option changes the complexity constant in the algorithm. Experimentally the best complexity value was 2.75.

Section 2.3

It appears that, when using the default base classifier, the best improvements were gained when the number of folds was decreased to 3, the complexity was increased to 2.75, and the data was standardized rather than normalized or left alone. This reduced our average error by about 0.17 and our maximum error by about 0.33 compared to the baseline. The total time for all 12 datasets was 2429 milliseconds, with an average of 202 milliseconds per dataset.

Section 3.1

Real name: Bayesian Logistic Regression

Weka Name: BayesianLogisticRegression

This is a classic classifier which is used to learn functions from arbitrary feature vectors to a binary class. It is described in many places, but page 7 of the ML book is a good place to start: http://www.cs.cmu.edu/~tom/mlbook/NbayesLogReg.pdf.

Section 3.2

The -P flag sets the distribution of the priors (Gaussian vs Laplacian).

The -N flag determines whether to normalize the data before training.

The -S flag sets the threshold value for determining which class a point belongs in.

The -H flag (and -V helper flag) are used to determined the hyperparameter selection method (and value if given directly). This is used to tune the regression algorithm, especially for data that is not normalized.

Section 3.3

We could not figure out how to actually tweak parameters through the MultiClassClassifier. We tried using a MultiClassClassifier with a FilteredClassifier but could not produce results.

Section 4

Among the classifiers we found Functional trees to be the most robust. They were already more robust than the other choices before parameter tuning, but achieved even better results afterwards. After changing the algorithm to apply logistic regression only on leaves, setting the minimum number of instances required to split the tree to 12, and setting the beta for trimming to .06 we were able to decrease the average robustness from .80351 to .66276 and decrease the maximum robustness from 1.34783 to 1.09677. We were able to improve the performance of the Dagging algorithm as well, but it did not perform as highly as Functional trees. We improved the average robustness of Dagging from 1.02134 to .84153, and the maximum robustness from 1.54545 to 1.21739