In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000),*
pp. 735-742, Morgan Kaufmann, San Francisco, 2000.

A Normative Examination of Ensemble Learning Algorithms

David M. Pennock DPENNOCK@RESEARCH.NJ.NEC.COM
NEC Research Institute, 4 Independence Way, Princeton, NJ 08540 USA

Pedrito Maynard-Reid II PEDMAYN@CS.STANFORD.EDU
Computer Science Department, Stanford University, Stanford, CA 94305 USA

C. Lee Giles GILES@RESEARCH.NJ.NEC.COM
NEC Research Institute, 4 Independence Way, Princeton, NJ 08540 USA

Eric Horvitz HORVITZ@MICROSOFT.COM
Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399 USA

Abstract

Ensemble learning algorithms combine the results of several classifiers to yield an aggregate classification. We present a normative evaluation of combination methods, applying and extending existing axiomatizations from social choice theory and statistics. For the case of multiple classes, we show that several seemingly innocuous and desirable properties are mutually satisfied only by a *dictatorship*. A weaker set of properties admit only the weighted average combination rule. For the case of binary classification, we give axiomatic justifications for majority vote and for weighted majority. We also show that, even when all component algorithms report that an attribute is probabilistically independent of the classification, common ensemble algorithms often destroy this independence information. We exemplify these theoretical results with experiments on stock market data, demonstrating how ensembles of classifiers can exhibit canonical voting paradoxes.

1. Introduction

A recent trend in machine learning is to aggregate the outputs of several learning algorithms together to produce a composite classification (Dietterich, 1997). Under favorable conditions, ensemble classifiers provably outperform their constituent algorithms, an advantage born out by much empirical validation. Yet there does not seem to be a single, obvious way to combine classifiers—many different methods have been proposed and tested, with none emerging as the clear winner. Most evaluation metrics center on generalization accuracy, either deriving theoretical bounds (Schapire, 1990; Freund & Schapire, 1999) or (more commonly) comparing experimental results (Bauer & Kohavi, 1999; Breiman, 1996; Dietterich, in press; Freund & Schapire, 1996).

We take instead a normative approach, informed by results from social choice theory and statistical belief aggregation. First, we identify several properties that an ensemble algorithm might ideally possess, and then characterize the implied form of the combination function. Section 4 examines the case of more than two classes. We show that, under a set of seemingly mild and reasonable conditions, *no* true combination method is possible. The aggregate classification is always identical to that of only one of the component algorithms. The analysis mirrors Arrow's celebrated Impossibility Theorem, which shows that the only voting mechanism that obeys a similar set of properties is a dictatorship (Arrow, 1963). Under slightly weaker demands, we show that the only possible form for the combination function is a weighted average of the constituent classifications.

Section 5 considers the special case of binary classification. Based on May's (1952) seminal work, we present a set of axioms that necessitate the use of simple majority vote to combine classifiers. We then extend this result, deriving an axiomatic justification for the *weighted* majority vote. Majority and weighted majority are two of the most common methods used for classifier combination (Dietterich, 1997). One contribution of this paper is to provide formal justifications for them.

Section 6 explores the independence preservation properties of common ensemble learning algorithms. Suppose that, with some attribute values missing, all of the constituent algorithms judge one attribute to be statistically independent of the classification. We demonstrate that this independence is generally lost after combination, rendering the aggregate classification statistically *dependent* on the attribute in question.

Section 7 presents empirical evidence of violations of the various axioms. We show that an ensemble of neural networks—trained to predict stock market data—can generate counterintuitive results, reminiscent of so-called *voting paradoxes* in the social choice literature. Section 8 summarizes and discusses future work.

2. Ensemble Learning

We present a very brief overview of ensemble learning; see (Dietterich, 1997) for an excellent survey. Representative algorithms include bagging (Breiman, 1996), boosting (e.g., ADABOOST (Freund & Schapire, 1999)), and a method based on Error-Correcting Output Codes (ECOC) (Dietterich & Bakiri, 1995). Ensemble algorithms generally proceed in two phases: (1) generate and train a set of weak learners, and (2) aggregate their classifications.

The first step is to construct component learners of sufficient diversity (Hansen & Salamon, 1990). One common technique is to subsample the training examples, either randomly with replacement (Breiman, 1996), by leaving out random subsets (as in cross-validation), or by an induced distribution meant to magnify the effect of difficult training examples (Freund & Schapire, 1999). Another technique bases each learner's predictions on different input features (Tumer & Ghosh, 1996). The method of Error-Correcting Output Codes (ECOC) generates classifiers by having each learn whether an example falls within a randomly chosen subset of the classes. Another approach injects randomness into the training algorithms themselves. These four techniques apply to arbitrary classifier algorithms—there are also many algorithm-specific techniques. And, of course, it is possible to create an ensemble by mixing and matching different techniques for different classifiers.

After generating and training a set of weak learners, the ensemble algorithm combines the individual learners' predictions into a composite prediction. The choice of combination method is the focus of this paper. Common methods can be categorized loosely into two categories: those that combine *votes*, and those that can combine confidence scores. The former type includes plurality vote[1] and weighted plurality; the latter includes stacking, serial combination, weighted average, and weighted geometric average.

Bagging and ECOC are examples of algorithms that use plurality vote. The ensemble's chosen class is simply that which is predicted most often by the individual learners. Weighted plurality is a generalization of plurality vote, where each algorithm's vote is discounted (or magnified) by a multiplicative weight; classes are then ranked according to the sum of the weighted votes they receive. Weights can be chosen to correspond with the observed accuracy of the individual classifiers, using Bayesian techniques, or using *gating networks* (Jordan & Jacobs, 1994), among other methods. The ADABOOST algorithm computes weights in an attempt to minimize the error of the final classification.

Stacking turns the problem of finding a good combination function into a learning problem itself (Breiman, 1996; Lee & Srihari, 1995; Wolpert, 1992): The constituent algorithms' outputs are fed to a meta learner's inputs; the meta learner's output is taken as the ensemble classification. *Serial combination* uses one learner's top k choices to reduce the space of candidate classes, passing the simplified problem onto the next learner, etc. (Madhvanath & Govindaraju, 1995). *Weighted algebraic (or geometric) average* computes the aggregate confidence in each class as a weighted algebraic (or geometric) average of the individual confidences in that class (Jacobs, 1995; Tax et al., 1997). Some variants of boosting employ weighted average combination (Drucker et al., 1993).

3. Notation

Let $\mathbf{A} = \langle A_1, A_2, \ldots, A_L\rangle$ denote a vector of L attribute variables with domain $\mathbf{D} = D_1 \times \cdots \times D_L$. Denote a corresponding vector of values (i.e., instantiated variables) as $\mathbf{a} = \langle a_1, a_2, \ldots, a_L\rangle \in \mathbf{D}$. Each vector $\mathbf{a}$ is categorized into one of M *classes*, $C_1, C_2, \ldots, C_M$. There are N *classifiers*, or learners, which attempt to learn a functional mapping from instantiated attributes to classes. Different types of classifiers return different amounts of information—some return a single vote for one predicted class, others return a ranking of the classes, and still others return confidence scores for all classes.[2] Our contention is that confidence information is usually available, whether explicitly (e.g., from neural net activation values, or Bayesian net or decision tree likelihoods) or implicitly from observed performance on the training data. Thus we denote learner i's *classification* as an assignment $\langle S_{i1}, \ldots, S_{iM}\rangle$ of confidence scores to the classes, where $S_{ij} \in \Re$. Each classifier is a function $f_i : \mathbf{D} \rightarrow \Re^M$. When confidence magnitude information is truly unavailable, we adopt Lee and Srihari's (1995) conventions for encoding classifications: A single vote for class C_j is represented as a classification vector with a 1 in the jth position and zeros elsewhere; a

[1] This is the familiar "one person, one vote" procedure where the candidate receiving the most votes wins. We reserve *majority vote* to refer to the special case of two candidates.

[2] These three output conditions correspond to Lee and Srihari's (1995) definitions of Type I, Type II, and Type III classifiers, respectively.

rank list of the classes is represented as a vector with a 1 in the top class position, $1 - 1/M$ in the second place position, $1 - 2/M$ in the third place position, etc. Note that, technically, these two encodings introduce unfounded comparative information. For example, a vote for C_j conveys only that all other classes are less preferred than C_j, but are otherwise incomparable among themselves. Variants of the limitative theorems in this paper are also possible using more faithful representations of votes and rankings.

An *ensemble combination function* g accepts an N-tuple of classifications and returns a composite classification; that is, $g : \mathbf{K} \to \Re^M$, where $\mathbf{K} \subseteq (\Re^M)^N$. Thus, assuming $\mathbf{K} = (\Re^M)^N$, the aggregate classification of arbitrary classifiers $f_1, \ldots, f_N$ on an input $\mathbf{a}$ is $g(f_1(\mathbf{a}), \ldots, f_N(\mathbf{a}))$.

For a given input vector $\mathbf{a} \in \mathbf{D}$, we find it convenient to define $\mathbf{S}$ as the $N \times M$ matrix of all learners' confidence scores for all classes. That is, S_{ij} is learner i's confidence that $\mathbf{a}$ is in class j. Let $\mathbf{r}_i$ be an N-dimensional row vector with a 1 in the ith position and zeros elsewhere; similarly, let $\mathbf{c}_j$ be an M-dimensional column vector with a 1 in the jth position and zeros elsewhere. Then $\mathbf{r}_i\mathbf{S}$ is the ith row of $\mathbf{S}$, and $\mathbf{S}\mathbf{c}_j$ is the jth column of $\mathbf{S}$. In other words, $\mathbf{r}_i\mathbf{S} = f_i(\mathbf{a})$ is learner i's classification, and $\mathbf{S}\mathbf{c}_j$ is the vector of all confidence scores for class j. Note that $\mathbf{r}_i\mathbf{S}\mathbf{c}_j = S_{ij}$. We denote the ensemble classification by $\mathbf{S}_0 = \langle S_{01}, S_{02}, \ldots, S_{0M} \rangle = g(\mathbf{S})$. We write $\mathbf{v} > \mathbf{w}$ to indicate that every component of $\mathbf{v}$ is strictly greater than the corresponding component of $\mathbf{w}$.

4. Multiple Classes

In this section, we propose a normative basis for ensemble learning when $M \geq 3$. Our treatment is similar in spirit to Pennock, Horvitz and Giles's (in press) analysis of the axiomatic foundations of collaborative filtering.

4.1 An Impossibility Theorem

We present five properties adopted from social choice theory, argue their merits in the context of ensemble learning, and describe which existing algorithms exhibit which properties. Each property places a constraint on the allowable form of g.

Property 1 (UNIV) Universal domain. $\mathbf{K} = (\Re^M)^N$.

UNIV requires that g be defined for any combination of classification vectors. Since an arbitrary classifier may return an arbitrary classification, it seems only reasonable that g should return some result in all circumstances. All existing ensemble combination methods, to our knowledge, are defined for all possible classifier output patterns.

Property 2 (ND) Non-dictatorship. *There is no dictator i such that, for all classification matrices $\mathbf{S}$ and all classes j and k, $S_{ij} > S_{ik} \Rightarrow S_{0j} > S_{0k}$.*

In words, g is not permitted to completely ignore all but one of the classifiers, irrespective of $\mathbf{S}$. We consider the desirability of this axiom to be self-evident, since the whole point of ensemble learning is to improve upon the performance of the individual classifiers.

Property 3 (WP) Weak Pareto principle. *For all classes j and k, $\mathbf{S}\mathbf{c}_j > \mathbf{S}\mathbf{c}_k \Rightarrow S_{0j} > S_{0k}$.*

WP captures the natural ideal that, if *all* classifiers are strictly more confident about one class than another, then this relationship should be reflected in the ensemble classification. Essentially all voting schemes (e.g., plurality, pairwise majority, Borda count) satisfy WP. Weighted plurality and weighted averaging methods obey WP when all weights are nonnegative (and at least one is positive). If a particular classifier's predictions are bad enough, some combination functions (e.g., weighted average with negative weights, or stacking) may establish a negative dependence between that classifier's opinion and the ensemble result, and thus violate WP. However, researchers typically strive to generate ensembles of algorithms that are as accurate as possible for a given amount of diversity (Dietterich, 1997; Dietterich, in press).

Property 4 (IIA) Independence of irrelevant alternatives. *Consider two classification matrices $\mathbf{S}, \mathbf{S}'$.*

$$\begin{array}{ll} \textit{If} & \mathbf{S}\mathbf{c}_j = \mathbf{S}'\mathbf{c}_j \ \textit{ and } \ \mathbf{S}\mathbf{c}_k = \mathbf{S}'\mathbf{c}_k, \\ \textit{then} & S'_{0j} > S'_{0k} \Leftrightarrow S_{0j} > S_{0k}. \end{array}$$

Under IIA, the final relative ranking between two classes cannot depend on the confidence scores for any other classes. For example, suppose that, in classifying a fruit as either an apple, a banana, or a pear, the ensemble concludes that "apple" is most likely. Now imagine that we learn one piece of categorical knowledge (and nothing else): the fruit is *not* a pear. Every classifier diminishes its confidence in "pear", but leaves its relative confidences between "apple" and "banana" untouched. Intuitively, the ensemble should not suddenly conclude that the fruit is a banana; indeed, admitting such a reversal is contrary to most formal reasoning procedures, including Bayesian reasoning. Seemingly unfounded reversals like this are precisely what IIA guards against. Weighted averaging methods do satisfy IIA, although plurality vote, and most other voting techniques, can violate it. In Section 7, we illustrate the paradoxical results than can occur when IIA is not met.

Property 5 (SI) Scale invariance. *Consider two classification matrices $\mathbf{S}, \mathbf{S}'$. If $\mathbf{r}_i\mathbf{S}' = \alpha_i\mathbf{r}_i\mathbf{S} + \beta_i$ for all i and for any positive constants α_i and any constants β_i, then $S'_{0j} > S'_{0k} \Leftrightarrow S_{0j} > S_{0k}$ for all classes j and k.*

Different classifiers (especially those based on different learning algorithms) may report confidences using different scales—one, say, ranging from 0 to 1; another from

-100 to 100. Even if they share a common range, one classifier may tend to report confidence scores in the high end of the scale, while another tends to use the low end. SI reflects the intuition that all classifiers' scores should be normalized to a common scale before combining them. One natural normalization is:

$$\mathbf{r}_i\mathbf{S}' \leftarrow \frac{\mathbf{r}_i\mathbf{S} - \min(\mathbf{r}_i\mathbf{S})}{\max(\mathbf{r}_i\mathbf{S}) - \min(\mathbf{r}_i\mathbf{S})}. \tag{1}$$

This transforms all confidence scores to the $[0, 1]$ range, filtering out any dependence on multiplicative (α_i) or additive (β_i) scale factors.[3] Lee and Srihari justify a similar normalization simply because "each output [classification] vector is defined over a different space" (1995, p.42). Ensemble combination schemes based on votes or rankings are by definition invariant to scale; weighted averaging methods, on the other hand, are not.

Different researchers favor differing subsets of these five properties, at least implicitly via their choice of combination methods. Roberts (1980) proves that *no* combination algorithm whatsoever can "have it all".

Proposition 1 (Impossibility) *If $M > 2$, no function g simultaneously satisfies UNIV, ND, WP, IIA, and SI.*

Proof: Follows from Sen's (1986) or Roberts's (1980) extensions of Arrow's (1963) original theorem. ■

4.2 Weighted Average Combination

We might weaken SI, allowing the final classification to depend on the magnitudes of confidence differences, but not on additive scale shifts.

Property 6 (TI) Translation invariance. *Consider two classification matrices $\mathbf{S}, \mathbf{S}'$. If $\mathbf{r}_i\mathbf{S}' = \alpha\mathbf{r}_i\mathbf{S} + \beta_i$ for all i and for any (single) positive constant α and any constants β_i, then $S'_{0j} > S'_{0k} \Leftrightarrow S_{0j} > S_{0k}$ for all classes j and k.*

TI can be enforced by an additive normalization, or aligning all classifiers' scores with a common reference point (e.g., $\mathbf{r}_i\mathbf{S}' \leftarrow \mathbf{r}_i\mathbf{S} - \min(\mathbf{r}_i\mathbf{S})$).

This weakening is sufficient to allow for a non-dictatorial combination function g. Moreover, the only such g computes the ensemble confidence in each class as a weighted average of the component learners' confidences in that class.

Proposition 2 (Weighted average) *If $M > 2$, then the only function g satisfying UNIV, WP, IIA, and TI is such that $\mathbf{wSc}_j > \mathbf{wSc}_k \Rightarrow S_{0j} > S_{0k}$, where $\mathbf{w} = \langle w_1, w_2, \ldots, w_N\rangle$ is a row vector of N nonnegative weights, at least one of which is positive. If g is also continuous, then $\mathbf{wSc}_j > \mathbf{wSc}_k \Leftrightarrow S_{0j} > S_{0k}$.*

Proof: Follows from Roberts's (1980) Theorem 2. ■

Certainly there may exist classification domains where some of these properties do not seem appropriate or justified. However, we believe that, because the properties are very natural, understanding the limitations that they place on the space of ensemble learning algorithms helps to clarify what potential algorithms can and cannot do.

5. Binary Classification

Now consider the subset of learning problems where $M = |C| = 2$. In this case, the impossibility outlined in Proposition 1 disappears; the five properties UNIV, WP, IIA, SI, and ND are in fact perfectly compatible. For example, all five are satisfied by the standard *majority vote*:

$$\|S_{01} - S_{02}\| \equiv \left\| \sum_{i=1}^{N} \|S_{i1} - S_{i2}\| \right\| \tag{2}$$

where

$$\|x\| = \begin{cases} 1 & : \text{ if } x > 0 \\ 0 & : \text{ if } x = 0 \\ -1 & : \text{ if } x < 0 \end{cases}.$$

Note that the properties are necessary but not sufficient for characterizing majority vote. Proposition 3 below provides one sufficient characterization.

5.1 Majority Vote

The use of majority vote for ensemble learning is typically motivated by its simplicity, its observed effectiveness, and its perceived fairness when the constituent algorithms are essentially "created equal" (Dietterich, 1997). For example, the component algorithms employed for bagging, ECOC, and randomization are generally *a priori* indistinguishable, and (2) is typically used to combine classifications in these cases.

May (1952) provides an *axiomatic* justification for majority vote. His treatment is directly applicable when the constituent algorithms return only votes (equivalent to rankings since $M = 2$), rather than arbitrary confidence scores. We now generalize his axioms and his characterization theorem to apply to confidence scores.

Property 7 (NTRL) Neutrality.

$$\begin{array}{rll} \textit{If} & g\left(\langle S_{11}, S_{12}\rangle, \ldots, \langle S_{N1}, S_{N2}\rangle\right) & = \langle S_{01}, S_{02}\rangle \\ \textit{then} & g\left(\langle S_{12}, S_{11}\rangle, \ldots, \langle S_{N2}, S_{N1}\rangle\right) & = \langle S_{02}, S_{01}\rangle. \end{array}$$

Under NTRL, the effect of every algorithm reversing its vote is simply to reverse the aggregate vote. NTRL establishes a symmetry between the two class names, C_1 and C_2, ruling out any *a priori* bias for one class name over the other. Indeed, the subscripts 1 and 2 are assigned to the

[3] If $\max(\mathbf{r}_i\mathbf{S}) = \min(\mathbf{r}_i\mathbf{S})$ then set $\mathbf{r}_i\mathbf{S}'$ to $\mathbf{0}$.

two classes arbitrarily; NTRL simply ensures that the final result does not depend on how the two classes are indexed. NTRL is a strictly stronger constraint than IIA.

Property 8 (SYM) Symmetry.

$$\begin{aligned} & g\left(\langle S_{11}, S_{12}\rangle, \ldots, \langle S_{N1}, S_{N2}\rangle\right) \\ & \quad = \quad g\left(\langle S_{i_1 1}, S_{i_1 2}\rangle, \ldots, \langle S_{i_N 1}, S_{i_N 2}\rangle\right) \end{aligned}$$

where $\{i_1, i_2, \ldots, i_N\}$ *is any permutation of* $\{1, 2, \ldots, N\}$.

SYM is stronger than ND and is sometimes referred to as *anonymity*. Whereas NTRL implies an invariance under class name reversal, SYM enforces an invariance under any permutation of algorithm names, or subscripts. It simply insists that our numbering scheme has no effect on the output of the combination rule. Note that SYM *does not*, by itself, rule out a posterior bias based on the classifiers' reported confidence scores.

Property 9 (POSR) Positive responsiveness. *Consider two classification matrices* $\mathbf{S}, \mathbf{S}'$. *If* $||S_{01} - S_{02}|| \in \{0, 1\}$, *and* $\mathbf{r}_i\mathbf{S}' = \mathbf{r}_i\mathbf{S}$ *for all* $i \neq h$, *and* $\mathbf{r}_h\mathbf{S}'$ *is such that either*

1. $S'_{h1} > S_{h1}$ *and* $S'_{h2} = S_{h2}$, *or*
2. $S'_{h1} = S_{h1}$ *and* $S'_{h2} < S_{h2}$,

then $||S'_{01} - S'_{02}|| = 1$.

If the current aggregate vote is tied ($||S_{01} - S_{02}|| = 0$), then, under POSR, any change by any algorithm i in a positive direction for C_1 (i.e., S_{h1} increases or S_{h2} decreases) breaks this deadlock, yielding $S_{01} > S_{02}$. Moreover, any change of one of the constituent votes that strictly favors C_1 cannot swing the ensemble vote in the opposite direction, from C_1 to undecided or to C_2. Combined with NTRL, POSR is a stronger version of WP, but is still quite reasonable. Note that, because there are only two classes, if any learner's votes are observed to be negatively correlated with the correct classification (and, for example, a weighted average method assigns a negative weight), then its votes can simply be reversed, rendering POSR (and a nonnegative weight) appropriate again.

Proposition 3 (Majority vote) *An aggregation function* g *is the majority vote (2) if and only if it satisfies UNIV, SI, NTRL, SYM, and POSR.*

Proof: Choose scaling parameters as in Equation 1: $\alpha_i = (|S_{i1} - S_{i2}|)^{-1}$ (or if $S_{i1} = S_{i2}$, set $\alpha_i = 1$) and $\beta_i = -\alpha_i \min(S_{i1}, S_{i2})$. Let $\mathbf{r}_i\mathbf{S}' = \alpha_i\mathbf{r}_i\mathbf{S} + \beta_i$ for all i. Then

$$\langle S'_{i1}, S'_{i2}\rangle = \begin{cases} \langle 1, 0\rangle & : \text{ if } S_{i1} > S_{i2} \\ \langle 0, 0\rangle & : \text{ if } S_{i1} = S_{i2} \\ \langle 0, 1\rangle & : \text{ if } S_{i1} < S_{i2} \end{cases}.$$

That is, with only two classes, and two degrees of freedom in choosing the scaling constants, SI effectively restricts the domain $\mathbf{K}$ of g to votes. May (1952) proves that NTRL, SYM, and POSR are necessary and sufficient conditions for majority vote when inputs are votes. We refer the reader to May's article for the remainder of the proof. ■

Notice that, when the component algorithms return only votes, and no other information is available, SI is a vacuous requirement; in this setting, Proposition 3 becomes a very compelling normative argument for the use of majority vote for classifier combination.

5.2 Weighted Majority Vote

When the component algorithms do return meaningful confidence scores, SI may seem overly severe, as it essentially strips away magnitude information. Confidence scores may reflect many sources of information—for example, the activation levels of a neural network's output nodes, the posterior probabilities of a Bayesian network's output variables, or an algorithm's observed performance on the training data (as is used in Boosting). Regardless of its origin we interpret $S_{i1} > S_{i2}$ as a prediction in favor of class one, $S_{i2} > S_{i1}$ as a prediction in favor of class two, and the magnitude of the difference in confidence scores $|S_{i2} - S_{i1}|$ as the *weight* of algorithm i's conviction.

Then we define the *weighted majority vote* as

$$\begin{aligned} ||S_{01} - S_{02}|| & \equiv \left\| \sum_{i=1}^{N} |S_{i1} - S_{i2}| \cdot ||S_{i1} - S_{i2}|| \right\| \\ & = \left\| \sum_{i=1}^{N} S_{i1} - S_{i2} \right\|. \end{aligned} \quad (3)$$

Property 10 (SSYM) Separable symmetry.

$$\begin{aligned} & g\left(\langle S_{11}, S_{12}\rangle, \ldots, \langle S_{N1}, S_{N2}\rangle\right) \\ & \quad = \quad g\left(\langle S_{i_1 1}, S_{j_1 2}\rangle, \ldots, \langle S_{i_N 1}, S_{j_N 2}\rangle\right) \end{aligned}$$

where $\{i_1, i_2, \ldots, i_N\}$ *and* $\{j_1, j_2, \ldots, j_N\}$ *are any two permutations of* $\{1, 2, \ldots, N\}$.

SSYM is a stronger constraint than SYM. Under SSYM, the ensemble classification depends on the set of confidence scores for class one and the set of confidence scores for class two, but not on the identity of the algorithms that return those scores.

Proposition 4 (Weighted majority vote) *The only aggregation function* g *that satisfies UNIV, TI, NTRL, SSYM, and POSR is the weighted majority vote (3).*

Proof: Under UNIV and NTRL, $\mathbf{S} = \mathbf{0}$ implies that $S_{01} = S_{02}$. Thus, under POSR, if $S_{N1} > S_{N2}$ and

$S_{i1} = S_{i2} = 0$ for all $i \neq N$, then $S_{01} > S_{02}$. Similarly, because of NTRL, if $S_{N2} > S_{N1}$ and $S_{i1} = S_{i2} = 0$ for all $i \neq N$, then $S_{02} > S_{01}$. Given an arbitrary classification matrix $\mathbf{S}$, we can make the following invariance transformations. We invoke TI and SSYM alternately and repeatedly as follows:

$$\begin{aligned}
&g(\langle\langle S_{11}, S_{12}\rangle, \langle S_{21}, S_{22}\rangle, \langle S_{31}, S_{32}\rangle, \ldots\rangle) = \\
&g(\langle\langle S_{11} - S_{12}, 0\rangle, \langle 0, S_{22} - S_{21}\rangle, \langle S_{31}, S_{32}\rangle, \ldots\rangle) = \\
&g(\langle\langle 0, 0\rangle, \langle S_{11} - S_{12}, S_{22} - S_{21}\rangle, \langle S_{31}, S_{32}\rangle, \ldots\rangle) = \\
&g(\langle\langle 0, 0\rangle, \langle S_{11} + S_{21} - S_{12} - S_{22}, 0\rangle, \\
&\qquad\qquad\qquad\qquad \langle 0, S_{32} - S_{31}\rangle, \ldots\rangle) = \\
&\cdots = \\
&g\left(\left\langle \langle 0,0\rangle, \langle 0,0\rangle, \langle 0,0\rangle, \ldots, \left\langle \sum_{i=1}^{N} S_{i1} - S_{i2}, 0 \right\rangle\right\rangle\right)
\end{aligned}$$

Thus if $\sum_i S_{i1} - S_{i2}$ is greater than (less than, equal to) zero, then $S_{01} - S_{02}$ is greater than (less than, equal to) zero, precisely the weighted majority vote (3). ■

6. Independence Preservation

Consider the learners' predictions when asked to evaluate an example $\mathbf{a}^*$ with some missing values. Without loss of generality, let $A_1, A_2, \ldots, A_m$ be the attribute variables with missing values, and let $A_{m+1}, \ldots, A_L$ be the variables with known values. Let $\mathbf{a}^*_{m+} = \langle a^*_{m+1}, \ldots, a^*_L\rangle$ denote the vector of known values. If we define a prior joint probability distribution $\Pr(\mathbf{a})$ over all possible combinations of attribute values, then we can compute each learner's induced posterior distribution over classifications given the known values $\mathbf{a}^*_{m+}$:

$$\Pr_i(\mathbf{r}_i\mathbf{S}|\mathbf{a}^*_{m+}) = \sum_{\substack{\mathbf{x}\in\{D_1\times\cdots\times D_m\}: \\ f_i(\mathbf{x},\mathbf{a}^*_{m+})=\mathbf{r}_i\mathbf{S}}} \Pr(\mathbf{x}|\mathbf{a}^*_{m+}).$$

Similarly, we can compute the ensemble's posterior distribution over classifications:

$$\Pr_0(\mathbf{S}_0|\mathbf{a}^*_{m+}) = \sum_{\substack{\mathbf{x}\in\{D_1\times\cdots\times D_m\}: \\ g(f_1(\mathbf{x},\mathbf{a}^*_{m+}),\ldots,f_N(\mathbf{x},\mathbf{a}^*_{m+}))=\mathbf{S}_0}} \Pr(\mathbf{x}|\mathbf{a}^*_{m+}).$$

Now we can ascertain whether some attributes are statistically independent of the classification. Again without loss of generality, select attribute A_{m+1} for this purpose. What if *every* constituent algorithm agrees that A_{m+1} is independent of the classification, given the remaining known values $a^*_{m+2}, \ldots, a^*_L$? It seems natural and desirable that such a unanimous judgment of "irrelevance" should be preserved in the ensemble distribution. The following property formally captures this ideal:

Table 1. Example where plurality vote violates IPP.

A_1	A_2	A_3	$\mathbf{r}_1\mathbf{S}$	$\mathbf{r}_2\mathbf{S}$	$\mathbf{r}_3\mathbf{S}$	$\mathbf{S}_0$
0	0	0	$\langle 1,0\rangle$	$\langle 0,1\rangle$	$\langle 1,0\rangle$	$\langle 1,0\rangle$
0	1	0	$\langle 0,1\rangle$	$\langle 1,0\rangle$	$\langle 0,1\rangle$	$\langle 0,1\rangle$
1	0	0	$\langle 1,0\rangle$	$\langle 0,1\rangle$	$\langle 0,1\rangle$	$\langle 0,1\rangle$
1	1	0	$\langle 1,0\rangle$	$\langle 1,0\rangle$	$\langle 1,0\rangle$	$\langle 1,0\rangle$
$\Pr_i(\langle 1,0\rangle \vert A_3 = 0)$			0.75	0.5	0.5	0.5

A_1	A_2	A_3	$\mathbf{r}_1\mathbf{S}$	$\mathbf{r}_2\mathbf{S}$	$\mathbf{r}_3\mathbf{S}$	$\mathbf{S}_0$
0	0	1	$\langle 0,1\rangle$	$\langle 0,1\rangle$	$\langle 1,0\rangle$	$\langle 0,1\rangle$
0	1	1	$\langle 1,0\rangle$	$\langle 0,1\rangle$	$\langle 1,0\rangle$	$\langle 1,0\rangle$
1	0	1	$\langle 1,0\rangle$	$\langle 1,0\rangle$	$\langle 0,1\rangle$	$\langle 1,0\rangle$
1	1	1	$\langle 1,0\rangle$	$\langle 1,0\rangle$	$\langle 0,1\rangle$	$\langle 1,0\rangle$
$\Pr_i(\langle 1,0\rangle \vert A_3 = 1)$			0.75	0.5	0.5	0.75

Property 11 (IPP) Independence preservation property.

If $\Pr_i(\mathbf{r}_i\mathbf{S}|\mathbf{a}^*_{m+}) = \Pr_i(\mathbf{r}_i\mathbf{S}|a^*_{m+2}, \ldots, a^*_L)$ *for all i then* $\Pr_0(\mathbf{S}_0|\mathbf{a}^*_{m+}) = \Pr_0(\mathbf{S}_0|a^*_{m+2}, \ldots, a^*_L)$.

Table 1 presents a constructive proof that plurality vote fails to satisfy IPP. Three attributes each have domain $D_i = \{0, 1\}$, and the prior distribution over attribute values $\Pr(\mathbf{a}) = 1/8$ is uniform. Variables A_1 and A_2 have missing values (i.e., $m = 2$). Each of three constituent algorithms agree that the classification is independent of A_3. But combination by plurality vote destroys this independence: According to the ensemble, the classification *does* in fact depend on the value of A_3. Similar examples demonstrate that algebraic and geometric averages also violate IPP. It remains an open question whether any reasonable ensemble combination function can satisfy IPP. Results from statistics concerning generalized variants of IPP are mostly negative: No acceptable aggregation function has been found that preserves independence (Genest & Zidek, 1986), and several impossibility theorems severely restrict the space of potential candidates (Genest & Wagner, 1987; Pennock & Wellman, 1999).

7. Experimental Observations

We have shown, in theory, that the class of potential ensemble algorithms is severely limited if we want a small number of intuitive properties satisfied. One might argue that situations where these properties come into conflict may never arise in practice if we use popular aggregation methods. The purpose of this section is to show by example that, in fact, such conflicts do occur in practice. Specifically, we illustrate the paradoxical nature of some ensemble classification results in a stock market prediction domain, when the voting rule g fails to satisfy either IIA or transitivity.

We report results of empirical tests of an ensemble learner

Table 2. Six learned vote patterns, and the number of neural networks that learned each. An instance of the Borda paradox.

rank order	#	rank order	#
UP > SAME > DOWN	6	DOWN > SAME > UP	5
UP > DOWN > SAME	1	SAME > UP > DOWN	5
DOWN > UP > SAME	3	SAME > DOWN > UP	1

Table 3. Confidence scores and corresponding vote patterns for three neural networks. An instance of the Condorcet paradox.

i	S_{i1}	S_{i2}	S_{i3}	rank order
1	-0.33	-0.41	-0.25	SAME > UP > DOWN
2	-0.45	-0.25	-0.27	DOWN > SAME > UP
3	-0.31	-0.35	-0.37	UP > DOWN > SAME

trained on stock market data. We retrieved daily closing prices of the Dow between 1/20/97 and 1/18/00 from MSN Investor.[4] From this, we generated an approximately zero-mean and unit-variance time series of the form $\{d_t = 85(\ln p_t - \ln p_{t-1})\}$, where p_t is the Dow's price on day t. The attributes are $\mathbf{A} = \langle d_{t-5}, d_{t-4}, \ldots, d_{t-1} \rangle$. The classes are discrete intervals of d_t such that $C_1 =$ UP $\equiv (d_t > 0.35)$, $C_2 =$ DOWN $\equiv (d_t < -0.35)$, and $C_3 =$ SAME $\equiv (-0.35 \leq d_t \leq 0.35)$. The intervals are such that each class frequency is roughly $1/3$. The component learning algorithms are backpropagation neural networks built using Flake's (1999) NODELIB code library; each consists of an input layer of five nodes, a hidden layer of from one to seven nodes, and an output layer of three nodes. Diversity is due only to differences in the number of hidden nodes and to randomization in the training algorithm. The time series d_t was divided into a training set of 562 days and a test set of 187 days.

Table 2 shows the learned class rankings for twenty one networks (three each with $1, 2, \ldots, 7$ hidden nodes) on test day 7/14/99. If we use standard plurality vote to combine predictions, then DOWN wins with 8 votes, UP places in second with 7 votes, and SAME comes in last with 6 votes. By this measure we should short the Dow. But are we sure? Since SAME is presumably the least likely outcome, let's focus on the relative likelihoods between only DOWN and UP.[5] If we ignore SAME and recompute the vote, we find that UP actually beats DOWN by 12:9! This is a vivid demonstration that plurality vote violates IIA; the preference between UP and DOWN depends on SAME. So should we invest in the Dow? Well, the other two pairwise majority votes reveal that SAME beats UP by 11:10 and SAME beats DOWN by 12:9. Then according to the pairwise majority, SAME wins against both other classes, UP comes in second, and DOWN is last, completely reversing the original order predicted by the three-way plurality vote. This is an illustration of the so-called *Borda voting paradox*, named after the eighteenth century scientist who discovered it.

Table 3 demonstrates another classic voting paradox, due to Condorcet, one of Borda's peers. The table lists the activation values (confidence scores) of three networks (with one, two, and three hidden nodes) on test day 4/23/99. Plurality vote is tied, since each algorithm ranks a different class highest. What about pairwise majority vote? In this case, SAME beats UP by 2:1, and UP beats DOWN by 2:1. So is SAME our predicted outcome? Not necessarily—DOWN beats SAME, also by 2:1. We see that pairwise majority vote (which actually satisfies all five properties from Proposition 1) can return cyclical predictions, a violation of our generic definition of a classification $\Re^M$, which assumes that aggregation returns a transitive ordering of classes.

These two "paradoxes" illustrate the undesirable consequences of violating some of the basic properties of g defined earlier. The examples also constitute an existence proof that some of the same counterintuitive outcomes that have perplexed social scientists for centuries can and do occur in the context of ensemble learning.

8. Conclusion

We identified several properties of combination functions that social choice theorists and statisticians have found compelling, and argued their applicability in the context of ensemble learning. We cataloged common ensemble methods according to the properties they do and do not satisfy, and showed that no combination function can possess them all. We provided axiomatic justifications for weighted average combination, majority vote, and weighted majority vote. We described how common aggregation methods fail to respect unanimous judgments of independence. Finally, we exemplified the fundamental and unavoidable tradeoffs among the various properties using an ensemble learner trained on stock market data.

Drucker et al. (1993) present empirical evidence that weighted average outperforms plurality vote in some circumstances. Future work will examine whether the axiomatic framework developed in this paper can aid in deriving theoretical bounds on the performance of weighted average and other combination rules. We also plan to explore normative justifications for *individual* classifiers, and investigate whether, in some cases, a complex individual

[4] `http://moneycentral.msn.com/investor`

[5] Or we may have received outside information that discounts the likelihood of SAME.

classifier might reasonably be interpreted as an ensemble of simpler constituent classifiers.

Acknowledgements

Thanks to Gary Flake, Michael Wellman, and the anonymous reviewers. Pedrito Maynard-Reid II was partially supported by a National Physical Science Consortium Fellowship.

References

Arrow, K. J. (1963). *Social choice and individual values*. New Haven, CT: Yale University Press. 2nd edition.

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, *36*, 105–142.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–140.

Dietterich, T. G. (1997). Machine learning research: Four current directions. *AI Magazine*, *18*, 97–136.

Dietterich, T. G. (in press). An experimental comparison of three methods for construction ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*.

Dietterich, T. G., & Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, *2*, 263–286.

Drucker, H., Schapire, R., & Simard, P. (1993). Boosting performance in neural networks. *International Journal of Pattern Recognition and Artificial Intelligence*, *7*, 704–719.

Flake, G. W. (1999). *An industrial strength numerical modeling research and development tool* (Technical Report 99-085). NEC Research Institute.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 148–156).

Freund, Y., & Schapire, R. E. (1999). A short introduction to boosting. *Journal of the Japanese Society for Artificial Intelligence*, *14*, 771–780.

Genest, C., & Wagner, C. G. (1987). Further evidence against independence preservation in expert judgement synthesis. *Aequationes Mathematicae*, *32*, 74–86.

Genest, C., & Zidek, J. V. (1986). Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, *1*, 114–148.

Hansen, L., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*, 993–1001.

Jacobs, R. A. (1995). Methods for combining experts' probability assessments. *Neural Computation*, *7*, 867–888.

Jordan, M. I., & Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, *6*, 181–214.

Lee, D., & Srihari, S. N. (1995). A theory of classifier combination: The neural network approach. *Proceedings of the Third International Conference on Document Analysis and Recognition* (pp. 42–45).

Madhvanath, S., & Govindaraju, V. (1995). Serial classifier combination for handwritten word recognition. *Proceedings of the Third International Conference on Document Analysis and Recognition* (pp. 911–914).

May, K. O. (1952). A set of independent necessary and sufficient conditions for simple majority decision. *Econometrica*, *20*, 680–684.

Pennock, D. M., Horvitz, E., & Giles, C. L. (in press). Social choice theory and recommender systmes: Analysis of the axiomatic foundations of collaborative filtering. *Proceedings of the Seventeenth National Conference on Artificial Intelligence*.

Pennock, D. M., & Wellman, M. P. (1999). Graphical representations of consensus belief. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence* (pp. 531–540).

Roberts, K. W. S. (1980). Interpersonal comparability and social choice theory. *Review of Economic Studies*, *47*, 421–439.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, *5*, 197–227.

Sen, A. (1986). Social choice theory. In K. J. Arrow and M. D. Intriligator (Eds.), *Handbook of mathematical economics, vol. iii*, 1073–1181. Amsterdam: Elsevier.

Tax, D. M. J., Duin, R. P. W., & van Breukelen, M. (1997). Comparison between product and mean classifier combination rules. *Proceedings of the Workshop on Statistical Pattern Recognition*.

Tumer, K., & Ghosh, J. (1996). Error correlation and error reduction in ensemble classifiers. *Connection Science*, *8*, 385–404.

Wolpert, D. (1992). Stacked generalization. *Neural Networks*, *5*, 241–260.