Research Article
A Theoretical Analysis of Why Hybrid Ensembles Work

Kuo-Wei Hsu

Department of Computer Science, National Chengchi University, No. 64, Sec. 2, Zhi Nan Rd., Wen Shan District, Taipei City 11605, Taiwan

Correspondence should be addressed to Kuo-Wei Hsu; kwhsu@nccu.edu.tw

Received 8 August 2016; Revised 6 December 2016; Accepted 5 January 2017; Published 31 January 2017

Academic Editor: Jussi Tohka

Copyright © 2017 Kuo-Wei Hsu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Inspired by the group decision making process, ensembles or combinations of classifiers have been found favorable in a wide variety of application domains. Some researchers propose to use the mixture of two different types of classification algorithms to create a hybrid ensemble. Why does such an ensemble work? The question remains. Following the concept of diversity, which is one of the fundamental elements of the success of ensembles, we conduct a theoretical analysis of why hybrid ensembles work, connecting using different algorithms to accuracy gain. We also conduct experiments on classification performance of hybrid ensembles of classifiers created by decision tree and naïve Bayes classification algorithms, each of which is a top data mining algorithm and often used to create non-hybrid ensembles. Therefore, through this paper, we provide a complement to the theoretical foundation of creating and using hybrid ensembles.

1. Introduction

Ensemble learning is inspired by the human group decision making process and has gained much attention [1–7]. It is to create an ensemble, which is a group of classifiers and combines classifications made by these classifiers to make an overall classification. The advantage of ensembles is not that the best combination of classifiers outperforms the best classifier but that a combination of classifiers is less probable to misclassify unseen data samples than a single classifier. Ensembles have shown their satisfactory classification performance in a large scale comparative study [8].

Ensemble learning has been applied in various application domains, such as image classification [9–14], fingerprint classification [15], weather forecasting [16], text categorization [17], image segmentation [18], visual tracking [19], protein fold pattern recognition [20], cancer classification [21], pedestrian recognition [22] or detection [23], prediction of software quality [24, 25], face recognition [26], email filtering [27], prediction of students’ performance [28], medical image analysis [29–32], churn prediction [33], sentiment analysis [34–37], steganalysis [38], prediction of air quality [39], and intrusion detection [40].

From one point of view, ensemble learning becomes popular because every classification algorithm has its own limitations. From another point of view, if every classifier in an ensemble has expertise in classifying data samples that belong to some portion of the given data set, the overall classification combined from all (or some) classifiers in the ensemble will potentially be more reliable.

Compared to classifiers working individually, classifiers working together will have a better potential for gaining better accuracy [41]. Diverse classifiers working together will have a better potential for gaining better accuracy compared to non-diverse classifiers working together, as suggested in [42]. In creating an ensemble, every classifier that will be part of the ensemble is expected to be adequately accurate, while the correlation between classifications made by two classifiers that will be part of the ensemble is expected to be small.

Some researchers use a mixture of different types of classification algorithms to create hybrid ensembles. For example, with reference to ensembles composed of decision trees and artificial neural networks, Langdon et al. use such an ensemble in drug discovery [43]; Lu et al. discuss sampling methods along with these ensembles for active learning [44]; in [45], the author analyzes the hybrid ensembles for binary classification from the standpoint of bias-variance decomposition proposed in [46]. Furthermore, Salgado et al. use ensembles of artificial neural networks and support vector machines [47, 48] to predict daily electricity load [49].
Min and Cho use naïve Bayes classifiers and support vector machines for activity recognition [50]. Verikas et al. provide a survey of hybrid ensembles designed for bankruptcy prediction [51]. In addition, Verma and Hassan use integrations of clustering and classification algorithms to create hybrid ensembles [52].

Most researchers simply use hybrid ensembles without further investigation. Therefore, we plan to have a better understanding of hybrid ensembles. Our goal is not only to show that the classification performance of hybrid ensembles can be comparable or even superior to that of non-hybrid ensembles, but also to provide an explanation of why hybrid ensembles work from the standpoint of diversity. Diversity among classifiers in an ensemble plays a significant role in the success of the ensemble. One type of ensemble learning algorithms uses sampling methods to generate different data sets used for training diverse classifiers, such as Bagging (Bootstrap Aggregating) [53], while another type is rather ad hoc and uses different classification algorithms to train diverse classifiers that will be used to create an ensemble.

We create hybrid ensembles by using an integration of these two types of ensemble learning algorithms. The classification algorithms that we use to create hybrid ensembles are C4.5 [48, 54] decision tree and naïve Bayes [48]. We provide an empirical comparison of these hybrid ensembles and others created by using Bagging. This paper is particularly essential because quite few papers empirically evaluate hybrid ensembles and at the same time theoretically analyze them.

The rest of this paper is structured as follows: We provide background information and present our analysis in Section 2. Next, we report and discuss experiment results in Section 3. Finally, we give conclusion in Section 4.

2. Materials and Methods

The goal of using several classifiers in an ensemble is to achieve better classification performance by combining classifications from these classifiers, each of which serves as an optimal model or an expert in a portion of the data set. These classifiers are expected to be uncorrelated and behave independently of each other; or, at least, they need to show different patterns of errors.

Diversity among classifiers in an ensemble is related to the success of the ensemble, because it can compensate for errors made by those classifiers individually [55]. In this section, we analyze diversity and classification performance, for example, accuracy, of ensembles created by using a single algorithm (i.e., non-hybrid ensembles) and hybrid ensembles created by using two different algorithms. The relationship between diversity and accuracy is not "straightforward" [56]. On one hand, combining classifications from classifiers of low diversity would not improve the overall accuracy because these classifiers behave similarly for some portions of the data set; on the other hand, using highly diverse classifiers to create an ensemble would not guarantee absolutely high accuracy. Hsu and Srivastava show that using different classification algorithms in an ensemble would likely increase diversity and decrease correlation between classifiers in the ensemble [57], and they build the connection between diversity and correlation, which can be indirectly connected to accuracy [58]. Referring to the analysis technique used in [59], we build a more direct connection between diversity and accuracy, and further we analyze the influence of using different algorithms to create an ensemble on accuracy; the analysis distinguishes this paper from the earlier papers.

In what follows, \( x \) is a \( d \)-dimensional vector to represent a data sample, and \( y \) is a binary class label or \( y \in \{-1,1\} \). \( C \) is a classifier and \( C : x \in \mathbb{R}^d \rightarrow \hat{y} \in \{-1,1\} \), where \( \hat{y} \) is a classification made by a classifier. \( A \) is a classification algorithm. \( T \) is a set of data samples. \( C(x \mid A, T) \) means that \( C \) is trained by applying \( A \) on \( T \) and is used to classify \( x \), and it returns a binary class label.

What is described in (1) is a general form of an ensemble where classifications from classifiers are combined through majority voting to make the overall classification. It can be modified such that it uses weighted majority voting. In (1), \( C \) is an ensemble of \( n \) classifiers, \( C_i \) is a classifier in the ensemble, \( x \) is a data sample that needs to be classified, and \( v \) is the overall classification. For those ensemble learning algorithms only using different data sets to train diverse classifiers, \( A_i = A_j \) and \( T_i \neq T_j \) for all \( i \neq j \), where \( 0 \leq i, j \leq n \), and Bagging is an example. For those using different classification algorithms and different data sets to train diverse classifiers, \( A_i = A_j \) for some (or, in few cases, all) \( i \neq j \) and \( T_i \neq T_j \) for all \( i \neq j \), where \( 0 \leq i, j \leq n \), and this is of our interest in this paper.

\[
C(x) = \max_{v \in \{-1,1\}} \sum_{i=1}^{n} I(v = \hat{y}_i)
\]

\[
= \max_{v \in \{-1,1\}} \sum_{i=1}^{n} I(v = C_i(x \mid A_i, T_i)), \quad (1)
\]

where \( I \) is the indicator function.

Given a data set \( D \) and a set of classification algorithms \( G \), we create a hybrid ensemble of \( n \) classifiers, each of which is trained by applying an algorithm selected in an alternating fashion from \( G \) on a set of data samples drawn from \( D \) with bootstrap sampling. This process is shown in Algorithm 1. Instead of selecting algorithms in an alternating fashion, we can simply do random selection such that a classifier in the hybrid ensemble could be trained by using one of the algorithms in \( G \) with an equal probability. Extending from this, we can incorporate prior knowledge and assign unequal probabilities to different algorithms.

The input of the bootstrap sampling method is a data set \( D \), and the output is a set \( D_b \) of data samples drawn with replacement from \( D \), \( |D_b| = |D| \). We use bootstrap sampling when we need different data sets to train diverse classifiers in creating an ensemble. Diversity among the classifiers in such an ensemble totally comes from differences among the data sets used to train the classifiers, and therefore bootstrap sampling is the single generator of diversity in such an ensemble. We train diverse classifiers by additionally using different classification algorithms, and by doing this we have an additional generator of diversity.
Afterward, we present our analysis of why hybrid ensembles work based on the analysis technique used in [59]. We start from the following definitions, where \( \hat{y}^{(A,T)} = C(x | A,T) \) and \( \hat{y} \) is the classification to a testing data sample \( x \) given by a classifier \( C \) trained by applying \( A \) on \( T \).

**Definition 1.** \( D \) is a given data set and \( W \) is a set of data sets generated with bootstrap sampling from \( D \). The \( i \)th element in \( W \) is of the same size of \( D \); that is, \( |W_i| = |D| \). \( T_1 \in W \) and \( T_2 \in W \) are used as training data sets, where \( T_1 \neq T_2 \). With respect to \( S \in W \) used as a testing data set, where \( S \neq T_1 \) and \( S \neq T_2 \), T-Diversity of a classification algorithm \( A \) is defined as an expectation of disagreement between classifiers trained by applying \( A \) on \( T_1 \) and \( T_2 \), as given in

\[
T-\text{Diversity} = E_S \left[ \mathbb{1} \left( \hat{y}^{(A,T_1)} \neq \hat{y}^{(A,T_2)} \right) \right],
\]

where \( E_S \) is the expectation upon \( S \).

Definition 1 describes how unstable a classification algorithm could possibly be on a given data set. Definition 2 describes the degree of being unstable (or the instability) for a classification algorithm.

**Definition 2.** Based on Definition 1, a classification algorithm \( A \) is \((\alpha, \beta)\)-unstable with respect to \( T \)-Diversity, if the following holds:

\[
P_W \left[ E_S \left[ \mathbb{1} \left( \hat{y}^{(A,T_1)} \neq \hat{y}^{(A,T_2)} \right) \right] \geq \alpha \right] \geq \beta,
\]

where \( P_W \) is the probability upon \( W \).

Here, disagreement between classifiers is from using different training data sets, and it is a type of diversity. In (3), \( \beta \) is the lower bound of the probability that we can observe disagreement at least \( \alpha \); the larger the \( \alpha \), the larger the diversity. When \( \alpha \) is treated as a constant, a larger value of \( \beta \) means a more unstable classification algorithm.

We use the process given below to estimate the instability of an algorithm on a data set. The idea is to use two sampled data sets to train two classifiers and then measure the difference in classifications made by the two classifiers on another sampled data set. Because these data sets are from the same given data set, difference in classifications mainly comes from the algorithm used to train the classifiers. If the algorithm is more unstable and sensitive to the changes in the data sets used for training, the trained classifiers would make more different classifications on a data set, part of which has been seen by both classifiers. The larger the value returned by the process, the more unstable the algorithm.

**Definition 3.** \( D \) is a given data set and \( W \) is a set of data sets generated with bootstrap sampling from \( D \). The \( i \)th element in \( W \) is of the same size of \( D \); that is, \( |W_i| = |D| \). \( T \in W \) is used as a training data set. With respect to \( S \in W \) used as a testing data set, where \( S \neq T \), A-Diversity for two classification algorithm \( A_1 \) and \( A_2 \) is defined as an expectation of disagreement between classifiers trained by applying \( A_1 \) and \( A_2 \) on \( T \), as given in

\[
A-\text{Diversity} = E_S \left[ \mathbb{1} \left( \hat{y}^{(A_1,T)} \neq \hat{y}^{(A_2,T)} \right) \right],
\]

where \( E_S \) is the expectation upon \( S \).

Definition 3 describes how different two classification algorithms could possibly be on a given data set. Definition 4 describes the degree of being different (or the differentiability) for two classification algorithms.

**Definition 4.** Based on Definition 3, two classification algorithms \( A_1 \) and \( A_2 \) are \((\delta, \gamma)\)-differentiable with respect to A-Diversity, if the following holds:

\[
P_W \left[ E_S \left[ \mathbb{1} \left( \hat{y}^{(A_1,T)} \neq \hat{y}^{(A_2,T)} \right) \right] \geq \delta \right] \geq \gamma,
\]

where \( P_W \) is the probability upon \( W \).

Here, disagreement between classifiers is from using different classification algorithms, and it is a type of diversity, too. In (5), \( \gamma \) is the lower bound of the probability that we can observe disagreement at least \( \delta \); the larger the \( \delta \), the larger the diversity. Similarly, when \( \delta \) is treated as a constant, a larger value of \( \gamma \) means that two classification algorithms are more different.

We use the process given below to estimate the differentiability of a pair of algorithms on a data set. The idea is to use two algorithms and a sampled data set to train two classifiers and then measure the difference in classifications made by the two classifiers on another sampled data set. If the two algorithms are more differentiable, the classifiers trained with them would make more different classifications on a data set, part of which has been seen by both classifiers. The larger the
value returned by the process, the more differentiable the two algorithms.

We show a connection between diversity and accuracy gain for using a hybrid ensemble through the proposition given below.

**Proposition 1.** If two classification algorithms $A_1$ and $A_2$ are $(\delta, \gamma)$-differentiable and $A_1$ is $(\alpha_1, \beta_1)$-unstable, then (6) holds, where $\Delta E[\text{Acc}]$ is the difference between the expected accuracy of a hybrid ensemble with $A_1$ and $A_2$ and the expected accuracy of a non-hybrid ensemble with only $A_1$, and $\text{Acc}_i$ is the accuracy of the classifier training by using $A_i$.

\[
P \left[ \Delta E \left[ \text{Acc} \right] \geq \alpha_1 \cdot \delta \cdot (\text{Acc}_2 - \text{Acc}_1) \right] \geq 1 - (1 - \beta_1) \cdot (1 - \gamma).
\]  

(6)

Regarding (6), $A_1$ is the classification algorithm used to train most classifiers in both ensembles, and $A_2$ is the classification algorithm used to train a classifier in a hybrid ensemble. When $\alpha_i$ and $\delta$ are treated as constants, the lower bound of the probability that we can observe accuracy gain depends on how unstable $A_1$ is and how different $A_1$ and $A_2$ are. If $\beta_1$ is larger, meaning that $A_1$ is more unstable, and $\gamma$ is larger, meaning that $A_1$ and $A_2$ are more different, then the lower bound of the probability is larger; if $\beta_1$ is smaller, meaning that $A_1$ is more stable, and $\gamma$ is smaller, meaning that $A_1$ and $A_2$ are more similar, then the lower bound of the probability is smaller, while this does not necessarily mean that the probability of observing accuracy gain is smaller. As a result, if the algorithm used to train most classifiers in a hybrid ensemble is unstable and it is different from the other algorithm, then it is more probable that accuracy gain would be observed. Furthermore, the lower bound of the accuracy gain depends on how unstable $A_1$ is, how different $A_1$ and $A_2$ are, and how accurate the classifiers trained with $A_1$ and $A_2$ are. If the classifier trained with $A_2$ is more accurate, $\text{Acc}_2 - \text{Acc}_1$ is larger and the lower bound of accuracy gain is larger. As a result, it is beneficial to replace a classifier (or some classifiers) in a non-hybrid ensemble with a classifier (or some classifiers) trained with a different yet accurate classification algorithm. Nevertheless, it may not be beneficial to replace all, because doing so would lower diversity among classifiers.

**Proof.** $T_1$, $T_2$, and $T_3$ are three data sets generated with bootstrap sampling from a given data set. Classifiers $C_1$, $C_2$, and $C_3$ are trained by applying a classification algorithm $A_1$ on $T_1$, $T_2$, and $T_3$, respectively. That is, $C_1(x \mid A_1, T_1) = \hat{y}^{(A_1, T_1)}$, $C_2(x \mid A_1, T_2) = \hat{y}^{(A_1, T_2)}$, and $C_3(x \mid A_1, T_3) = \hat{y}^{(A_1, T_3)}$. The ensemble composed of $C_1$, $C_2$, and $C_3$ is a non-hybrid ensemble, because it is with only a classification algorithm. Classifier $C'_2$ is trained by applying another classification algorithm $A_2$ on $T_2$. That is, $C'_2(x \mid A_2, T_2) = \hat{y}^{(A_2, T_2)}$. The ensemble composed of $C_1$, $C'_2$, and $C_3$ is a hybrid ensemble, because it is with two different classification algorithms.

Initially, we represent the expected accuracy of the non-hybrid ensemble in (7), which is based on majority voting. In what follows, $y$ is the class label of a data sample

\[
E \left[ \mathbb{1} \left( \hat{y}^{(A_1, T_1)} = y \wedge \hat{y}^{(A_2, T_2)} = y \wedge \hat{y}^{(A_1, T_3)} = y \right) \right] + \mathbb{1} \left( \hat{y}^{(A_1, T_1)} = y \wedge \hat{y}^{(A_2, T_2)} = y \wedge \hat{y}^{(A_1, T_3)} \neq y \right) + \mathbb{1} \left( \hat{y}^{(A_1, T_1)} \neq y \wedge \hat{y}^{(A_2, T_2)} = y \wedge \hat{y}^{(A_1, T_3)} = y \right) \]

(7)

Similarly, we represent the expected accuracy of the hybrid ensemble in

\[
E \left[ \mathbb{1} \left( \hat{y}^{(A_1, T_1)} = y \wedge \hat{y}^{(A_2, T_2)} = y \wedge \hat{y}^{(A_1, T_3)} = y \right) \right] + \mathbb{1} \left( \hat{y}^{(A_1, T_1)} = y \wedge \hat{y}^{(A_2, T_2)} = y \wedge \hat{y}^{(A_1, T_3)} \neq y \right) + \mathbb{1} \left( \hat{y}^{(A_1, T_1)} \neq y \wedge \hat{y}^{(A_2, T_2)} = y \wedge \hat{y}^{(A_1, T_3)} = y \right) \]

(8)

The difference between the expected accuracy of the hybrid ensemble and that of the non-hybrid ensemble or the difference between (8) and (7) is denoted by $\Delta E[\text{Acc}]$. It is given in

\[
\Delta E[\text{Acc}] = E \left[ \mathbb{1} \left( \hat{y}^{(A_1, T_1)} = y \wedge \hat{y}^{(A_2, T_2)} = y \wedge \hat{y}^{(A_1, T_3)} = y \right) \right] - \mathbb{1} \left( \hat{y}^{(A_1, T_1)} = y \wedge \hat{y}^{(A_2, T_2)} = y \wedge \hat{y}^{(A_1, T_3)} \neq y \right) - \mathbb{1} \left( \hat{y}^{(A_1, T_1)} \neq y \wedge \hat{y}^{(A_2, T_2)} = y \wedge \hat{y}^{(A_1, T_3)} = y \right) - \mathbb{1} \left( \hat{y}^{(A_1, T_1)} \neq y \wedge \hat{y}^{(A_2, T_2)} \neq y \wedge \hat{y}^{(A_1, T_3)} = y \right) \]

(9)

Next, (9) is rewritten as (10) by rearranging its components

\[
\Delta E[\text{Acc}] = E \left[ \mathbb{1} \left( \hat{y}^{(A_1, T_1)} = y \right) \cdot \mathbb{1} \left( \hat{y}^{(A_2, T_2)} = y \right) \right] \times \left( \mathbb{1} \left( \hat{y}^{(A_1, T_1)} = y \right) - \mathbb{1} \left( \hat{y}^{(A_1, T_1)} \neq y \right) \right) + \mathbb{1} \left( \hat{y}^{(A_1, T_1)} = y \right) \cdot \mathbb{1} \left( \hat{y}^{(A_2, T_2)} \neq y \right)
\]
Next, (11) is obtained:

\[ P[\delta \cdot (\text{Acc}_2 - \text{Acc}_1)] \geq (1 - \beta_1 \cdot (1 - y)) \cdot \delta \cdot (\text{Acc}_2 - \text{Acc}_1) \]  

Finally, (15) is obtained through (14) and the proof is complete:

\[ \forall \alpha_1 \cdot [\delta \cdot (\text{Acc}_2 - \text{Acc}_1)] \geq (1 - (1 - \beta_1) \cdot (1 - y)) \cdot \delta \cdot (\text{Acc}_2 - \text{Acc}_1) \]

We treat \( \alpha_1 \) and \( \delta \) as constants, as in the processes shown in Algorithms 2 and 3. Given an ensemble of classifiers trained by using \( A_1 \). We replace some classifiers with those trained by using \( A_2 \), which generally provides a higher value of accuracy (so that \( \text{Acc}_2 \) is larger than \( \text{Acc}_1 \)), and the lower bound of accuracy gain will be positive, meaning that we would probably obtain a hybrid ensemble that could achieve better classification performance. If \( \beta_1 \) is more unstable, \( \beta_1 \) will be larger. If \( \beta_2 \) is more different from \( \beta_1 \), \( \beta_2 \) will be larger. The larger \( \beta_1 \) and the larger \( y \), the larger the lower bound of the probability that we would observe better accuracy.

3. Results and Discussion

3.1. Data Sets. In experiments, we use 20 data sets from UCI Machine Learning Repository [60], supported by School of Information and Computer Science, University of California, Irvine, and from Department of Statistics, Carnegie Mellon University. Using public data sets is to allow the reproduction of the experiments.

The data sets used in experiments are from various application domains. Their characteristics are summarized
Input: A data set \( D \), two classification algorithms \( A_1 \) and \( A_2 \), \( \delta \) as in (5), the number of trials \( t \)
Output: A possible value of \( \gamma \) as in (5)

Steps:
1. \( s \leftarrow 0 \)
2. For \( i = 1 \) to \( t \)
3. Use bootstrap sampling to sample \( D \) and generate \( T_1 \)
4. Train \( C_1 \) by applying \( A_1 \) on \( T_1 \)
5. Train \( C_2 \) by applying \( A_2 \) on \( T_1 \)
6. Sample \( D \) and generate \( T_2 \)
7. Use \( T_2 \) to test \( C_1 \) and \( C_2 \)
8. If the disagreement rate \( \geq \delta \) then
   \( s \leftarrow s + 1 \)
9. End If
10. End For
11. Return \( s/t \)

Algorithm 3: Process to estimate differentiability.

in Table 1, where the first column is the serial number, the second column is the name of a data set, the third column is the number of data samples, the fourth column is the number of data samples that belong to the minority class, the fifth column is the number of nominal attributes, and the numbers in parentheses present the numbers of attributes with missing values. The proportion of minority samples affects the classification performance of a classifier or an ensemble, and so does the proportion of attributes with missing values. For a data set that is usually used in regression analysis, we apply discretization on its target attribute and divide continuous values of the attribute properly into two intervals each of which corresponds to a class label; such a data set is with the suffix “binary.”

3.2. Settings. We consider C4.5 decision tree and naïve Bayes classification algorithms. The former is denoted by DT, and the latter is denoted by NB. We consider single classifiers trained by using DT or NB, non-hybrid ensembles trained with DT or NB, and hybrid ensembles trained with DT and NB.

We implement the process given in Algorithm 1 by using WEKA [61], and we compare the classification performances given by ensembles created by it to that given by Bagging. The ensembles created by using Bagging with DT or NB are non-hybrid ensembles with only DT or NB. We set the number of classifiers in an ensemble to 10 for all ensemble learning algorithms. We use \( 20 \times 10 \)-fold cross-validation for classification performance evaluation. That is, for each combination of an ensemble learning algorithm and a data set, we run 10-fold cross-validation 20 times randomly and independently. In addition, we investigate instability and differentiability.

3.3. Instability. We use the process described earlier to estimate the instability of an algorithm on a data set. We set \( \alpha \) to 0.05 (by statistical convention) and the number of trials to 100. We report the results in Table 2. The content in a cell indicates the value of \( \beta \) or the probability that the disagreement rate is larger than or equal to \( \alpha \) or 0.05. The disagreements are given by two classifiers trained by applying the algorithm corresponding to the column on two data sets sampled from the data set corresponding to the row. The probability is calculated with reference to 100 trials. The larger the value of \( \beta \), the more unstable the algorithm on the data set. DT is more unstable than NB on 13 out of these 20 data sets. This is consistent with the general thought that decision tree algorithm is unstable (or it is a classification algorithm with high variance) and therefore suitable for being used to
Table 2: Results for instability.

| Number | DT   | NB   |
|--------|------|------|
| (1)    | 0.16 | 0.01 |
| (2)    | 0.04 | 0.06 |
| (3)    | 0    | 0    |
| (4)    | 0.61 | 0.45 |
| (5)    | 0.72 | 0.25 |
| (6)    | 1    | 0.97 |
| (7)    | 0.61 | 0.23 |
| (8)    | 0.96 | 0.66 |
| (9)    | 0.9  | 0.37 |
| (10)   | 0.9  | 0.18 |
| (11)   | 0.7  | 0.33 |
| (12)   | 0.72 | 0.58 |
| (13)   | 1    | 0.24 |
| (14)   | 0.76 | 0.68 |
| (15)   | 0.05 | 0.49 |
| (16)   | 0    | 0.18 |
| (17)   | 1    | 0.85 |
| (18)   | 0    | 0.04 |
| (19)   | 0.19 | 0.9  |
| (20)   | 0    | 0    |

Table 3: Results for differentiability.

| Number | DT versus NB |
|--------|--------------|
| (1)    | 0.93         |
| (2)    | 1            |
| (3)    | 0.05         |
| (4)    | 1            |
| (5)    | 1            |
| (6)    | 1            |
| (7)    | 1            |
| (8)    | 1            |
| (9)    | 1            |
| (10)   | 1            |
| (11)   | 1            |
| (12)   | 0.98         |
| (13)   | 1            |
| (14)   | 0.97         |
| (15)   | 1            |
| (16)   | 1            |
| (17)   | 1            |
| (18)   | 1            |
| (19)   | 1            |
| (20)   | 0.95         |

create ensembles. The difference of values of instability is larger than or equal to 0.5 on 4 data sets: heart-c-binary, heart-h-binary, hprice-binary, and sonar. The results clearly show that instability results from not only the nature of a data set but also the nature of a classification algorithm.

3.4. Differentiability. We use the process described earlier to estimate the differentiability of two algorithms on a data set. We set $\delta$ to 0.05 (by statistical convention) and the number of trials to 100. We report the results in Table 3. The content in a cell indicates the value of $\gamma$ or the probability that the disagreement rate is larger than or equal to $\delta$ or 0.05. The disagreements are given by two classifiers trained by applying DT and NB separately on a data set sampled from the data set corresponding to the row. The probability is calculated with reference to 100 trials. The larger the value of $\gamma$, the more different the two algorithms on the data set. According to Table 3, DT and NB are different or behave differently on 19 out of these 20 data sets; they are not sufficiently different on the data set breast-w. Decision tree and naïve Bayes algorithms are fundamentally different: For example, the former makes no assumptions on the data set, while the latter assumes statistical independence between attributes; the former discretizes numeric attributes, while the latter can apply density estimation on numeric attributes; the former uses special treatment for missing values, while the latter handles missing values naturally.

3.5. Performance. We report the values of accuracy in Table 4. The results have shown support for the idea that we could possibly obtain better classification performance by using different classification algorithms to train classifiers in an ensemble. In the table, HE is for the hybrid ensemble, and the content of a cell indicates the mean and standard deviation calculated over 20 runs of 10-fold cross-validation for applying the algorithm corresponding to the column on the data set corresponding to the row. Generally speaking, compared to a single classifier, an ensemble would show stabilized classification performance, especially when the underlying classification algorithm is an unstable one. We can see this from the fact that the standard deviations given by ensembles are lower than those given by single classifiers in most cases. For example, on the first data set, biomed, the standard deviation given by DT is 0.014, that given by Bagging DT is 0.009, and that given by the hybrid ensemble DT + NB is 0.007.

In some cases, such as those where data distributions are skewed, accuracy is not a good measure for classification performance evaluation, while F1-measure is a more comprehensive measure. In Table 5, we report the values of F1-measure, which is the harmonic mean of precision and recall, for data samples that belong to the minority class. A higher value of F1-measure means better classification performance. F1-measure for minority is used to evaluate how well a classification algorithm performs on data samples that belong to the minority class, which are usually the targets in most real-world machine learning applications. A classifier or an ensemble can achieve a low error rate simply by classifying all samples to the majority class or simply by ignoring minority samples on highly unbalanced data sets. In the table, similarly, HE is for the hybrid ensemble, and the content of a cell indicates the mean and standard deviation.

To assess the significance of differences in classification performance achieved by two algorithms, we perform the
Wilcoxon test, a non-parametric test, whose use is recommended by Demšar [62]. We report results for the test for accuracy and F1-measure in Tables 6 and 7, respectively. In the tables, B-DT means Bagging DT, B-NB means Bagging NB, and similarly HE is for the hybrid ensemble. In the tables, the content in a cell indicates the number of data sets on which the algorithm in the corresponding row significantly wins (outperforms) and loses to (is outperformed by) the
algorithm in the corresponding column. For example, in Table 6, the first cell shows that decision tree algorithm wins in accuracy naïve Bayes algorithm on 10 data sets and it loses on 9 data sets, while the difference in accuracy is not significant on 1 data set; in Table 7, the first cell shows that decision tree algorithm wins in F1-measure naïve Bayes algorithm in 18 data sets and it does not lose on a data set, while the difference in F1-measure is not significant on 2 data sets.

We can see from Table 6 that, in terms of accuracy, the number of data sets on which HE DT + NB outperforms B-DT is slightly smaller than that on which HE DT + NB is outperformed by B-DT; the number of data sets on which HE DT + NB outperforms B-NB is much larger than that on which HE DT + NB is outperformed by B-NB. We can see from Table 7 that, in terms of F1-measure, the results are similar but HE DT + NB outperforms less. The results seem to be in opposition to creating and using hybrid ensembles. On the contrary, the results show that hybrid ensembles outperform single classifiers not deterministically but probabilistically, which is also shown by our analysis. Moreover, in this paper, we intend to explain why and when hybrid ensembles are better than non-hybrid ensembles.

We propose using different classification algorithms to train more diverse classifiers in order to create better ensembles. The basic idea is to use a combination of classifiers to naturally reduce variance and use a stronger algorithm to explicitly increase classification performance. The ensemble creation process that we propose is distinguishing, because it uses fundamentally different classification algorithms to create a hybrid ensemble. For example, it uses DT, which is often with high variance (related to high instability), and NB, which is often with low variance (related to low instability or high stability). Using such a combination of classification algorithms goes against the generally accepted sense that one should only use classification algorithms with high variance in an ensemble like Bagging. We evaluate the proposed process by using a varied collection of public data sets and two metrics. Experiment results reveal that the proposed process could achieve better performance when compared to Bagging.

3.6. Discussion. There are 8 data sets on which the hybrid ensemble DT + NB outperforms both Bagging DT and Bagging NB: biomed, breast-w, credit-g, diabetes, heart-c-binary, heart-statlog, hepatitis, and schizo. According to Table 4, naïve Bayes algorithm performs better in accuracy than decision tree algorithm does on these data sets. Therefore, we consider that \( A_1 \) is DT and \( A_2 \) is NB, which is newly introduced into the ensemble and used as another generator of diversity. For these data sets, we summarize lower bounds for accuracy gain and the probability that the gain is obtained in Table 8. In the table, the second column (\( Acc_1 \)) and the third column (\( Acc_2 \)) are from the second column and the third column in Table 4, respectively; the fourth column is from the second column in Table 2; the fifth column is from the second column in Table 3; the sixth and seventh columns are lower bounds for accuracy gain and the probability that the gain is obtained, respectively, and both are calculated by (6), or (15), given that \( \alpha_1 = 0.05 \) and \( \delta = 0.05 \); the eighth and ninth columns are from the fourth and sixth columns in Table 4, respectively; the tenth column is the accuracy difference between HE DT + NB and B-DT. According to Table 8, on all these data sets, the accuracy difference, or the actual gain given by the hybrid ensemble DT + NB against Bagging NB, is larger than the lower bound; all the probabilities are high, except the one for the data set number 3, breast-w, and this means that the hybrid ensemble DT + NB would highly probably outperform Bagging DT (and this indeed the case). Nevertheless, the lower bound is too loose (but it is still the only one presented so far), and finding a tighter bound would be part of the future work.

4. Conclusion

Ensemble learning is to train classifiers and then combine their classifications to make an overall classification. Many researchers use ensembles of classifiers created by using a single classification algorithm in various applications. These are non-hybrid ensembles, and why they work is becoming clearer. Some researchers propose the use of the mixture of two different types of classification algorithms in the creation of a hybrid ensemble. In this paper, we investigate why hybrid ensembles work, which is somewhat unclear. We present our theoretical analysis from the standpoint of diversity, which plays a significant role in ensemble learning and is one of the fundamental elements of the success of ensembles. This is the most distinguishing characteristic of this paper. We also report and discuss experiment results obtained from hybrid ensembles of classifiers created by decision tree and naïve Bayes classification algorithms, each of which is a top data mining algorithm and often used to create non-hybrid ensembles. These are two fundamentally different classification algorithms, and therefore it is interesting to see that hybrid ensembles created with them together can achieve similar or even better classification performance compared to non-hybrid ensembles created with them individually. In short, we contribute to a complement to the theoretical

### Table 6: Results for Wilcoxon test for accuracy (\( W/L \): row versus column).

|          | NB  | B-DT | B-NB | HE DT + NB |
|----------|-----|------|------|------------|
| DT       | 10/9| 0/16 | 10/9 | 8/11       |
| NB       | 7/12| 0/1  | 0/17 | 0/1        |
| B-DT     | 12/7| 9/8  | 1/17 |            |
| B-NB     | 1/17|      |      |            |

### Table 7: Results for Wilcoxon test for F1-measure (\( W/L \): row versus column).

|          | NB  | B-DT | B-NB | HE DT + NB |
|----------|-----|------|------|------------|
| DT       | 18/0| 0/18 | 9/9  | 7/11       |
| NB       | 8/11| 1/2  | 2/13 |            |
| B-DT     | 11/7| 10/7 |      |            |
| B-NB     | 2/12|      |      |            |
foundation of creating and using hybrid ensembles. The hybrid ensemble performs better not in a deterministic but a probabilistic manner. In a hybrid ensemble where two classification algorithms are used, if two are different and one is unstable while the other is more accurate, then there is a higher probability that we can have a higher value of accuracy gain; the gain is measured against an ensemble where only one of the two classification algorithms is used.

Competing Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

References

[1] R. Ranawana and V. Palade, “Multi-classifier systems: review and a roadmap for developers,” International Journal of Hybrid Intelligent Systems, vol. 3, no. 1, pp. 35–61, 2006.
[2] R. Polikar, “Ensemble based systems in decision making,” IEEE Circuits and Systems Magazine, vol. 6, no. 3, pp. 21–45, 2006.
[3] R. Polikar, “Bootstrap-inspired techniques in computation intelligence,” IEEE Signal Processing Magazine, vol. 24, no. 4, pp. 59–72, 2007.
[4] G. Brown, “Ensemble learning,” in Encyclopedia of Machine Learning, pp. 312–320, Springer, New York, NY, USA, 2010.
[5] C. Zhang and Y. Ma, Eds., Ensemble machine learning, Springer, NY, USA, 2012.
[6] Z.-H. Zhou, Ensemble Methods: Foundations and Algorithms, Chapman & Hall/CRC, 2012.
[7] M. Woźniak, M. Graña, and E. Corchado, “A survey of multiple classifier systems as hybrid systems,” Information Fusion, vol. 16, no. 1, pp. 3–17, 2014.
[8] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, “Do we need hundreds of classifiers to solve real world classification problems?” The Journal of Machine Learning Research, vol. 15, no. 1, pp. 3133–3181, 2014.
[9] G. Giacinto and F. Roli, “Design of effective neural network ensembles for image classification purposes,” Image and Vision Computing, vol. 19, no. 9–10, pp. 699–707, 2001.
[10] K.-S. Goh, E. Chang, and K.-T. Cheng, “SVM binary classifier ensembles for image classification,” in Proceedings of the 10th International Conference on Information and Knowledge Management (ACM CIKM ’01), pp. 395–402, November 2001.
[11] M. Pal, “Ensemble of support vector machines for land cover classification,” International Journal of Remote Sensing, vol. 29, no. 10, pp. 3043–3049, 2008.
[12] A. Merentitis, C. Debes, and R. Heremans, “Ensemble learning in hyperspectral image classification: toward selecting a favorable bias-variance tradeoff,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 7, no. 4, pp. 1089–1102, 2014.
[13] A. Samat, P. Du, S. Liu, J. Li, and L. Cheng, “E²LMs: ensemble extreme learning machines for hyperspectral image classification,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 7, no. 4, pp. 1060–1069, 2014.
[14] M. Han and B. Liu, “Ensemble of extreme learning machine for remote sensing image classification,” Neurocomputing, vol. 149, pp. 63–70, 2015.
[15] R. Cappelli, D. Maio, and D. Maltoni, “A multi-classifier approach to fingerprint classification,” Pattern Analysis and Applications, vol. 5, no. 2, pp. 136–144, 2002.
[16] I. Maqsood, M. R. Khan, and A. Abraham, “An ensemble of neural networks for weather forecasting,” Neural Computing and Applications, vol. 13, no. 2, pp. 112–122, 2004.
[17] Y.-S. Dong and K.-S. Han, “A comparison of several ensemble methods for text categorization,” in Proceedings of the IEEE International Conference on Services Computing (SCC ’04), pp. 419–422, September 2004.
[18] T. Rohlfing and C. R. Maurer Jr., “Multi-classifier framework for atlas-based image segmentation,” Pattern Recognition Letters, vol. 26, no. 13, pp. 2070–2079, 2005.
[19] S. Avidan, “Ensemble tracking,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 2, pp. 261–271, 2007.
[20] H.-B. Shen and K.-C. Chou, “Ensemble classifier for protein fold pattern recognition,” Bioinformatics, vol. 22, no. 14, pp. 1717–1722, 2006.
[21] S. B. Cho and H.-H. Won, “Cancer classification using ensemble of neural networks with multiple significant gene subsets,” Applied Intelligence, vol. 26, no. 3, pp. 243–250, 2007.
[22] D. Gray and H. Tao, “View point invariant pedestrian recognition with an ensemble of localized features,” in Proceedings of the 10th European Conference on Computer Vision: Part I, pp. 262–275, Marseille, France, 2008.
[23] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, “Pedestrian detection with spatially pooled features and structured ensemble learning,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 6, pp. 1243–1257, 2016.
[24] H. I. Aljamaan and M. O. Elish, “An empirical study of bagging and boosting ensembles for identifying faulty classes in object-oriented software,” in Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM ’09), pp. 187–194, IEEE, Nashville, Tenn, USA, April 2009.

Table 8: Lower bounds for accuracy gain and the probability.

| Number | Acc1 | Acc2 | β1 | y | Acc. gain | Prob. | B-DT | HE | Acc. diff |
|--------|------|------|----|---|----------|-------|------|----|----------|
| (1)    | 0.891| 0.894| 0.16| 0.93| −7.5 × 10⁻⁶| 0.9412| 0.908| 0.909| 0.001    |
| (3)    | 0.948| 0.96 | 0   | 0.05| −0.00003 | 0.05  | 0.958| 0.967| 0.009    |
| (6)    | 0.714| 0.751| 1   | 1   | −9.25 × 10⁻⁵| 1     | 0.738| 0.755| 0.017    |
| (8)    | 0.745| 0.755| 0.96| 1   | −0.000025| 1     | 0.759| 0.767| 0.008    |
| (9)    | 0.775| 0.833| 0.9 | 1   | −0.000145| 1     | 0.787| 0.835| 0.048    |
| (11)   | 0.784| 0.839| 0.7 | 1   | −0.0001375| 1     | 0.801| 0.847| 0.046    |
| (12)   | 0.784| 0.839| 0.72| 0.98| −0.0001375| 0.9944| 0.805| 0.85 | 0.045    |
| (17)   | 0.56 | 0.575| 1   | 1   | −3.75 × 10⁻⁵| 1     | 0.595| 0.6  | 0.005    |
[59] K.-W. Hsu and J. Srivastava, “Improving bagging performance through multi-algorithm ensembles,” *Frontiers in Computer Science*, vol. 6, no. 5, pp. 498–512, 2012.

[60] M. Lichman, *UCI Machine Learning Repository*, School of Information and Computer Science, University of California, Irvine, Calif, USA, 2013, http://archive.ics.uci.edu/ml.

[61] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, ”The WEKA data mining software: an update,” *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[62] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.