CSC 529 Final Project Proposal

Matt Triano, Online Student

In the course assignments so far, I've been able to quickly generate strong learners with $> 90\%$ accuracy fairly quickly using sklearn's GridSearchCV to identify approximately optimal model parameters. I would like to explore a larger dataset and I would like to explore a much noisier problem space, so that I can explore weaker learners and compare the performance of ensembles of several different types of classifiers with different diversity generation strategies and different aggregation strategies.

Base Classifiers

I'd like to compare and contrast the performance of ensembles of classifiers built from [logistic regression[1], KNN, RandomForest[2], BAgging, flavors of AdaBoost, gradient boosting, stacking, SVM, and/or possibly neural network[3]] algorithms over some of the credit-worthiness data sets below. I will mainly use Python's scikit-learn package for building classifiers, but if I need more control, I'll implement algorithms by hand. If time allows, I'll try to use TensorFlow for to build neural network classifiers, but I have no experience to date with TensorFlow so I can't promise that I'll be able to include neural network classifiers although I have a popular book[3] and a white paper from the designers are Google Research[4] on the package to help.

Sampling

I also want to investigate whether a statistically significant improvement/difference in performence can be achieved through using SMOTE or stratification versus a simple holdout proceedure.

Recombining/aggregation

If appropriate, I'll examine classifier aggregation strategies beyond simple majority voting. For examble, obviously more sophisticated weighting methods will be used for AdaBoost (and variants). I'm also interested in evaluating stacking[5].

Data

The top data set consists of 24 attributes of 30,000 Taiwanese credit card holders and the labeled target data consists of the binary categories [1: defaulted on payment, 0: did not default on payment]. I included this dataset to attempt to apply our classification methods to a large dataset.

The second data set is a much smaller data set that consists of 20 attributes of 1000 German credit-seekers and the labeled target data consists of the binary categories [1: low risk, 2: high risk].

Ideally, I will just use one or both of the two data sets above, but if I run into implementation problems, I'll explore the third and 4th data sets which focus on mortgage default rates.

Possible Datasets

- https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients# (https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients#)
- https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data (https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data))
- http://www.creditriskanalytics.net/datasets-private.html (http://www.creditriskanalytics.net/datasets-private.html)
- https://www.fhfa.gov/DataTools/Downloads/Pages/Public-Use-Databases.aspx (https://www.fhfa.gov/DataTools/Downloads/Pages/Public-Use-Databases.aspx)

References

- 1) Dreiseitl, Stephan, and Lucila Ohno-Machado. “Logistic Regression and Artificial Neural Network Classification Models: a Methodology Review.” Journal of Biomedical Informatics, vol. 35, no. 5-6, Oct. 2002, pp. 352–359.
- 2) Breiman, Leo. “Random Forests.” Machine Learning, vol. 45, 11 Apr. 2001, pp. 5–32.
- 3) Géron, Aurélien. Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. Beijing: OReilly, 2017.
- 4) Abadi, Martin. “TensorFlow: A System for Large-Scale Machine Learning.” USENIX Symposium on Operating Systems Design and Implementation, vol. 12, 2016, research.google.com/pubs/pub45381.html.
- 5) Dzerosky, Saso. "Is Combining Classifiers with Stacking Better than Selecting the Best One?" Machine Learning 54 (April 24, 2003): 255-73.