Ensemble Data Mining Methods

Nikunj C. Oza
NASA Ames Research Center, USA

INTRODUCTION

Ensemble Data Mining Methods, also known as Committee Methods or Model Combiners, are machine learning methods that leverage the power of multiple models to achieve better prediction accuracy than any of the individual models could on their own. The basic goal when designing an ensemble is the same as when establishing a committee of people: each member of the committee should be as competent as possible, but the members should be complementary to one another. If the members are not complementary, that is, if they always agree, then the committee is unnecessary—any one member is sufficient. If the members are complementary, then when one or a few members make an error, the probability is high that the remaining members can correct this error. Research in ensemble methods has largely revolved around designing ensembles consisting of competent yet complementary models.

BACKGROUND

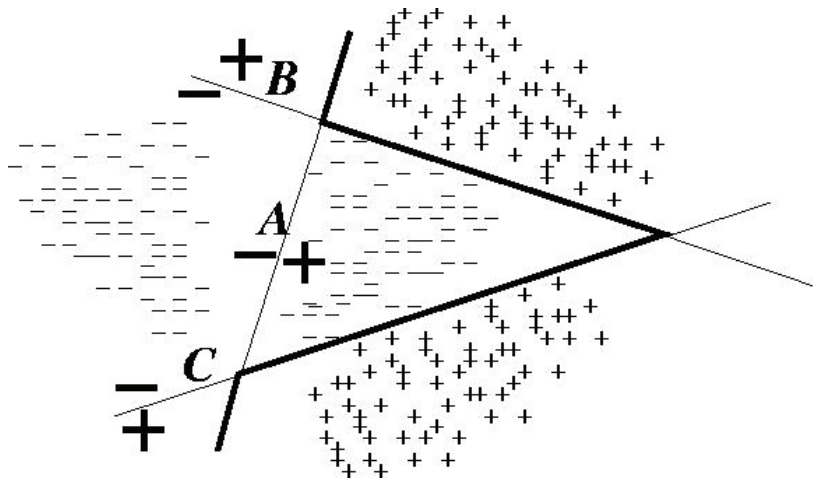
A supervised machine learner constructs a mapping from input data (normally described by several features) to the appropriate outputs. It does this by learning from a training set—N inputs $x_1, x_2, ..., x_N$ for which the corresponding true outputs $y_1, y_2, ..., y_N$ are known. The model that results is used to map new inputs to the appropriate outputs. In a classification learning task, each output is one or more classes to which the input belongs. The goal of classification learning is to develop a model that separates the data into the different classes, with the aim of classifying new examples in the future. For example, a credit card company may develop a model that separates people who defaulted on their credit cards from those who did not based on other known information such as annual income. A model would be generated based on data from past credit card holders. The model would be used to predict whether a new credit card applicant is likely to default on his credit card and thereby decide whether to approve or deny this applicant a new card. In a regression learning task, each output is a continuous value to be predicted (e.g., the average balance that a credit card holder carries over to the next month).

Many traditional machine learning algorithms generate a single model (e.g., a decision tree or neural network). Ensemble learning methods instead generate multiple models. Given a new example, the ensemble passes it to each of its multiple *base* models, obtains their predictions, and then combines them in some appropriate manner (e.g., averaging or voting). As mentioned earlier, it is important to have base models that are competent but also complementary (Tumer and Ghosh, 1996). To further motivate this point, consider Figure 1. This figure depicts a classification problem in which the goal is to separate the points marked with plus signs from points marked with minus signs. None of the three individual linear classifiers (marked A, B, and C) is able to separate the two classes of points. However, a majority vote over all three linear classifiers yields the piecewise-linear classifier shown as a thick line. This classifier is able to separate the two classes perfectly. For example, the plusses at the top of the figure are correctly classified by A and B, but are misclassified by C. The majority vote over these correctly classifies these points as plusses. This happens because A and B are very different from C. If our ensemble instead consisted of three copies of C, then all three classifiers would misclassify the plusses at the top of the figure, and so would a majority vote over these classifiers.

MAIN THRUST OF THE CHAPTER

We now discuss the key elements of an ensemble learning method and ensemble model and, in the process, discuss several ensemble methods that have been developed.

Figure 1. An ensemble of linear classifiers. Each line—A, B, and C—is a linear classifier. The boldface line is the ensemble that classifies new examples by returning the majority vote of A, B, and C



Ensemble Methods

The example shown in Figure 1 is an artificial example. We cannot normally expect to obtain base models that misclassify examples in completely separate parts of the input space and ensembles that classify all the examples correctly. However, there are many algorithms that attempt to generate a set of base models that make errors that are as different from one another as possible. Methods such as Bagging (Breiman, 1994) and Boosting (Freund and Schapire, 1996) promote diversity by presenting each base model with a different subset of training examples or different weight distributions over the examples. For example, in figure 1, if the plusses in the top part of the figure were temporarily removed from the training set, then a linear classifier learning algorithm trained on the remaining examples would probably yield a classifier similar to C. On the other hand, removing the plusses in the bottom part of the figure would probably yield classifier B or something similar. In this way, running the same learning algorithm on different subsets of training examples can yield very different classifiers which can be combined to yield an effective ensemble. Input Decimation Ensembles (IDE) (Tumer and Oza, 2003) and Stochastic Attribute Selection Committees (SASC) (Zheng and Webb, 1998) instead promote diversity by training each base model with the same training examples but different subsets of the input features. SASC trains each base model with a random subset of input features. IDE selects, for each class, a subset of features that has the highest correlation with the presence or absence of that class. Each feature subset is used to train one base model. However, in both SASC and IDE, all the training patterns are used with equal weight to train all the base models.

So far we have distinguished ensemble methods by the way they train their base models. We can also distinguish methods by the way they combine their base models' predictions. Majority or plurality voting is frequently used for classification problems and is used in Bagging. If the classifiers provide probability values, simple averaging is commonly used and is very effective (Tumer and Ghosh, 1996). Weighted averaging has also been used and different methods for weighting the base models have been examined. Two particularly interesting methods for weighted averaging include Mixtures of Experts (Jordan and Jacobs, 1994) and Merz's use of Principal Components Analysis (PCA) to combine models (Merz, 1999). In Mixtures of Experts, the weights in the weighted average combining are determined by a gating network, which is a model that takes the same inputs that the base models take, and returns a weight for each of the base models. The higher the weight for a base model, the more that base model is trusted to provide the correct answer. These weights are determined during training by how well the base models perform on the training examples. The gating network essentially keeps track of how well each base model performs in each part of the input