Creating Ensembles of Classifiers

Nitesh Chawla, Steven Eschrich, and Lawrence O. Hall
Department of Computer Science and Engineering, ENB 118, University of South Florida
4202 E. Fowler Ave. Tampa, Fl 33620, USA
chawla, eschrich, hall@csee.usf.edu

Abstract

Ensembles of classifiers offer promise in increasing overall classification accuracy. The availability of extremely large datasets has opened avenues for application of distributed and/or parallel learning to efficiently learn models of them. In this paper, distributed learning is done by training classifiers on disjoint subsets of the data. We examine a random partitioning method to create disjoint subsets and propose a more intelligent way of partitioning into disjoint subsets using clustering. It was observed that the intelligent method of partitioning generally performs better than random partitioning for our datasets. In both methods a significant gain in accuracy may be obtained by applying bagging to each of the disjoint subsets, creating multiple diverse classifiers. The significance of our finding is that a partition strategy for even small/moderate sized datasets when combined with bagging can yield better performance than applying a single learner using the entire dataset.

1. Introduction

Dataset sizes are continually increasing as more and more information is stored electronically. Machine learning techniques are being utilized to learn models over increasingly large feature and example spaces. Efficiently learning from these large datasets is difficult, as datasets can not always be completely loaded into a computer's memory. Reducing training set sizes to the size of available memory or less is a practical approach in machine learning. An attractive option for learning from large datasets is *distributed learning*: data and learning are distributed across different processors (and computers). Our expanded version of the paper [3] carries more detailed discussion on the related distributed learning work, and also more details on our work and experiments. The approach discussed here is to learn an ensemble of individual classifiers, with each learner creating it's own classifier from a subset of the total dataset. We examine both a random partitioning method and a more intelligent partitioning method using clustering. With the addition of the bagging technique [2] applied to subsets contained in partitions, we show that disjoint dataset partitioning can actually yield better classifier performance than learning one model over the entire dataset.

2. Method

We describe below each of the methods used to partition a dataset into subsets from which an ensemble of classifiers can be built. In all instances, the ensemble of classifiers is composed of decision trees learned using C4.5 release 8 [4].

One of the simplest data partitioning approaches is to separate the dataset into n random disjoint subsets. The partitioning is done without respect to the class distribution within the dataset. Each disjoint subset is independently used in the generation of a decision tree classifier. This approach is well suited to distributed learning, since the entire dataset is never required to be loaded in memory at one time. Examples can be randomly chosen and distributed across a set of processors.

Fuzzy c-means (FCM) clustering [1] is used to examine the effects of intelligent partitioning of a dataset. A cluster-splitting FCM algorithm is applied to the dataset in order to create meaningful partitions of the data. The algorithm begins with two clusters ($c = 2$) and clusters until the fuzzy membership values are stable. The validity of the partition is evaluated and the "worst" cluster is split into two distinct clusters. This process is repeated until the stopping conditions are met. Since the number of clusters in the dataset is not known, values of c from 2 to 25 were used by the cluster splitting process. The splitting process is terminated early if the partition validity after clustering is worse than 5 times the best partition validity seen. This number was empirically observed to prune the search well, since a bad cluster split could almost never be improved by successive splits. Once the algorithm finishes clustering, the FCM step is repeated a final time with the best c found. A maximum membership function was used to harden the fuzzy clusters, creating a disjoint partition of the data.

Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM'01)
0-7695-1119-8/01 $17.00 © 2001 IEEE

3 Experiments

We evaluate the proposed approaches to learning by experiments on 7 well-known machine learning datasets. In all experiments, 10-fold cross validation is used. Results are reported as the mean classification performance over the 10 folds. The number of clusters found by FCM was taken as the number of random partitions to create. Comparisons between methods is done via a two-tailed paired two sample for means t-test among fold results, setting the confidence level, $\alpha = 0.025$.

In the random partition experiments, a simple majority vote is used to combine classification predictions. In the clustering experiments, an ask-expert combination method can be used. When FCM clustering of the training set is completed, the values of the cluster centroids are stored. When a test example is presented for classification, the closest centroid (using the Euclidean distance metric) is determined. This centroid corresponds to a cluster of training data, from which a decision tree was created. Only this decision tree (i.e. the expert) is consulted for a classification prediction.

The results of training and testing ensembles of classifiers according to the several partitioning methods described above can be seen in Table 1.

Dataset	Clusters	Full C4.5	Random	FCM
Page-block	2	96.90	96.82	96.95
Phoneme	5	86.50^{+}	83.44	85.99
Satimage	9	86.30	87.44^{+}	86.01
Pendigits	4	96.57^{+}	96.06	96.42
Mammography	2	98.50	98.51	98.40
Letter	2	88.10^{*+}	83.54	86.08
Shuttle	3	99.96^{+}	99.92	99.95

$^{+}$ C4.5/Random winner * C4.5/Cluster winner

Table 1. Partitioning Results vs. C4.5.

The next phase of experiments was to investigate the bagging phenomenon within our partitions. The resulting clusters and random partitions from Table 1 were bagged using 50 bags per partition (80% bag size). Most significantly, a random partition of a dataset, when combined with bagging performs better than a single decision tree learning the entire dataset.

4 Conclusion

In this paper, we present a novel approach to distributed learning using fuzzy clustering. This intelligent method of partitioning a dataset is compared to simpler, random methods of partitioning. In general, intelligent partitioning of a dataset provides better performance than random partitioning, and generally performs as well as C4.5 over the entire dataset. The results presented in this paper suggest that for very large datasets, the creation of ensembles of classifiers can perform reasonably well.

Dataset	Full C4.5	Random Bag 50 bags	FCM Bag 50 bags
Page-block	96.90	97.11	97.26*
Phoneme	86.50	85.77	88.71*
Satimage	86.30	87.61^{+}	86.76
Pendigits	96.57	97.22^{+}	98.18*
Mammography	98.50	98.52	98.78*
Letter	88.10	90.82^{+}	93.01*
Shuttle	99.96^{*+}	99.89	99.93

$^{+}$ C4.5/Random winner * C4.5/Cluster winner

Table 2. Bagging Results.

Interestingly, our results indicate that bagging of individual partitions can yield better results than learning from the entire dataset. It is surprising as bagged classifiers created from subsets, in effect, see much less data. Even in the case of random partitioning, where any individual classifier created on a subset often performs significantly worse than a single classifier learned on the entire dataset, bagging on disjoint subsets can improve performance. We believe this is due to the same effects that cause bagging to improve performance in general - bagging produces diverse classifiers from the data partitions, despite the smaller number of examples within a partition. We have thus proposed a novel and effective three-stage learning technique - partition, bag each partitioned subset, and learn.

5 Acknowledgements

This research was partially supported by the United States Department of Energy through the Sandia National Laboratories ASCI VIEWS Data Discovery Program, contract number DE-AC04-76DO00789 and by Tripos, Inc.

References

[1] J. C. Bezdek and S. K. Pal, editors. *Fuzzy Models For Pattern Recognition*. IEEE Press, New Jersey, 1991.

[2] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[3] N. Chawla, S. Eschrich, and L. O. Hall. Creating ensembles of classifiers. Technical Report ISL-01-01, University of South Florida, http://isl.csee.usf.edu/reports, 2001.

[4] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1992.