Empirical comparison of one-versus-all and multiclass supervised learning approaches

Case Study: Random Forests

Michael Hatch

A09910883

Abstract

The goal of this project was to investigate the efficacy of multiclass categorization methods using machine learning. Toward this goal, the UCI letter recognition dataset was considered as an example of a multiclass machine learning problem. The letter dataset was considered holistically as a multiclass (26 class) dataset, and also examined in 26 one-vs.-all cases (ie ‘A’ vs. all, ‘B’ vs. all, etc). 10-fold cross validation was used to select the optimal number of trees for the forest. Classifiers were trained, one forest for each (26) one-vs-all binary letter task, and one classifier was trained for the multiclass case (27 classifiers total). Total error in the one-vs-all and multiclass paradigms was used to evaluate their comparative accuracy (more on this decision later).

It was found that one-vs-all can perform roughly comparably to multiclass classification on this particular dataset. However, due to the widespread availability of open-source multiclass algorithms their lower training times,, and their higher accuracy, multiclass approaches seem superior for the type of problem discussed.

Methods

The letter recognition dataset presents 16 features of approximately 20,000 examples of letters, each of which is a member of one of 26 classes (the letter itself, ie “Q”, “M”, etc). For the multiclass approach, the string value of the letter was converted to an integer value representing a numerical label -- 1 for “A”, 2 for “B”, and so on. The dataset was then sent through 10-fold cross-validation to determine the optimal number of trees for the Random Forest to use (in the interest of training time, a range of 1 to 50 trees for each forest was tested). A new Random Forest model was then created with the optimal number of trees, trained on 80% of the dataset, and tested on 20% of the data (set was randomly shuffled and the shuffled set was split sequentially).

For the one-vs-all paradigm, each letter was treated sequentially as the target label, a model was trained with 10-fold cross-validation to find the optimal number of trees for identifying each letter. In other words, for the first case the letter ‘A’ was assigned class 1, and all other labels given class -1 and cross-validation was performed; then the letter ‘B’ was assigned class 1 and all others class -1 and cross-validation was performed; and so on for all labels. The optimal number of trees for classifying each letter was stored, and new random forests were trained for each letter and its corresponding optimal number of trees. Each of these were trained on 80% of the dataset and tested on 20% as above.

Results

As discussed above, total error is used as the accuracy metric for this experiment. Each 1-vs-all classifier has a much easier task (identifying one letter) than the multiclass classifier. Because of this, each classifier gets on the order of 99.95% accuracy for it’s test set. However, since we have 26 classifiers competing against one, we need a measure that aggregates errors across all of the classifiers in a manner comparable to the multiclass classifier, hence total error.

Results can be found below. Included are the paradigm, optimal number of trees, total error, a measure representing a modified testing accuracy (1 minus total error divided by number of points in the test set.), and raw testing accuracy (or average raw where applicable) for reference.

Paradigm	Optimal number of trees	Total Error	Modified testing accuracy	Raw testing accuracy
Multiclass	45	153	96.18%	96.17%
One-vs-All	Range: 15 to 49 Avg: ~34	502	87.45%	Avg: 99.52% Max:99.92%

Discussion and conclusions

It's fairly clear that for the general task of letter recognition, the multiclass approach is superior when using Random Forests. Not only is the accuracy better, training and testing together took a grand total of over 20 hours of computation for the one-vs-all cases (although to be fair, about 30% of this time was re-running code due to programmer errors), whereas end to end the multiclass experiment took less than an hour of computation. Since open-source implementations of multiclass classifiers are widely available, it seems that in most cases there is little reason to use one-vs-all approaches.

This doesn't mean that one-vs-all is entirely obsolete, however. One-vs-all classifiers performed much better than the mutliclass Random Forest for identifying individual letters. If a particular letter recognition task can be broken down into identification of a small number of letters (for example, if analysis of a linguistics experiment can be boiled down to identification of vowels), it may be more accurate and possibly faster to use one-vs-all.

Appendix:

Code can be found here:
https://github.com/TrackAddict/Cogs185/blob/master/Homework%201.ipynb

Note I will not be sharing this link and do not anticipate much, if any, traffic to this particular repo outside of the grader. However, if this solution being (technically) public is problematic in any way, please let me know and I'll take it down immediately.