A Hierarchical Multiple Classifier Learning Algorithm *

Yu-Yu Chou
University of Washington
Department of Electrical Engineering
Box 352350, Seattle, WA 98195, U.S.A.
yuyu@ee.washington.edu

Linda G. Shapiro
University of Washington
Department of Computer Science and Engineering
Box 352350, Seattle, WA 98195, U.S.A.
shapiro@cs.washington.edu

Track: Pattern Recognition and Neural Networks
Keywords: Multiple Classifier, Data Clustering, Machine Learning

Abstract

In this paper we describe a new hierarchical multiple classifier learning algorithm that was developed to provide a tool for classifier construction in medical applications. The new mechanism uses data clustering and sub-class labeling to reduce the overhead of training and enhance the classification accuracy. The results of using this method are compared to a hand-built classifier and to other multiple classifier algorithms.

1 Introduction

Multiple classifier learning algorithms have been widely utilized in various machine learning problems. In the survey paper by Dietterich [5], various algorithms and techniques were described and the reasons that their performances surpass the stand-alone learning algorithms were discussed. However, those algorithms do not scale well for the mega-data [1] in some real applications. In this paper, we will describe a new multiple classifier mechanism and its success in a medical diagnosis application.

The pattern recognition (classification) / learning methodology described in this paper is a hierarchical multiple classifier mechanism. It is summarized in the block diagram of Figure 1. The system includes a training phase and a classification phase. In the training phase, the data are clustered into sub-populations where statistical information for each cluster is computed. The data are then relabeled with the sub-class labels and a set of component classifiers are constructed for each sub-population. A reduced feature set is produced according to the initial results from component classifiers and can be used to refine the construction of component classifiers if necessary. A set of error instances can be obtained to refine the training process as well. After the component classifiers are built, they are applied to the entire data set to produce a new set of data with the additional classifications as new features. The new data set can be considered as a non-linear transformation of the original data set or as a cross-validation data set. A super-classifier is constructed from the new data set and the cluster information computed previously. The process can be repeated for another layer of super-classifiers, recursively. In the classification phase, the process is similar to the training phase. For an unknown data instance, the component classifiers are applied to it to produce the input data for the super-classifier. The final result is produced by the super-classifier according to the the input and the cluster information.

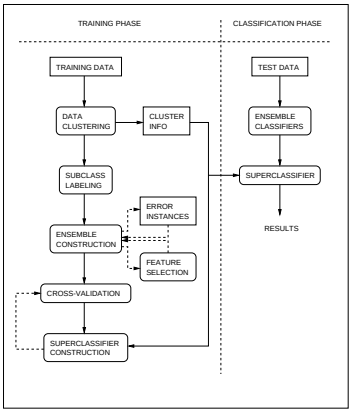

Figure 1: A block diagram of data clustering and component classifier training.

This paper describes the new multiple-classifier learning algorithm. Section 2 describes the characteristics of the medical data sets that motivated the research. Section 3 describes the algorithm in detail, and section 4 describes the experiments and results.

2 Data Characteristics

The application for which we developed the learning algorithm is pap smear slide prescreening [8] [9]. The task

* This research was partially supported by the Washington Technology Center and NeoPath, Inc.

[1] Data with a huge amount of cases and complicated features

is to identify an extracted cell instance as one of many possible classifications. There are several characteristics for this particular application: 1) the tremendous amount of data ((# of cells/per slide) $\times$ (# of slides/per day) $\times$ (# of labs)); 2) the complicated feature sets ($\geq$ 300); and 3) many different sources of noise in the data. These characteristics make the training and classification task difficult.

We received two sample data sets from NeoPath, Inc [2], one of the pioneers of the automated pap smear prescreening technology. One of the characteristics in the data set is that there are three levels of classes for target classification. There are three classes (*Normal, Abnormal, Artifact*) at the top-most level. Each class has its sub-classes and each sub-class has its own sub-classes. This is quite helpful in designing the component classifiers. The objective of the classifiers is to derive decision boundaries to separate data instances from different classes. For a complicated or noisy data set, the decision boundaries are also very complicated. All of the classification algorithms derive the decision boundaries with the goal of minimizing the misclassification rate of the training data. They also need to keep the decision boundaries smooth (generalized) enough to avoid over-fitting the training data. When data are noisy or complicated and the target consists of only a few classes, the decision boundaries are usually over-generalized and are in favor of those classes with a larger number of data instances. With the existence of sub-classes, the estimated boundaries can be fit closer to the true boundaries, improving the classification accuracy.

For data sets without multiple level classes, the sub-class labeling is still applicable with additional steps. A graph theoretic clustering algorithm [11] can be utilized to produce clusters that are then assigned new sub-class labels and used just like the predefined sub-classes in the construction of the classifier.

3 Hierarchical Multiple Classifier Algorithm

The hierarchical multiple classifier mechanism described in this paper consists of two major parts: 1) the construction of component classifiers; and 2) the combination scheme. Figure 2 shows the construction of component classifiers. The training data are first partitioned into various data clusters, each data cluster represents a sub-population of the original training data. The class distribution of each data cluster is different from that of the original training data. Instead of utilizing the original classes as the target outputs, we assign the sub-classes as the target outputs of each data cluster; the component classifiers are trained to learn the decision boundaries between these sub-classes. The label "Level II Classes" in the figure refers to the assignment of the second-level sub-classes. There are 7 sub-classes [3] in the pap-smear data; 5 of them are abnormal. A similar label "Level I Classes" utilized in Figure 3 refers to the three top-most level classes *Normal, Abnormal,* and *Artifact*. For each data cluster, the mean and the variance of the features of its members are computed and stored along with the member instances.

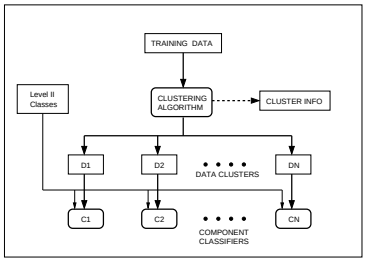

Figure 2: A block diagram of data clustering and component classifier training.

Figure 3 illustrates the combination scheme. After the component classifiers {C1, C2, ..., CN} are constructed for each data cluster, the whole data set is evaluated for the component classifiers with the unseen instances as the cross-validation set. Each component classifier generates a probabilistic output; they are combined as the input data for the next level classifier (super-classifier). A ranking of the data clusters is generated for each data instance. The rank of a data cluster is assigned according to the distance between the data instance and the centroid of a data cluster. The highest rank is assigned to the data cluster whose centroid is closest to the data instance. These ranks are utilized to control the combination of outputs of the component classifiers. An additional error classifier CE, which is trained on the error instances generated from the other component classifiers, can also be added to produce additional data for the super-classifier. In training the super-classifier, either the original classes 'Level I Classes" or the sub-classes 'Level II Classes" can be assigned as the target.

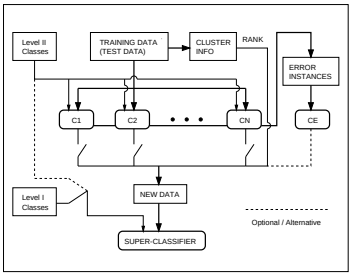

Figure 3: A block diagram of the combination scheme.

There are four approaches in the literature for constructing the classifier ensembles, namely 1) manipulating the training data; 2) manipulating the input features; 3) manipulating the target function; and 4) incorporating randomness into the algorithms. Our hierarchical system is developed based on two of these principles: subsampling the training data and manipulating the target function.

Subsampling is achieved by clustering the data using

[2] Now part of TriPath, Inc.

[3] There are 142 classes associated with the third-level sub-classes.

a graph-theoretic clustering algorithm [11]. Each feature vector in the training data becomes a node in the graph. Those node pairs that have feature-space distance less than a preset threshold are connected by edges in the graph. The clusters produced are controlled by parameters that specify the compactness, interrelatedness, size, and number of clusters. The result is a "soft partition" of the data, since clusters may overlap. The advantage of this clustering algorithm is that it allows the borderline data instances to be trained for different component classifiers. Therefore the decision boundaries won't be skewed by a specific component classifier in the combining stage.

The target function is manipulated by using different classification categories for different stages of learning. Most learning algorithms treat a multi-class problem as a recursive two-class problem. Since the decision boundaries can be very easily and reliably detected for a two-class problem, the learning algorithm can focus on finding the best dividing boundaries first, then break the data into separate regions and repeat the process. Unfortunately, this approach did not work well on the complex, noisy pap-smear data. While high accuracy could be achieved on the training data, results on the test data were unacceptable.

In our methodology, instead of simplifying the problem into a recursive two-class problem, we introduce the concept of sub-class relabeling. While it may seem that we have complicated the problem; on the contrary we can compensate for the shortcomings of the two-class methodology by using the more complicated decision boundaries associated with multiple classes. We can consider the sub-class output as an intermediate representation for a data instance in a sense similar to the output of the hidden nodes in a back-propagation neural network. The function of the super-classifier can be considered to be a piece-wise smoothing process to connect the boundaries constructed by different component classifiers. Our results will demonstrate that the use of sub-class relabeling improves the classification accuracy for both the training and test data.

Our super-classifier approach can be recursively applied; that is, different learning algorithms or different settings can be utilized to learn the super-classifier. We can have a set of super-classifiers just as we have a set of component classifiers. This next stage of super-classifiers can be trained in the same way. In our experiments, the results are usually stabilized after three levels of classifiers.

4 Experiments and Results

We evaluated our system with various learning algorithms including neural networks (NevProp [7]) and decision trees (C4.5 [10]) as the building blocks for component classifiers and super-classifiers. We also evaluated different clustering algorithms and different numbers of clusters (see [3] for experimental details and full results.) Two terms are used to describe the expected results. They are 1) **sensitivity** - the percentage of abnormal cases classified as abnormal, and 2) **specificity** - the percentage of normal cases classified as normal. The goal for the classifier is to achieve a high *sensitivity* and a high *specificity* simultaneously.

Figure 4 shows some of the results for the first data set. There are 19,125 cases with 323 features for each case in the set. The classifiers were trained using the NevProp algorithm. Different numbers of clusters are recorded in the figure (5, 10, 14, and 20.) The notation "S-S" denotes that both the component classifiers and the super-classifier were trained with the Level II classes as targets. "S-O" means that the component classifiers were trained with the Level II classes as targets, while the super-classifier was trained with the Level I classes as targets. Our results, shown with symbols, can be compared to results from a semi-automatically produced, hand-tuned classifier developed expressly for the pap-smear application, shown with dashed lines. The use of sub-class labeling clearly increases the accuracy. Notice that when the cluster number grows to 20, there is a significant drop in specificity accuracy. The reason is that the component classifiers are trained on a sub-population of the training data; the larger the cluster number, the fewer cases in each data cluster for training. Since the data were prepared with fewer normal cases than abnormal cases, the performance drops in specificity, but not in sensitivity.

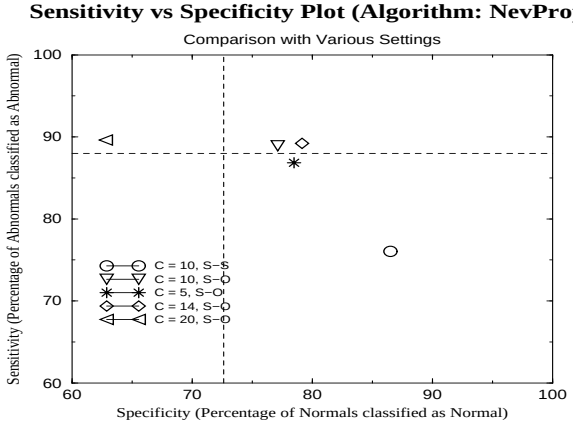

Figure 4: Results of different settings for pap-smear data set 1. The dashed lines denote the results of the hand-crafted classifier. A result located in the upper-right region is preferred.

Figure 5 shows some of the results for the second data set. There are 24,345 cases with 291 features for each case in the set. The notation "k" in the figure indicates how many

outputs of the component classifiers were fed to the super-classifier. The last entry in the figure shows the result of a three-level classifier. The second-level super-classifiers are constructed by setting k = 3, 5, and 10 respectively.

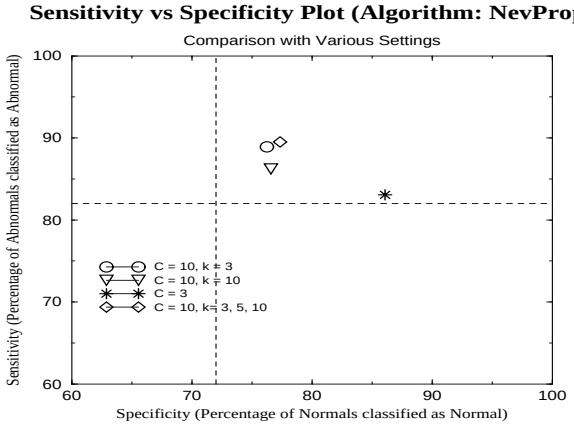

Figure 5: Results of different settings for pap-smear data set 2.

We also compared our approach against two other popular multiple classifier algorithms, bagging [2] and boosting [6]. The drawback of these two algorithm is that when constructing the classifier ensembles, the number of data instances for each component classifier must be at least as large as the number of the total training instances. For the application demonstrated in this paper, this created a tremendous burden both on the training process and the system resource usage. Since our component classifiers are trained on smaller data subsets, we can accelerate the training process without sacrificing the performance. In this experiment, only a subset of features (74) was utilized due to a system resource limitation. Table 1 shows the results for the second data set with various algorithms.

Table 1. The comparison of various multiple classifier algorithms.

	Accuracy %		
Algorithm	Abnormal	Normal	Overall
Bagging	81.9	70.9	78.9
Boosting	80.0	69.8	77.2
Hierarchical	84.4	66.1	79.4

In addition to the pap-smear data set, we tested the hierarchical classifier learning algorithm on a forest cover data set from the UCI Knowledge Discovery in Databases Archive and originally from Colorado State University [1]. This data contains 581,012 instances with 54 fields per instance. Using C4.5 for the component classifiers, the hierarchical method obtained an accuracy of 70.81% on the test data, compared to only 63.64% using C4.5 alone, 23.96% using NevProp alone, and 68% reported for the commercial product NeuNet [4], which used twice the training data.

5 Conclusion

In this paper, we described a hierarchical multiple-classifier classification scheme, which preserves the strength of the multiple-classifier approach and also manages to reduce some of the problems faced by other multiple-classifier algorithms. In our scheme, the system resource requirements are reduced and so is the training time. From the results on pap-smear data, it can be seen that our approach produces better performance than other multiple-classifier algorithms and better than a classifier produced by human experts.

References

[1] Blackard, J. A. Forest Covertype data. ftp://kdd.ics.uci.edu/databases/covertype/covertype.data.gz, 1996. Remote Sensing and GIS Program, Department of Forest Sciences, College of Natural Resources, Colorado State University.

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

[3] Chou, Y. Y. *Hierarchical Multiple Classifier Learning System*. PhD thesis, University of Washington, 1999.

[4] CorMac Technologies Inc. NeuNet Pro v2.1 for Windows. http://www.cormactech.com/neunet/, 1999.

[5] Dietterich, T. G. Machine-Learning Research: Four Current Directions. *AI Magazine*, 18(4):97–136, 1997.

[6] Freund, Y.and Schapire, R. E. Experiments with a New Boosting Algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156, 1996.

[7] Goodman, P., Rosen, D., Egbert, D., Carlson, D., Hallett, J, and Ju, W. Nevprop version 3. http://www.scs.unr.edu/∼cbmr/, 1998.

[8] Lee, J. S. J., Bannister, W. I., Kuan, L. C., Bartels, P. H., and Nelson, A. C. A Processing Strategy for Automated Papanicolaou Smear Screening. *Analytical and Quantitative Cytology and Histology*, 14(5):415–425, Oct 1992.

[9] Patten, Jr., S. F., Lee, J. S. J., and Nelson, A. C. Neopath, Inc. NeoPath AutoPap 300 Automatic Pap Screener System. *Acta Cytologica*, 40(1):45–52, Jan–Feb 1996.

[10] Quinlan, J. R. *C4.5 Programs for Machine Learning*. Morgan Kaufmann, 1993.

[11] Shapiro, L. G. and Haralick, R. M. Decomposition of Two-Dimensional Shapes by Graph-Theoretic Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(1):10–20, Jan 1979.