Extracting Independent Rules: A New Perspective of Boosting

Yuchen Zhang[1] and **Li Zhang**[2]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, 10084, China
[2]School of Software, Tsinghua University, Beijing, 10084, China

Abstract

Boosting is one of the most significant development in machine learning areas in recent years. Although boosting has already achieved great success in practical applications, its internal mechanism has not been entirely understood. In this paper, we present a new perspective to design boosting algorithms: extracting independent weak rules. A boosting algorithm can be divided into two parts, an extractor and a combiner. We first introduce the concept of independency into boosting. Our target is to use an extractor to generate a sequence of high-accuracy weak rules that are mutually independent on the original data distribution, then use a combiner to merge these independent rules into a strong classifier. In order to design such a boosting algorithm, we introduce an assumption based on the essence of weak learners. In this perspective, the mechanism of AdaBoost can be interpreted very naturally, and a criterion evaluating whether a weak learner is suitable to be used for boosting is proposed. A series of experiments are conducted on real datasets to verify the theoretical conclusions we derived in this paper.

Keywords: machine learning, boosting, adaboost, independent rules.

1 Introduction

The concept of boosting was first introduced by (Kearns & Valiant, 1988, 1994). The target is to "boost" a group of weak classifiers that perform only slightly better than random guessing, to an arbitrarily accurate strong classifier. The most typical and widely used boosting algorithm is the AdaBoost algorithm introduced by (Freund & Schapire, 1995). A brief version of the pseudocode of AdaBoost is given in Figure 1. For more information about the histories and recent developments of boosting, see the overview article by (Schapire, 2002).

Several theoretical researches have been conducted to explain the mechanism of boosting. (Friedman, Hastie, & Tibshirani, 2000) showed that boosting can be viewed as an approximation to additive modeling on the logistic scale using maximum Bernoulli likelihood as a criterion. (Schapire, 2001) introduced a kind of game named drift games that can model some boosting and online-learning algorithms. An analysis from (Kivinen & Warmuth, 1999) showed that Adaboost minimizes the relative entropy between the new distribution and previous distributions, under the constraint that the vector of mistakes of the current weak hypothesis is always orthogonal to the old ones. However, these ideas seem to be more mathematical but less intuitive, and each of them can only cover a single part of the boosting area.

Given: training examples $(x_1, y_1), \cdots, (x_m, y_m)$. Start with distribution $D_1(k) = 1/m$.
For $t = 1, 2, \cdots, T$:

1. Generate classifier h_t using weak learner W on distribution D_t.
2. Evaluate the accuracy of h_t on D_t by $\mathrm{c}_t = \sum_{h(x_k)=y_k} D_t(k)$, set $\alpha_t = \frac{1}{2} \ln \left(\frac{c_t}{1-c_t} \right)$.
3. Update: $D_{t+1}(k) = \frac{D_t(k) \exp(-\alpha_t y_k h_t(x_k))}{Z_t}$ where Z_t is the normalization factor.

Output the final classifier: $H(x) = \mathrm{sgn}\Big(\sum_{t=1}^{T} \alpha_t h_t(x) \Big)$.

Figure 1: The AdaBoost Algorithm.

In this paper, we consider the problem of boosting in a more natural perspective. When boosting algorithm generates a group of rough rules from the instance space, we hope that these rules are *independent*, or *mutually uncorrelated*. The concept of independency of a group of rules means that for an example picked randomly from the instance space according to some distribution, whether this example conforms to (can be correctly classified by) one rule is entirely unrelated to its conformance with other rules. We will give a mathematical definition of independency in Section 2. In fact, people often follow the principles of independency when they make decisions in daily life. For example, before a law is made the government would collect opinions from people who do different jobs and represent different social classes, because these people consider the same problem in independent perspectives. In machine learning problems, we believe that a group of independent rules have the ability to describe the characteristics of a dataset most efficiently and most comprehensively. Moreover, if a group of rules are mutually independent, their correctness on an instance space can be represented by a group of independent random variables, this property holds obvious advantages for further analysis.

Our target is to extract a group of weak rules using an *ex-*

tractor. These rules should be mutually independent on the original distribution D_1, and we hope their accuracies on D_1 being as high as possible. Then a *combiner* can be designed to merge these independent rules into a strong classifier. It is hard to extract such a group of independent rules from the original distribution directly. But fortunately, we have an alternative choice. We start from the original distribution D_1, train a rule h_1 from D_1 using the weak learner W. We try to construct a new distribution D_2 from (D_1, h_1) such that the rule h_2 trained from D_2 is independent with h_1 on D_2. Then we construct another distribution D_3 from (D_1, h_1, D_2, h_2) such that the rule h_3 trained from D_3 is independent with h_1, h_2 on D_3. In general, we sequentially construct new distributions according to the properties of existing distributions and rules, and try to make each new rule being independent with existed rules on its training distribution. When all rounds of training are finished, it can be proved that the independency of these rules is preserved when they are applied to the original distribution D_1, this property is called *the lifting of independency*. It can also be proved that each rule has the accuracy on D_1 as high as its training accuracy, this is called *the lifting of accuracy*. Through this method, we can reduce the problem of constructing a group of independent rules to the problem of constructing a sequence of appropriate distributions, the latter problem is relatively easier to be solved. However, an assumption is still needed to design the practical extractors and combiners. We call this assumption *the independency assumption*, which is based on the essence of the weak learner W.

In this paper, we introduce a most naive, or most intuitive version of the independency assumption. Under this assumption, a *naive extractor* can be designed to generate a group of rules which are mutually independent on D_1 and has high accuracies, and a *naive combiner* can be designed to merge these rules. The feasibility will be proved in Section 3. The upper bound of the error for the final classifier can be estimated by a very simple statistical analysis. We will point out that Adaboost is exactly composed by a naive extractor and a naive combiner, so AdaBoost is the algorithm derived from the naive version of independency assumption.

In Section 4 we conduct a series of experiments to verify the adaptability of the independency assumption under weak learners C4.5, NaiveBayes and Artificial Neural Networks. Since all advantages on independency proved in Section 3 are based on the independency assumption, so we can also use the independency assumption as a criterion to see whether a weak learning is suitable to be improved through boosting.

2 The Independency of Rules

An instance space $X \times Y$ contains m examples: $(x_1, y_1), \cdots, (x_m, y_m)$, where $x_k \in X$ and $y_k \in Y = \{-1, 1\}$. A map set $\mathcal{H}$ is the set of weak rules that can be generated by the weak learner W. $h \in \mathcal{H} : X \to Y$ is a rule on the instance space, we also call it as a *hypothesis*, in this paper, we assume that $\mathcal{H}$ is symmetric, which means that if $h \in \mathcal{H}$, then its opposite $-h \in \mathcal{H}$. A distribution over the instance space is a function $D : \{1, \cdots, m\} \to [0, 1]$, which satisfies the normalization condition $\sum_{k=1}^{m} D(k) = 1$. $D(k)$ indicates the weight, or the quantity proportion, of example (x_k, y_k). The *conforming function* of h is defined to be

$$\varphi(x) = \begin{cases} 1 & h(x) = y \\ 0 & h(x) \neq y \end{cases} \tag{1}$$

It indicates whether (x, y) can be correctly classified by h. The complement of φ is $\overline{\varphi}(x) = 1 - \varphi(x)$. Two simple properties of the conforming function will be used frequently in latter derivations, they are $\varphi(x)\varphi(x) = \varphi(x)$ and $\varphi(x)\overline{\varphi}(x) = 0$. The accuracy of h on distribution D is defined to be

$$c(h, D) = \sum_{k=1}^{m} D(k)\varphi(x_k) \tag{2}$$

which is the sum of weights of the examples that can be correctly classified by h.

For a sequence of hypotheses $h_1, \cdots, h_n$, their conforming functions are denoted by $\varphi_1, \varphi_2, \cdots, \varphi_n$. Given a distribution D, if for any arbitrary nonempty set $S \subseteq \{1, \cdots, n\}$ the following equation holds,

$$\sum_{k=1}^{m} \Big(D(k) \prod_{i \in S} \varphi_i(x_k) \Big) = \prod_{i \in S} \Big(\sum_{k=1}^{m} D(k)\varphi_i(x_k) \Big) \tag{3}$$

then we call $h_1, \cdots, h_n$ to be *mutually independent* on D, or briefly *independent* on D. If a group of hypotheses are mutually independent on D, a generalization of equation (3) can be derived. That is, if we replace some φ_i by $\overline{\varphi_i}$ in both sides of (3), it does not change the validity of this equation.

Lemma 1 *If $h_1, \cdots, h_n$ are mutually independent on distribution D, then for two arbitrary sets S^+, S^- such that $S^+ \cup S^- \subseteq \{1, \cdots, n\}$ and $S^+ \cap S^- = \emptyset$, the following equation holds*

$$\begin{aligned} &\sum_{k=1}^{m} \Big(D(k) \prod_{i \in S^+} \varphi_i(x_k) \prod_{j \in S^-} \overline{\varphi_j}(x_k) \Big) \\ = &\prod_{i \in S^+} \Big(\sum_{k=1}^{m} D(k)\varphi_i(x_k) \Big) \prod_{j \in S^-} \Big(\sum_{k=1}^{m} D(k)\overline{\varphi_j}(x_k) \Big) \end{aligned} \tag{4}$$

This property of independency will be used in some derivations in Section 3. A corollary can be derived directly from Lemma 1. We show it as follow.

Corollary 1 *If $h_1, \cdots, h_{n+1}$ are mutually independent on distribution D, then for two arbitrary sets S^+, S^- such that $S^+ \cup S^- \subseteq \{1, \cdots, n\}$ and $S^+ \cap S^- = \emptyset$, the following equation holds*

$$\begin{aligned} &\frac{\sum_{k=1}^{m} \Big(D(k)\varphi_{n+1}(x_k) \prod_{i \in S^+} \varphi_i(x_k) \prod_{j \in S^-} \overline{\varphi_j}(x_k) \Big)}{\sum_{k=1}^{m} \Big(D(k) \prod_{i \in S^+} \varphi_i(x_k) \prod_{j \in S^-} \overline{\varphi_j}(x_k) \Big)} \\ = &\sum_{k=1}^{m} D(k)\varphi_{n+1}(x_k). \end{aligned} \tag{5}$$

Given two sets S^+, S^- specified in Corollary 1, a set of hypotheses can be constructed by $V = \{h_i | i \in S^+\} \cup \{-h_j | j \in S^-\}$. The denominator of (5) is the proportion of the examples that satisfy all hypotheses in V on distribution D, and the numerator of (5) is the proportion of the examples that satisfy all hypotheses in $V \cup \{h_{n+1}\}$. Therefore, the left part of (5) is a conditional accuracy, which indicates the accuracy of h_{n+1} on D under the condition that all hypotheses in V are satisfied. Corollary 1 states that if

h_{n+1} is independent with $h_1, \cdots, h_n$ on D, this conditional accuracy is always equal to the general accuracy of h_{n+1}, independent of the selection of S^+ and S^- . This corollary describes an important property of independency.

We introduce the concept of *correlation coefficient* in probability theory in order to measure the correlation of two hypotheses. Define the correlation coefficient of h_i and h_j on distribution D to be

$$\rho(h_i, h_j, D) = \frac{\sum_{k=1}^{m} D(k)\varphi_i(x_k)\varphi_j(x_k) - \sum_{k=1}^{m} D(k)\varphi_i(x_k) \sum_{k=1}^{m} D(k)\varphi_j(x_k)}{\sqrt{Var(\varphi_i, D)Var(\varphi_j, D)}} \quad (6)$$

where $Var(\varphi, D)$ is the variance of $\varphi(x)$ on D:

$$Var(\varphi, D) = \sum_{k=1}^{m} D(k)\varphi^2(x_k) - \Big(\sum_{k=1}^{m} D(k)\varphi(x_k)\Big)^2. \quad (7)$$

The correlation coefficient ρ is a real number lies between -1 and $+1$. The numerator part of (6) shows that h_i is independent with h_j on D if and only if their correlation coefficient is 0. It is easy to see that $\rho(h_i, h_j, D) = -\rho(-h_i, h_j, D)$, so h_i is independent with h_j on D if and only if $\rho(h_i, h_j, D) = \rho(-h_i, h_j, D)$, which means that h_i and $-h_i$ have the same degree of correlation with h_j. Intuitively, h_i is independent with h_j if h_j inclines neither to h_i nor to $-h_i$.

Now we return to the problem of boosting. Consider the weak learner W. Using W, a hypothesis h_i can be generated on an arbitrary distribution D_i. The effectiveness of W guarantees that h_i has relatively high accuracy on D_i. Given a sequence of existing hypotheses $h_1, \cdots, h_n$ and a distribution D_{n+1}, a new hypothesis can be generated on D_{n+1} using W. If the $n+1$ hypotheses $h_1, \cdots, h_n, h_{n+1}$ are mutually independent on D_{n+1}, then we say that the group of hypotheses $\{h_1, \cdots, h_n\}$ is *indiscernible* on distribution D_{n+1}, for weak learner W. Indiscernibility is a core concept in this paper. This concept holds two meanings. First, $h_1, \cdots, h_n$ have to be mutually independent on D_{n+1} themselves. Second, for an example sampled randomly by D_{t+1}, the probability that it can be correctly classified by h_{n+1} is entirely uncorrelated to whether it was correctly classified by $h_1, \cdots, h_n$, which means that the weak learner W is not influenced by any one of $h_1, \cdots, h_n$ when the new hypothesis is trained on D_{n+1}. The mathematical explanation for this property was showed in Corollary 1.

In boosting algorithms, the sequence of distributions and hypotheses $D_1, h_1, D_2, h_2, \cdots$ are generated sequentially, from the original distribution D_1 using weak learner W. In this process, we hope that for an arbitrary n, the group of hypotheses $\{h_1, \cdots, h_n\}$ is always indiscernible on D_{n+1}. If this requirement is satisfied, it can be proved that the sequence of hypotheses generated in this process are mutually independent on D_1, and their accuracies on D_1 are as high as their training accuracies. We hope to find an extractor to achieve this, but it is not a simple task, even the feasibility of this can not be guaranteed theoretically. However, if we introduce an assumption according to the behavior of the weak learner, it is possible to do so. In next section, we will discuss this in details.

3 The Independency Assumption and AdaBoost

In this section, an extractor and a combiner will be designed based on the principle of independency. Given a weak learner W, we try to find such a method, which uses the existing distributions $D_1, \cdots, D_n$ and hypotheses $h_1, \cdots, h_n$ to construct a new distribution D_{n+1}, such that $\{h_1, \cdots, h_n\}$ is indiscernible on D_{n+1}. The algorithm that construct a sequence of distributions and hypotheses in this way is called an *extractor*. According to the definition in Section 2, the indiscernibility of a group of hypotheses on distribution D_{n+1} can be verified only after h_{n+1} is generated. However, our target is to construct such a D_{n+1} before the training of h_{n+1} begins, so it is necessary to find another criterion for indiscernibility.

If a group of hypotheses $\{h_1, \cdots, h_n\}$ is indiscernible on D_{n+1}, we first know that $h_1, \cdots, h_n$ are mutually independent on D_{n+1}. Moreover, for an arbitrary $h_i \in \{h_1, \cdots, h_n\}$, h_i has to be independent with h_{n+1}, which means $\rho(h_i, h_{n+1}, D_{n+1}) = \rho(-h_i, h_{n+1}, D_{n+1})$. We notice that every example in the instance space must conform to one and only one of h_i and $-h_i$. So if we require the two opposite hypotheses to have the same degree of correlation with h_{n+1}, a very intuitive idea is to force these two hypotheses having the same accuracy on D_{n+1}, which means that we require the accuracy of h_i on D_{n+1} to be $\frac{1}{2}$. To explain this idea, consider the situation that h_i and $-h_i$ have distinct accuracies. Without loss of generality, we can assume that $c(h_i, D_{n+1}) > c(-h_i, D_{n+1})$. Remember that the weak learner W has an instinctive tendency to choose stronger hypotheses, so if h_i is stronger than $-h_i$ on D_{n+1}, the new hypothesis trained on D_{n+1} should has a very high probability to be more similar to h_i, and less similar to its opposite $-h_i$, which means $\rho(h_i, h_{n+1}, D) > \rho(-h_i, h_{n+1}, D)$. This contradict with the fact that $\rho(h_i, h_{n+1}, D) = \rho(-h_i, h_{n+1}, D)$. Therefore, it is reasonable to require the accuracy of h_i on distribution D_{n+1} to be $\frac{1}{2}$. By considerations above, we introduce an assumption as follow.

Assumption 1 (The independency assumption) *If a group of hypotheses $h_1, h_2, \cdots, h_n$ are mutually independent on distribution D_{n+1}, and their accuracies on D_{n+1} are all $\frac{1}{2}$, then $\{h_1, h_2, \cdots, h_n\}$ is indiscernible on D_{n+1}.*

Assumption 1 is based on the fact that the weak learner W always prefers stronger hypotheses. We have not considered any other factors that affect the behavior of W in this assumption, so it is in its simplest form, we call this *the naive version of independency assumption*. Assumption 1 is composed of two parts: the condition part, which requires $h_1, \cdots, h_n$ to be mutually independent on D_{n+1} and have accuracy $\frac{1}{2}$; and the conclusion part, which asserts that $\{h_1, h_2, \cdots, h_n\}$ is indiscernible on D_{n+1}. Assumption 1 only states the logic relation between the condition part and the conclusion part, but never asserts if these two parts are true or not. However, we will see that, under the premise that Assumption 1 is true, an extractor can be designed to satisfy the condition part, and then guarantees that the conclusion part always holds.

In this section, we adopt the correctness of Assumption 1, and use it as a fact. Our task is to design a T rounds extractor to make sure that the condition part of the assumption holds in each rounds of training. As before, we denote the sequence of distributions by $D_1, \cdots, D_T$, they are constructed sequentially in T rounds of training. Denote the hypotheses trained from these distributions by $h_1, \cdots, h_T$, and denote their conforming functions by $\varphi_1, \varphi_2, \cdots, \varphi_T$. For $1 \leq t \leq T$, the accuracy of h_t on D_t(its training accuracy) is written as c_t. Consider the extractor which constructs a sequence of distributions by the following recursion:

$$\begin{aligned} &D_1(k) = \tfrac{1}{m} \\ &D_{t+1}(k) = \begin{cases} \frac{D_t(k)}{2c_t} & \varphi_t(x_k) = 1 \\ \frac{D_t(k)}{2(1-c_t)} & \varphi_t(x_k) = 0 \end{cases} \\ &(1 \leq t \leq T-1) \end{aligned} \tag{8}$$

This recursion has another form which is more commonly used:

$$\begin{aligned} &D_1(k) = \tfrac{1}{m} \\ &D_{t+1}(k) = \tfrac{D_t(k)}{2c_t}\varphi_t(x_k) + \tfrac{D_t(k)}{2(1-c_t)}\overline{\varphi_t}(x_k) \\ &(1 \leq t \leq T-1) \end{aligned} \tag{9}$$

Under this recursion, it is easy to see that $c(h_t, D_{t+1}) = \frac{1}{2}$, this is exactly what the design of this recursion aims to. The following two lemmas prove the validity and rationality of the distributions generated according to this recursion. Lemma 2 states that all of these distributions meet the condition of normalization; Lemma 3 states that the condition part of Assumption 1 is satisfied in each round of training. The proof of Lemma 2 is omitted because of its simplicity.

Lemma 2 *The sequence of distributions $D_1, \cdots, D_T$ construced by recursion (8) meet the normalization condition, that is, for an arbitrary $1 \leq t \leq T$, $\sum_{k=1}^{m} D_t(k) = 1$.*

Lemma 3 *If $D_1, \cdots, D_n$ are constructed by recursion (8), then for an arbitrary $1 \leq t \leq T-1$, hypotheses $h_1, \cdots, h_t$ are mutually independent on D_{t+1} and their accuracies are all $\frac{1}{2}$ on D_{t+1}.*

Proof We prove Lemma 3 by induction. When $t = 1$, we only need to prove $c(h_1, D_2) = \frac{1}{2}$, this can be concluded directly from (8). When $2 \leq t \leq T-1$, assume that the conclusion holds for $t-1$, which means that $h_1, \cdots, h_{t-1}$ are mutually independent on D_t, and their accuracies on D_t are all $\frac{1}{2}$. By Assumption 1, we also know that $h_1, \cdots, h_t$ are mutually independent on D_t. Thus for $i = t$,

$$\begin{aligned} & c(h_t, D_{t+1}) = \sum_{k=1}^{m} D_{t+1}(k)\varphi_t(x_k) \\ = & \sum_{k=1}^{m} \Big(\tfrac{1}{2c_t} D_t(k)\varphi_t(x_k)\varphi_t(x_k) + \tfrac{1}{2(1-c_t)} D_t(k)\overline{\varphi_t}(k)\varphi_t(x_k) \Big) \\ = & \tfrac{1}{2c_t} \sum_{k=1}^{m} D_t(k)\varphi_t(x_k) = \tfrac{1}{2} \end{aligned}$$

And for $1 \leq i \leq t-1$,

$$\begin{aligned} & c(h_i, D_{t+1}) = \sum_{k=1}^{m} D_{t+1}(k)\varphi_i(x_k) \\ = & \tfrac{1}{2c_t} \sum_{k=1}^{m} D_t(k)\varphi_t(x_k)\varphi_i(x_k) \\ & + \tfrac{1}{2(1-c_t)} \sum_{k=1}^{m} D_t(k)\overline{\varphi_t}(k)\varphi_i(x_k) \\ = & \Big(\tfrac{1}{2c_t} \sum_{k=1}^{m} D_t(k)\varphi_t(x_k) + \tfrac{1}{2(1-c_t)} \sum_{k=1}^{m} D_t(k)\overline{\varphi_t}(k) \Big) \\ & \cdot \Big(\sum_{k=1}^{m} D_t(k)\varphi_i(x_k) \Big) = \tfrac{1}{2} \end{aligned}$$

This proves the conclusion that $h_1, \cdots, h_t$ all have accuracy $\frac{1}{2}$ on D_{t+1}. Next we prove their independencies. For an arbitrary nonempty set $S \subseteq \{1, \cdots, t\}$, if $t \in S$ then

$$\begin{aligned} & \sum_{k=1}^{m} \Big(D_{t+1}(k) \prod_{i \in S} \varphi_i(x_k) \Big) \\ = & \sum_{k=1}^{m} \Big(D_{t+1}(k)\varphi_t(x_k) \prod_{i \in S-\{t\}} \varphi_i(x_k) \Big) \\ = & \tfrac{1}{2c_t} \sum_{k=1}^{m} \Big(D_t(k)\varphi_t(x_k) \prod_{i \in S-\{t\}} \varphi_i(x_k) \Big) \\ = & \tfrac{1}{2c_t} \prod_{i \in S} \Big(\sum_{k=1}^{m} D_t(k)\varphi_i(x_k) \Big) \\ = & \tfrac{1}{2^{|S|}} = \prod_{i \in S} \Big(\sum_{k=1}^{m} D_{t+1}(k)\varphi_i(x_k) \Big) \end{aligned}$$

Notice that the last equation above uses the conclusion we have just proved, which states that $c(h_i, D_{t+1}) = \frac{1}{2}$ for $1 \leq i \leq t$. The similar techniques can be applied to the cases when $t \notin S$:

$$\begin{aligned} & \sum_{k=1}^{m} \Big(D_{t+1}(k) \prod_{i \in S} \varphi_i(x_k) \Big) \\ = & \tfrac{1}{2c} \sum_{k=1}^{m} \Big(D_t(k)\varphi_t(x_k) \prod_{i \in S} \varphi_i(x_k) \Big) \\ & + \tfrac{1}{2(1-c_t)} \sum_{k=1}^{m} \Big(D_t(k)\overline{\varphi_t}(x_k) \prod_{i \in S} \varphi_i(x_k) \Big) \\ = & \Big(\tfrac{1}{2c} \sum_{k=1}^{m} D_t(k)\varphi_t(x_k) + \tfrac{1}{2(1-c_t)} \sum_{k=1}^{m} D_t(k)\overline{\varphi_t}(x_k) \Big) \\ & \cdot \prod_{i \in S} \Big(\sum_{k=1}^{m} D_t(k)\varphi_i(x_k) \Big) \\ = & \tfrac{1}{2^{|S|}} = \prod_{i \in S} \Big(\sum_{k=1}^{m} D_{t+1}(k)\varphi_i(x_k) \Big) \end{aligned}$$

In this way, we proved the conclusion for t. This completes the induction. □

Corollary 2 *If $D_1, \cdots, D_n$ are constructed by recursion (8), then for arbitrary $1 \leq t \leq T$, hypotheses $h_1, \cdots, h_t$ are mutually independent on D_t.*

According to Lemma 2 and Lemma 3, we have already obtained a valid extractor. This extractor maintains a sequence of distributions $D_1, \cdots, D_T$ according to recursion (8), and generate a hypothesis h_t from D_t in each round t using the weak learner W. In this process, the conclusion part of Assumption 1 always holds. We call this extractor as *the naive extractor* since it is designed to match the naive version of independency assumption. All distributions $D_1, \cdots, D_T$ and hypotheses $h_1, \cdots, h_T$ discussed in the latter part of this section are generated by the naive extractor. Our wish is to *lift* the independencies and accuracies of $h_1, \cdots, h_T$ up to the original distribution D_1, the following theorem shows that we can achieve this target.

Theorem 1 *When T rounds of training are finished, for arbitrary (i, t) satisfying $1 \leq i, t \leq T$, the accuracy of h_i on D_t is*

$$c(h_i, D_t) = \begin{cases} c_i & i \geq t \\ \frac{1}{2} & i < t \end{cases} \tag{10}$$

and $h_1, \cdots, h_T$ are mutually independent on D_t.

Proof We prove Theorem 1 by backward induction. When $t = T$, the proof is provided by the conclusion of Lemma 3 and Corollary 2. For $1 \leq t \leq T-1$, assume that the conclusion holds for $t+1$. Consider the accuracy of h_i

on D_t: if $i < t$, $c(h_i, D_t) = \frac{1}{2}$ by theorem 1; if $i = t$, $c(h_i, D_t) = c_i$ by the definition of c_i; if $i \geq t+1$, according to the inductive assumption, we know that $c(h_i, D_{t+1}) = c_i$, $c(h_t, D_{t+1}) = \frac{1}{2}$, and h_i is independent with h_t on D_{t+1}. So,

$$\begin{aligned}
&\sum_{k=1}^{m} D_t(k)\varphi_i(x_k) \\
=\ &\sum_{k=1}^{m} D_t(k)\varphi_t(x_k)\varphi_i(x_k) + \sum_{k=1}^{m} D_t(k)\overline{\varphi_t}(x_k)\varphi_i(x_k) \\
=\ &2c_t \sum_{k=1}^{m} D_{t+1}(k)\varphi_t(x_k)\varphi_i(x_k) \\
&+2(1-c_t)\sum_{k=1}^{m} D_{t+1}(k)\overline{\varphi_t}(x_k)\varphi_i(x_k) \\
=\ &\Big(2c_t \sum_{k=1}^{m} D_{t+1}(k)\varphi_t(x_k) + \\
&2(1-c_t)\sum_{k=1}^{m} D_{t+1}(k)\overline{\varphi_t}(x_k)\Big)\sum_{k=1}^{m} D_{t+1}(k)\varphi_i(x_k) \\
=\ &\sum_{k=1}^{m} D_{t+1}(k)\varphi_i(x_k) = c_i
\end{aligned}$$

This proves that the accuracy of h_i on D_t is coincident with (10). Notice that $\sum_{k=1}^{m} D_t(k)\varphi_i(x_k) \neq \sum_{k=1}^{m} D_{t+1}(k)\varphi_i(x_k)$ only if $i = t$, thus for an arbitrary nonempty set $S \subseteq \{1, \cdots, T\}$, if $t \in S$ we have

$$\begin{aligned}
&\sum_{k=1}^{m}\Big(D_t(k)\prod_{i\in S}\varphi_i(x_k)\Big) \\
=\ &\sum_{k=1}^{m}\Big(D_t(k)\varphi_t(x_k)\prod_{i\in S-\{t\}}\varphi_i(x_k)\Big) \\
=\ &2c_t\sum_{k=1}^{m}\Big(D_{t+1}(k)\varphi_t(x_k)\prod_{i\in S-\{t\}}\varphi_i(x_k)\Big) \\
=\ &2c_t\prod_{i\in S-\{t\}}\Big(\sum_{k=1}^{m} D_{t+1}(k)\varphi_i(k)\Big) \\
=\ &2c_t\Big(\sum_{k=1}^{m} D_{t+1}(k)\varphi_t(k)\Big)\cdot\prod_{i\in S-\{t\}}\Big(\sum_{k=1}^{m} D_t(k)\varphi_i(k)\Big) \\
=\ &\prod_{i\in S}\Big(\sum_{k=1}^{m} D_t(k)\varphi_i(k)\Big)
\end{aligned}$$

If $t \notin S$,

$$\begin{aligned}
&\sum_{k=1}^{m}\Big(D_t(k)\prod_{i\in S}\varphi_i(x_k)\Big) \\
=\ &2c_t\sum_{k=1}^{m}\Big(D_{t+1}(k)\varphi_t(x_k)\prod_{i\in S}\varphi_i(x_k)\Big) \\
&+2(1-c_t)\sum_{k=1}^{m}\Big(D_{t+1}(k)\overline{\varphi_t}(x_k)\prod_{i\in S}\varphi_i(x_k)\Big) \\
=\ &\Big(2c_t\sum_{k=1}^{m} D_{t+1}(k)\varphi_t(x_k) + \\
&2(1-c_t)\sum_{k=1}^{m} D_{t+1}(k)\overline{\varphi_t}(x_k)\Big)\prod_{i\in S}\Big(\sum_{k=1}^{m} D_{t+1}(k)\varphi_i(x_k)\Big) \\
=\ &\prod_{i\in S}\Big(\sum_{k=1}^{m} D_{t+1}(k)\varphi_i(x_k)\Big) \\
=\ &\prod_{i\in S}\Big(\sum_{k=1}^{m} D_t(k)\varphi_i(x_k)\Big)
\end{aligned}$$

In this way the independency of $h_1, \cdots, h_T$ on D_t is proved, this completes the induction. □

Corollary 3 *On distribution D_1, all of the hypotheses $h_1, \cdots, h_T$ are mutually independent. For an arbitrary hypothesis $h_i \in \{h_1, \cdots, h_T\}$, $c(h_i, D_1) = c_i$.*

Corollary 3 shows that the naive extractor is exactly what we want, since all hypotheses generated by this extractor are mutually independent on D_1, and their accuracies on D_1 are equal to their training accuracies, which are relatively high.

Another algorithm is still needed to combine these independent hypotheses into a strong final classifier, this algorithm can be referred as a *combiner*. A combiner receives a set of T bits strings $(h_1(x_k), \cdots, h_T(x_k))_{k=1,\cdots,m}$ as input on distribution D_1, and tries to output a correct label y_k for each k. This is also a learning problem, so the combiner can be constructed by some other learning algorithms. In this paper, we only consider the simplest form of the combiner, which is a weighted majority vote of weak hypotheses.

Based on the fact that $h_1, \cdots, h_T$ are mutually independent, the combiner can be designed in a statistical perspective as follow. Define random variables $u_i = yh_i(x)$ for $1 \leq i \leq T$, (x, y) is an example randomly picked from the instance space according to distribution D_1. Notice that $yh_i(x) = 2\varphi_i(x) - 1$, so u_i has c_i probability to be $+1$, and $1 - c_i$ probability to be -1. Furthermore, $u_1, \cdots, u_T$ are mutually independent random variables. If the final output of the combiner is determined by a weighted majority vote, namely $H(x) = \mathrm{sgn}(\sum_{i=1}^{T} \lambda_i h_i(x))$, then the accuracy of the final classifier is exactly the expectation value of $\frac{H(x)+1}{2}$, which is equal to $E(\frac{\mathrm{sgn}(\sum_{i=1}^{T}\lambda_i u_i)+1}{2})$. Considering the inequity $1 - \frac{\mathrm{sgn}(x)+1}{2} \leq e^{-x}$, we can estimate the error rate of the final classifier as:

$$\begin{aligned}
&E\Big(1 - \tfrac{1}{2}\Big(\mathrm{sgn}\Big(\sum_{i=1}^{T}\lambda_i u_i\Big) + 1\Big)\Big) \\
\leq\ &E\Big(\exp\Big(-\sum_{i=1}^{T}\lambda_i u_i\Big)\Big) \\
=\ &\prod_{i=1}^{T} E\Big(\exp(-\lambda_i u_i)\Big) \\
=\ &\prod_{i=1}^{T}\Big((1-c_i)\exp(\lambda_i) + c_i\exp(-\lambda_i)\Big)
\end{aligned}$$

The last term reaches its minimum value when $\lambda_i = \frac{1}{2}\ln\Big(\frac{c_i}{1-c_i}\Big)$, this minimum value is $\prod_{i=1}^{T}\Big(2\sqrt{c_i(1-c_i)}\Big)$. The combiner using such λ_i as combination coefficients is called *the naive combiner*, because it merges the hypotheses extracted by a naive extractor into a simplest form of the final classifier. If there exist a constant $\delta > 0$ such that $c_i > \frac{1}{2} + \delta$, the error rate of the final classifier can be upper bounded by $\Big(\sqrt{1-4\delta^2}\Big)^T$. This is coincident with the result of (Freund & Schapire, 1995) for AdaBoost. Actually, Adaboost is precisely such an algorithm that composed by a naive extractor and a naive combiner which we designed above. Therefore, AdaBoost is the algorithm derived from the naive version of independency assumption.

Theorem 2 *The AdaBoost algorithm is composed by a naive extractor and a naive combiner.*

Proof To prove the theorem, it is sufficient to prove that Adaboost uses the same recursion as the naive extractor, and uses the same combination coefficients as the naive combiner. The naive extractor's recursion (8) can also be written as:

$$D_{t+1}(k) = \begin{cases} \frac{1}{2\sqrt{c_i(1-c_i)}}\sqrt{\frac{1-c_i}{c_i}}D_t(k) & y_k h_t(x_k) = 1 \\ \frac{1}{2\sqrt{c_i(1-c_i)}}\sqrt{\frac{c_i}{1-c_i}}D_t(k) & y_k h_t(x_k) = -1 \end{cases}$$

Table 1: Datasets Characteristics

dataset	# examples	# classes	# attributes
Credit Approval(crx)	690	2	15
Heart Disease(heart)	920	2	35
Pima Indians Diabetes(pima)	768	2	8

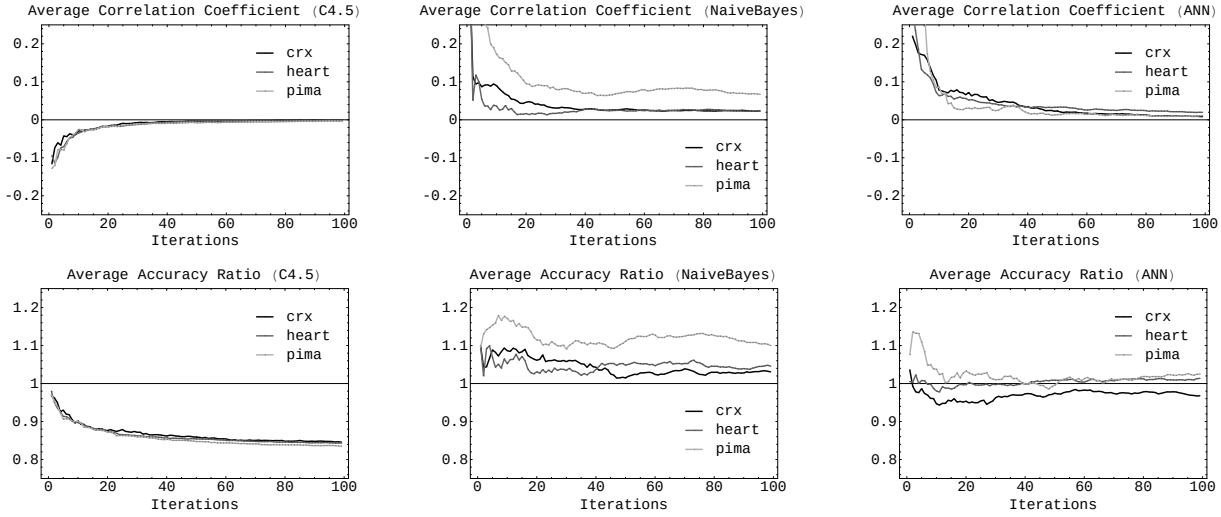

Figure 2: Average Correlation Coefficients and Average Accuracy Ratios.

(11)

We set $\alpha_t = \frac{1}{2}\ln\left(\frac{c_i}{1-c_i}\right)$ and $Z_t = 2\sqrt{c_i(1-c_i)}$, then (11) is equivalent to:

$$D_{t+1}(k) = \frac{D_t(k)\exp(-\alpha_t y_k h_t(x_k))}{Z_t} \tag{12}$$

which is precisely the recursion of AdaBoost. For combination coefficients, the naive combiner's coefficients $\lambda_i = \frac{1}{2}\ln\left(\frac{c_i}{1-c_i}\right)$ is also the same as AdaBoost. So the conclusion is proved. □

According to Corollary 3 and Theorem 2, AdaBoost has two advantages if Assumption 1 is true. They are: all weak hypotheses are mutually independent on the original distribution, and their accuracies on this distribution are as high as their training accuracies. This explains why AdaBoost is efficient on improving the performance of weak learners. On the other hand, Assumption 1 can also be used as a criterion on evaluating whether a weak learner W is suitable to be used for boosting through AdaBoost. Because if W does not satisfy Assumption 1, we can not guarantee that AdaBoost works efficiently on W.

4 Experiments and Analysis

In this section, some experiments on real datasets are presented to verify the conclusions in Section 3. We want to test the adaptability of Assumption 1 on three kinds of weak learners: Quinlan's C4.5 decision-tree algorithm(Quinlan, 1993), Naive Bayes Classifier and Artificial Neural Networks(ANN). Instead of measuring Assumption 1 itself, we test two properties derived form this assumption, which are stated in Corollary 3. According to Corollary 3, for an arbitrary iteration t of AdaBoost, the weak hypotheses $h_1, \cdots, h_t$ are mutually independent on D_1 and their accuracies on D_1 are equal to their training accuracies, if Assumption 1 is true. In our experiment, two quantities are measured after each round t of AdaBoost. One is the *average correlation coefficient*, defined by

$$C(t) = \frac{\sum_{1\le i<j\le t}\rho(h_i, h_j, D_1)}{t(t-1)/2}. \tag{13}$$

The other one is the *average accuracy ratio*, defined by

$$R(t) = \frac{\sum_{i=1}^{t} c(h_i, D_1)}{\sum_{i=1}^{t} c_i}. \tag{14}$$

Corollary 3 asserts that these two values should be 0 and 1 respectively for each t.

Three datasets from UCI Machine Learning Repository are tested on each of the three weak learners. The detail information of these datasets are showed in Table 1. We use RPROP(Riedmiller & Braun, 1993) algorithm to train neural networks, each neural network contains a single hidden layer with six hidden nodes, and it takes 100 epoches of training in each round of boosting. All of the three datasets are converted to the PROBEN1(Prechelt, 1994) format when they are applied to ANN weak learners. Both of the C4.5 weak learner and the Naive Bayes weak learner are used with default options. In each round of boosting, the training exam-

ples are chosen independently at random according to the distribution D_t. The number of examples sampled in each round is exactly equal to the size of the dataset. This sampling method can be referred as *resampling*. Results of these experiments are showed in Figure 2.

According to these results, the average independencies are satisfied well by the three weak learners. When t is small, the average correlation coefficient inclines to be negative on C4.5, and positive on Naive Bayes and ANN. As t grows large, it converges to a constant value very close to zero, except for the pima dataset on the Naive Bayes weak learner. As for accuracies, ANN weak learners keep the average accuracy ratios very close to 1, Naive Bayes hold this ratio slightly larger than 1 on crx and heart dataset, and about 10% larger than 1 on pima dataset. For C4.5 weak learners, this ratio converges to a value about 15% less than 1 on all of the three datasets.

We can make some analysis on these results. All of the three weak learners keep the correlation coefficients very close to zero on almost all datasets, which means that AdaBoost works well on extracting independent rules. The average accuracy ratio almost reaches its theoretical values for ANN weak learners, and holds some deviations for C4.5 and Naive Bayes. However, theses deviations are at most 15%, so accuracies are not changed too much when they are lifted. In general, the experiment confirms that it is reasonable to apply Assumption 1 on these weak learners. On the other hand, this result also gives an experimental verification to the effectiveness of Adaboost.

The experiment suggests that there exists some relationship between the independencies and accuracies of weak hypotheses. In fact, a corollary can be presented to describe this relationship in a mathematical way. This corollary does not depend on Assumption 1.

Corollary 4 *For* $1 \leq t \leq T$, $c(h_t, D_1) - c_t = \sum_{i=1}^{t-1} \rho(h_i, h_t, D_{i+1})(2c_i - 1)\sqrt{Var(\varphi_t, D_{i+1})}$.

The proof of Corollary 4 relies on the definition of recursion (8) and the definition of correlation coefficient (6), we omit the proof here. From this corollary it can be seen that, if $c_i > \frac{1}{2}$ for $1 \leq i \leq T$, the value of $c(h_t, D_1) - c_t$ is seriously influenced by the signs of $\rho(h_i, h_t, D_{i+1})$. If all hypotheses are mutually independent, $\rho(h_i, h_t, D_{i+1})$ is always equal to 0, which means that $c(h_t, D_1) = c_t$. This is coincident with the conclusion of Corollary 3. However, if these hypotheses have an inclination to negative correlations, the lifted accuracy $c(h_t, D_1)$ is very likely to be lower than c_t, this explains the phenomenon observed on C4.5 weak learners. On the other hand, positive correlations may strengthen the weak hypotheses on D_1, as it is observed on the Naive Bayes weak learner when the pima dataset is trained. Nevertheless, we can not conclude that positive correlations are beneficial. Actually, if a group of weak hypotheses are positive correlated, it means that there exist some similarities among these hypotheses, and these similarities will cause the inefficiency of the boosting algorithm. For instance, "bagging"(Breiman, 1994) is a typical algorithm that generates positive correlated hypotheses. (Freund & Schapire, 1996) showed that bagging performs significantly and uniformly worse than Adaboost.

In summery, Corollary 4 shows that both of the positive correlations and negative correlations bring some bad effects to the boosting process, so keeping independency is a sensible strategy. Therefore, we can measure the efficiency of a boosting algorithm on weak learners or datasets by measuring the independency of generated weak hypotheses, this method is instructive for the design and analysis of boosting algorithms.

5 Conclusions and Open Problems

In this paper, we proposed a new way to design boosting algorithms. We divide a boosting algorithm into an extractor and a combiner. Our target is to extract a sequence of high-accuracy weak rules which are mutually independent on the original distribution, then merge theses hypotheses into a strong classifier. We found that an assumption is needed to get a practical boosting algorithm, this assumption is called the independency assumption. Our algorithm is composed of a naive extractor and a naive combiner, derived from the naive version of independency assumption in Section 3. We point out that this algorithm is equivalent to AdaBoost exactly. In other words, we give an explanation to the mechanism of AdaBoost. Experiments are arranged in Section 4 to test the independencies and accuracies of weak rules generated by AdaBoost on three weak learners, which verify the adaptability of the independency assumption indirectly. Results on these experiments can also be used as criterions to evaluate whether a weak learner can be improved efficiently by boosting algorithms.

This paper represents the first work that systematically introduced the concept of independency into boosting, so there are many open problems remained. Experiments in Section 4 show that the accuracies on original distributions are weakened for AdaBoost when using C4.5 as the weak learner. It is promising to solve this problem by using a more complex version of independency assumption and derive an improved version of AdaBoost from this new assumption. The independency assumption for two-class problems may also be generalized into multi-class or continuous-valued versions to design boosting algorithms that can be applied to these cases. The statistical method to estimate the error rate of final classifiers may also worth some more research.

Acknowledgement

The research work in this paper has been supported by China National Natural Science Foundation (No.60673008 and No.60973100).

References

Breiman, L. 1994. Bagging predictors. *Technical Report 421*.

Freund, Y., and Schapire, R. E. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. *In MSRI Workshop on Nonlinear Estimation and Classification*.

Freund, Y., and Schapire, R. E. 1996. Experiments with a new boosting algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference*.

Friedman, J.; Hastie, T.; and Tibshirani, R. 2000. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics* 38(2):337–374.

Kearns, M., and Valiant, L. G. 1988. Learning boolean formulae or finite automata is as hard as factoring. *Technical Report TR-14-88* 586–591.

Kearns, M., and Valiant, L. G. 1994. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the Association for Computing Machinery* 41(1):67–95.

Kivinen, J., and Warmuth, M. K. 1999. Boosting as entropy projection. 134–144.

Prechelt, L. 1994. Proben1 - a set of neural network benchmark problems and benchmarking rules.

Quinlan, J. R. 1993. C4.5: Programs for machine learning.

Riedmiller, M., and Braun, H. 1993. A direct adaptive method for faster backpropagation learning: The rprop algorithm. *In Proceedings of the IEEE International Conference on Neural Networks* 586–591.

Schapire, R. E. 2001. Drifting games. *Machine Learning* 43:265–291.

Schapire, R. E. 2002. The boosting approach to machine learning: An overview. *In MSRI Workshop on Nonlinear Estimation and Classification*.