ENSEMBLE
Zhi-Hua Zhou
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing, China
http://cs.nju.edu.cn/zhouzh/

SYNONYMS
Committee-Based Learning; Multiple Classifier System; Classifier Combination.

DEFINITION
Ensemble is a machine learning paradigm where multiple learners are trained to solve the same problem. In contrast to ordinary machine learning approaches which try to learn one hypothesis from training data, ensemble methods try to construct a set of hypotheses and combine them to use.

HISTORICAL BACKGROUND
It is difficult to trace the starting point of the history of ensemble methods since the basic idea of deploying multiple models has been in use for a long time. However, it is clear that the hot wave of research on ensemble methods since the 1990s owes much to two works. The first is an applied research conducted by Hansen and Salamon at the end of 1980s [5], where they found that predictions made by the combination of a set of neural networks are often more accurate than predictions made by the best single neural network. The second is a theoretical research conducted in 1989, where Schapire proved that weak learners which are slightly better than random guess can be boosted to strong learners that are able to make very accurate predictions, and the proof resulted in Boosting, one of the most influential ensemble methods [10].

SCIENTIFIC FUNDAMENTALS
Terminologies
Learners composing an ensemble are usually called base learners. Many ensemble methods are able to boost weak learners which are slightly better than random guess to strong learners which can make very accurate predictions. So, “base learners” are also referred as “weak learners”. However, it is noteworthy that although most theoretical analyses work on weak learners, base learners used in practice are not necessarily weak since using not-so-weak base learners often results in better performance.

Base learners are usually generated from training data by a base learning algorithm which can be decision tree, neural network or other kinds of machine learning algorithms. Most ensemble methods use a single base learning algorithm to produce homogeneous base learners, but there are also some methods which use multiple learning algorithms to produce heterogeneous learners. In the latter case there is no single base learning algorithm and
thus, some people prefer calling the learners \textit{individual learners} or \textit{component learners} to “base learners”, while the names “individual learners” and “component learners” can also be used for homogeneous base learners.

\section*{Methods}

Typically, an ensemble is constructed in two steps. In the first step, a number of base learners are produced. Here the base learners can be generated in a parallel style or in a sequential style where the generation of a base learner has influence on the generation of subsequent learners. In the second step, the base learners are combined to use. The most popular combination scheme for classification is \textit{majority voting}, while the most popular combination scheme for regression is \textit{weighted averaging}. The employment of different base learner generation processes and/or different combination schemes leads to different ensemble methods.

The following three paragraphs briefly describe the working routines of three representative ensemble methods, Boosting \cite{Freund1997}, Bagging \cite{Breiman1996} and Stacking \cite{Wolpert1992}, respectively. Here, binary classification is considered for simplicity. That is, let $\mathcal{X}$ and $\mathcal{Y}$ denote the instance space and the set of class labels, respectively, assuming $\mathcal{Y} = \{-1, +1\}$. A training set $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$ is given, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ ($i = 1, \cdots, m$).

For Boosting, let us consider the most famous algorithm, AdaBoost, as an example. In the first step, a number of base learners are produced by emphasizing each learner on the training examples that are wrongly predicted by preceding learners. Here, a weight distribution is maintained over the training examples, which is initialized by equal weights. In the $t$-th learning round, a base learner $h_t : \mathcal{X} \rightarrow \mathcal{Y}$ is generated from the training set by using the current weight distribution, $D_t$. Then, $h_t$ is evaluated on the training set and let $\epsilon_t$ denote the error. The weight distribution for the next learning round, $D_{t+1}$, is generated in the way that for the training example $(x_i, y_i)$, $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \exp \left( \frac{1}{\epsilon_t} y_i h_t(x_i) \right)$ where $Z_t$ is a normalization factor which enables $D_{t+1}$ to be a distribution. In the second step, the base learners $h_t$’s ($t = 1, \cdots, T$) are combined by $H(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha_t h_t(x) \right)$.

For Bagging, in the first step a number of base learners are trained from bootstrap samples. A bootstrap sample is obtained by subsampling the training set with replacement, where the size of a sample is as the same as that of the training set. Thus, for a bootstrap sample, some training examples may appear but some may not, where the probability that an example appears at least once is about 0.632. On each sample a base learner $h_t : \mathcal{X} \rightarrow \mathcal{Y}$ is produced by calling a base learning algorithm. In the second step, Bagging combines the learners $h_t$’s ($t = 1, \cdots, T$) by majority voting, i.e., $H(x) = \text{argmax}_{y \in \mathcal{Y}} \sum_{t=1}^{T} 1(y = h_t(x))$ where the value of $1(a)$ is 1 if $a$ is true and 0 otherwise.

For a typical implementation of Stacking, in the first step, a number of individual learners, $h_t : \mathcal{X} \rightarrow \mathcal{Y}$ ($t = 1, \cdots, T$), are generated from the training set by employing different learning algorithms. In the second step, the individual learners are combined by using another learner. Here, for the training example $(x_i, y_i)$, a corresponding example $(z_i, y_i)$ is produced, where $z_i = (h_1(x_i), \cdots, h_T(x_i))$. Then, from $\{(z_i, y_i)\}$ ($i = 1, \cdots, m$) a learner $F : h^2 \rightarrow \mathcal{Y}$ is generated by calling a learning algorithm, which is used to combine the individual learners by $H(x) = F(h_1(x), \cdots, h_T(x))$.

The above methods have many variants. For example, \textit{Random Forests} \cite{Liaw2002}, which has been deemed as one of the most powerful ensemble methods, is a variant of Bagging.

There are many other established ensemble methods, e.g., \textit{Random Subspace} \cite{Ho1998}. This method generates a number of base learners from different \textit{subspaces} of the training set in the first step, and then combines these base learners via majority voting in the second step. Here, a subspace is actually a subset of the original attribute set.

It is worth mentioning that in addition to classification and regression, ensemble methods have also been designed for clustering \cite{Dawson2003} and other kinds of machine learning tasks.
It was thought that using more base learners will lead to a better performance, yet Zhou et al. [15] proved the “many could be better than all” theorem which indicates that this may not be the fact. It was shown that after generating a set of base learners, selecting some base learners instead of using all of them to compose an ensemble is a better choice. Such ensembles are called selective ensembles.

Note that usually the computational cost for building an ensemble comprising $T$ base learners is roughly $T$ times the cost of training a single learner. So, from the view of computational complexity, training an ensemble is almost as efficient as training a single learner.

**Why Useful?**

The generalization ability of an ensemble is usually much stronger than that of a single learner, which makes ensemble methods very attractive. An important question is, why can the ensembles be superior to single learners? For this, Dietterich [4] gave three reasons by viewing the nature of machine learning as searching a hypothesis space for the most accurate hypothesis. The first reason is that, the training data might not provide sufficient information for choosing a single best learner. For example, there may be many learners performing equally well on the training set. Thus, combining these learners may be a better choice. The second reason is that, the search processes of the learning algorithms might be imperfect. For example, even if there exists a unique best hypothesis, it might be difficult to achieve since running the algorithms results in sub-optimal hypotheses. Thus, ensembles can compensate for such imperfect search processes. The third reason is that, the hypothesis space being searched might not contain the true target function, while ensembles can give some good approximation. For example, it is well-known that the classification boundaries of decision trees are linear segments parallel to coordinate axes. If the target classification boundary is a smooth diagonal line, using a single decision tree can not lead to a good result but a good approximation can be achieved by combining a set of decision trees. Although these intuitive explanations are reasonable, they lack rigorous theoretical analyses.

The bias-variance decomposition is often used in studying the performance of ensemble methods [1, 15]. It is known that Bagging can significantly reduce the variance, and therefore it is better to be applied to learners suffered from large variance, e.g., unstable learners such as decision trees or neural networks. Boosting can significantly reduce the bias in addition to reducing the variance, and therefore, on weak learners such as decision stumps, Boosting is usually more effective.

There are many theoretical studies on famous ensemble methods such as Boosting and Bagging, yet it is far from a clear understanding of the underlying mechanism of these methods. For example, empirical observations show that Boosting often does not overfit even after a large number of rounds, and sometimes it is even able to reduce the generalization error after the training error has already reached zero. Although many people have studied this phenomenon, theoretical explanations are still in arguing.

**Accuracy and Diversity**

Generally, in order to construct a good ensemble, the base learners should be as more accurate as possible, and as more diverse as possible. This has been formally shown by Krogh and Vedelsby [7], and emphasized by many other people.

The definition of accuracy is clear, and there are many effective processes for estimating the accuracy of learners, such as cross-validation, hold-out test, etc. However, there is no rigorous definition on what is intuitively perceived as diversity. Although a number of diversity measures have been designed, Kuncheva and Whitaker [8] revealed that the usefulness of existing diversity measures in building ensembles is suspectable.

In practice, the diversity of the base learners can be introduced from different channels, such as subsampling the training examples, manipulating the attributes, manipulating the outputs, injecting randomness into learning algorithms, or even using multiple mechanisms simultaneously.
KEY APPLICATIONS

Ensemble methods have already been used in diverse applications such as optical character recognition, text categorization, face recognition, medical diagnosis, gene expression analysis, etc. Actually, ensemble methods can be used wherever machine learning techniques can be used.

FUTURE DIRECTIONS

A serious deficiency of ensemble methods is the lack of comprehensibility, i.e., the knowledge learned by ensembles is not understandable to the user. Improving the comprehensibility of ensembles [14] is an important yet largely understudied direction.

Diversity plays an important role in ensembles, yet currently no diversity measures is satisfying [8]. Exploring the relation between the performance of ensembles and the properties of base learners to design useful diversity measures is an interesting direction, which may lead to the development of more powerful ensemble methods.

Although there are much theoretical analyses on ensemble methods, many underlying mechanisms of successful ensemble methods are not clear. So, more theoretical studies are needed. Another ambitious attempt is to establish a general theoretical framework for ensemble methods.

EXPERIMENTAL RESULTS

Empirical studies on popular ensemble methods have been reported in many papers, such as [1, 12, 9].

DATA SETS

A large collection of datasets commonly used for experiments can be found at http://www.ics.uci.edu/~mlearn/MLRepository.html

URL TO CODE

The code of Random Forests can be found at http://www.stat.berkeley.edu/~breiman/RandomForests/

CROSS REFERENCE

BAGGING
BOOSTING
SUPPORT VECTOR MACHINE
DECISION TREE
NEURAL NETWORKS
RECOMMENDED READING
Between 3 and 15 citations to important literature, e.g., in journals, conference proceedings, and websites.