Ensembles and Methods Constructing Ensembles

Li Zichao

15220162202173

May 5, 2019

Abstract

This report mainly concludes the basic concept of ensemble methods and several methods for constructing ensembles.

1 Ensemble Methods

Ensemble methods are learning algorithms that construct a set of calssifiers and then classify new data points by taking a weighted vote of their prediction. The original ensemble method is Bayesian averaging, but more recent algorithms include error-correcting output coding, Bagging and boosting. Ensembles are often considered to be better than any single reveiwed for three fundamental reasons[1]. Those fundamental issues are the most important ways in which existing learning algorithms fail. But ensemble methods have the promise of reducing these key shortcomings of standard learning algorithms. These issues will be elaborated later in this section.

Let's begin the introduction of ensenmble methods from the standard supervised learning problem. In some unknown function $y = f(x)$, the training examples of a learning program is given in the form of $(\mathbf{x}_1, y1), \ldots, (\mathbf{x}_m, y_m)$. The $\mathbf{x}_i$ values are typically vectors of the form $\langle x_{i,1}, x_{i,2}, \ldots, x_{i,n} \rangle$, whose components are called the *features* of $\mathbf{x}_i$.

Given a set S of training examples, a learning algorithms can output a *classifier*, which is a hypothesis about the true function f. With new $\mathbf{x}$ values, the classifier predicts the corresponding y values. An **ensemble of classifier** is a set of classifiers

[1]Including **statistical**, **computational** and **representiational**.

whose individual decisions are combined in some way, typically by weighted or un weighted voting, to classify new examples.

Accuracy and **diversity** are two necessary and sufficient condition for and an ensemble of classifiers to be mor accurate than any of its individual members. Accuracy means that the classifier has an error rate of better than random guessing on new $\mathbf{x}$ values. Diversity means each classifier makes different errors on new data points. The importance of accuracy is intuitive. To demonstrate the importance of diversity, imagine the case that now we have three totally mutually indepentdent classifier with identical error rates equal to 0.3. We predict y as *true* if at least two of the three classifiers predict true. Then, the overall error rate is $0.3^3 + 3 \times 0.3^2 \times 0.7 = 0.216 < 0.3$. More precisely, if the error rates of L hypotheses h_l are all equal to p, then the probability that the majority vote will be rong will be the area under the binomial distribution where more than $L/2$ hypothese are wrong.

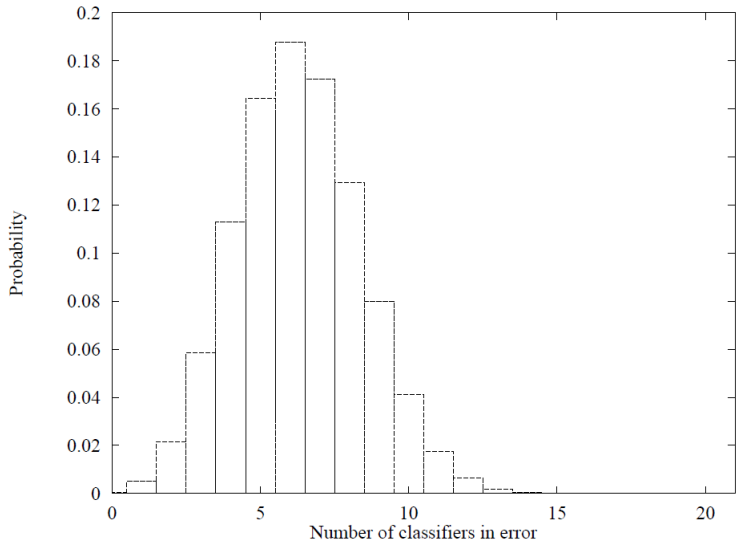

Figure 1: The probability that exactly l hypotheses will make an error, assuming each hypothesis has an error rate of 0.3 and makes its errorss independently of the other hypotheses.

As we mentioned before, there are three fundamental reasons make it often possible to construct very good ensembles, including:

1. **Statistical:** Ensembles can "average" acurate classifiers' votes and reduce the risk of choosing the wrong classifier.

2. **Computational:** An ensemble constructed by running the local search from many different starting points may provide a better approximation to the true unknown function.

3. **Representational:** By forming weighted sums of hypotheses it may be possible to expand the space of representable functions.

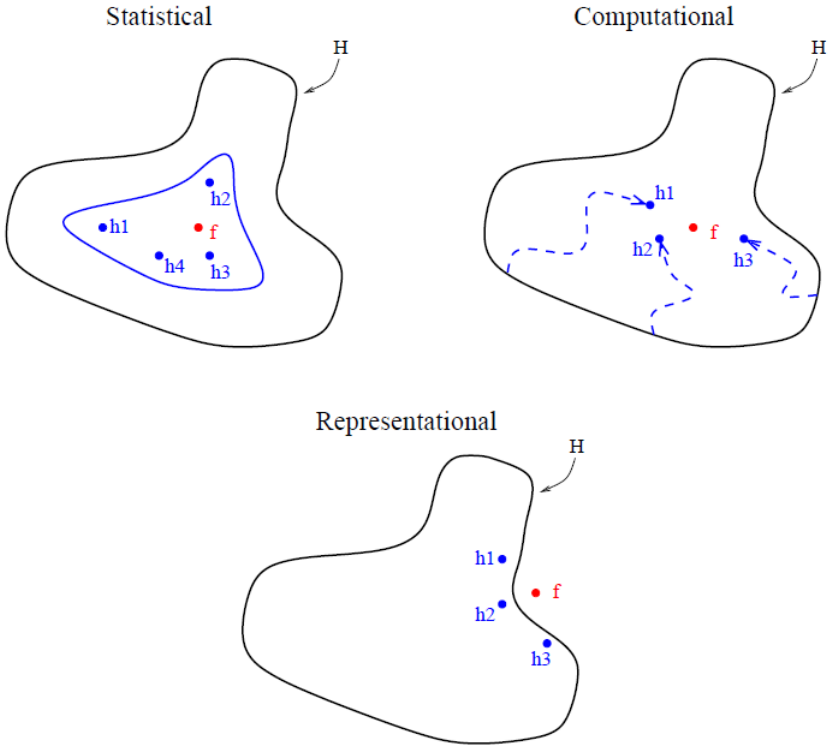

Figure 2: Three fundamental reasons why an ensemble may work better than a single classifier.

2 Methods for Constructing Ensembles

2.1 Bayesian Voting: Enmerating the Hypotheses

In a Bayesian probabilistic setting, the problem of predicting the value of $f(x)$ can be written as weighted sum over all hypotheses in H:

$$P(f((x)) = y|S, \mathbf{x}) = \sum_{h \in H} h(\mathbf{x}) P(h|S)$$

It can be viewed as an ensemble method in which the ensemble consists of all of the hypotheses in H, each weighted by its posterior probability $P(h|S)$:

$$P(h|S) \propto P(S|h)P(h)$$

In some learning problems, it is possible to completely enumerate each $h \in H$, compute $P(S|h)$ and $P(h)$, and evaluate this Bayesian form. In complex problems where H cannot be enumerated, it is sometimes possible to approximate Bayesian voting by drawing a random sample of hypotheses distributed according to $P(h|S)$.

The most idealized aspect of the Bayesian analysis is the prior belief $P(h)$. If this prior completely captures all of the knowledge about f before obtaining S, then by definition that is the best prediction. But in practice, it is often difficult to construct a space H and assign a prior $P(h)$ that captures our prior knowldge adequately.

2.2 Manipulating the Training Examples

The second method for constructing ensembles manipulates the training examples to generate multiple hypotheses. The learning algorithm runs several times, each time with a different subset of the training examples. This technique works especially well for unstable learning algorithms.

The most straightforward way of manipulating the training set is called **Bagging**[2]. On each run, Bagging presents the learning algorithm with a training set that consists of a sample of m training examples drawn randomly with replacement from the originla training set of m items. Such a training set is called a *bootstrap replicate* of the original training set.

2.3 Manipulating the Input Features

A third genearl technique for generating multiple classifiers is to manipulate the set of *input features* available to the learning algorithm. For example, in a project to identify volcanoes on Venus, Cherkauer(1996) trained an ensemble of 32 neural networks. The 32 networks were based on 8 different subsets of the 118 available input features and 4 different network sizes. The input feature subsets were selected to group

[2]Short for "bootstrap aggregation".

together features that were based on different image processing opreations. Obviously, this technique only works when the input features are highly redundant.

2.4 Manipulating the Output Targets

Suppose the number of classes, K, is large. Then new learning problems can be constructed by randomly partioning the K classes into two subsets A_l and B_l. The input data can be re-labeled so that any of the original classes in set A_l are given the dreived label 0 and the original classes in set B_l are given the derived label 1. This relabeled data is then used to construct a classifier h_l. By repeating this process L times, we can obtain an ensemble of L classifiers.

Given a new data point $\mathbf{x}$, we classify it with each h_l. If $h_l(\mathbf{x}) = 0$, then each class in A_l receives a vote. Otherwise each class in B_l reveives a vote. The class with the highest number of votes is selected as the prediction of the ensemble.