MANAGEMENT SCIENCE
Vol. 55, No. 5, May 2009, pp. 827–842
ISSN 0025-1909 | EISSN 1526-5501 | 09 | 5505 | 0827

DOI 10.1287/mnsc.1080.0977
© 2009 INFORMS

Information Market-Based Decision Fusion

Johan Perols
University of San Diego, San Diego, California 92110, jperols@sandiego.edu

Kaushal Chari, Manish Agrawal
University of South Florida, Tampa, Florida 33620 {kchari@coba.usf.edu, magrawal@coba.usf.edu}

Improved classification performance has practical real-world benefits ranging from improved effectiveness in detecting diseases to increased efficiency in identifying firms that are committing financial fraud. Multiclassifier combination (MCC) aims to improve classification performance by combining the decisions of multiple individual classifiers. In this paper, we present information market-based fusion (IMF), a novel multiclassifier combiner method for decision fusion that is based on information markets. In IMF, the individual classifiers are implemented as participants in an information market where they place bets on different object classes. The reciprocals of the market odds that minimize the difference between the total betting amount and the potential payouts for different classes represent the MCC probability estimates of each class being the true object class. By using a market-based approach, IMF can adjust to changes in base-classifier performance without requiring offline training data or a static ensemble composition. Experimental results show that when the true classes of objects are only revealed for objects classified as positive, for low positive ratios, IMF outperforms three benchmarks combiner methods, majority, average, and weighted average; for high positive ratios, IMF outperforms majority and performs on par with average and weighted average. When the true classes of all objects are revealed, IMF outperforms weighted average and majority and marginally outperforms average.

Key words: multiclassifier combination; decision fusion; information markets; software agents
History: Received March 4, 2008; accepted November 16, 2008, by Ramayya Krishnan, information systems. Published online in *Articles in Advance* February 19, 2009.

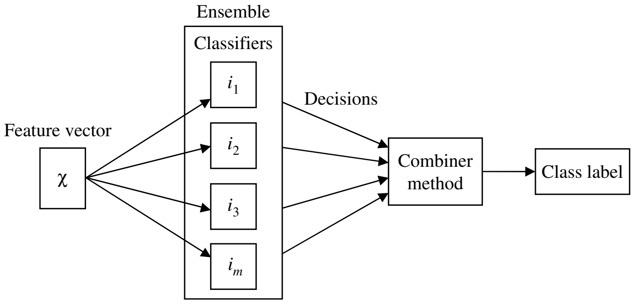
1. Introduction

In many decision-making scenarios, decisions of multiple human experts or classifiers are fused to determine the overall decision. Examples include a group of accounting experts and classifiers making going-concern decisions and an ensemble of classifiers in a fraud-detection application making decisions on whether a transaction is fraudulent. Multiclassifier combination (MCC) is a technique that can be used to improve the classification performance in various classification problems by combining the decisions of multiple individual classifiers (Suen and Lam 2000). In MCC, individual classifiers, commonly referred to as base classifiers, classify objects based on inputs consisting of object feature vectors (see Figure 1). These classifications or decisions are then combined using a combiner method into a single decision about the object's class label.

The basic premise behind MCC is that different classifiers in an ensemble have different strengths and weaknesses and therefore provide complementary information (referred to as diversity in MCC) about the classification problem. These differences can be leveraged to improve classification performance by combining base-classifiers' decisions (Kittler et al. 1998). Different combiner methods have been proposed and examined in the literature; these can be categorized based on whether they require training data. For example, naive Bayes, decision templates, and weighted average (WAVG) require training data, whereas average (AVG), majority (MAJ), and product do not. Existing combiner methods that require training data have limitations including the requirement for training data, and restrictive assumptions such as (1) constant ensemble base-classifier composition and (2) training data performance being a good proxy for subsequent actual performance. Experimental results generally indicate that MCC provides performance benefits and that the performance of MAJ and AVG methods is comparable or superior to that of methods requiring training (Duin and Tax 2000).

To improve performance while overcoming these limitations, we propose an information market-based fusion approach for multiclassifier combination that (1) has superior performance, (2) does not require offline training data, and (3) through online learning can adapt to changes in ensemble composition and base-classifier performance. In evaluating the effectiveness of our proposed approach, we compare information market-based fusion (IMF) against three combiner methods, AVG, MAJ, and WAVG. These methods have performed relatively well in prior

Figure 1 Generic Classifier Combiner Architecture



research (Duin and Tax 2000) and have been used as benchmarks in recent MCC research.[1] For example, Zheng and Padmanabhan (2007) use AVG, which they refer to as unweighted average, and a version of WAVG with variance-based weights, which they refer to as variance-based weighting. Our experimental evaluation was performed using computational experiments with 17 data sets that were obtained from the University of California, Irvine Machine Learning Repository (Newman et al. 1998) and 22 different base classifiers from Weka (Witten and Frank 2005).

The remainder of this paper is organized as follows. In §2, we provide a review of related research. IMF is introduced in §3 along with an overview of information markets. We then present details on the computational experiments and results in §§4 and 5, respectively. In §6, we discuss these results and conclude in §7 with a review of our contributions and suggestions for future research.

2. Related Research

A classifier is a model that makes decisions about an object's class membership based on the object's feature set. Examples of classifiers include neural networks, logistic regression, decision trees, and Bayesian classifiers (Witten and Frank 2005). Classifier performance is typically dependent on the problem domain as well as on the calibration of the classifier. Multiple classifiers are therefore typically tested to identify the best classifier for a given problem domain. However, it is generally difficult to determine which classifier(s) will perform well in subsequent classifications. Furthermore, classification for certain cases may even be improved by an "inferior" classifier (Kittler et al. 1998). Thus, by combining the decisions of diverse classifiers, it is possible to improve the overall performance.

Prior MCC research has primarily focused on one of two areas: (1) training and selection of ensemble base classifiers or (2) combination of base-classifier decisions. Methods such as bagging, boosting, and stacking fall into the first category (Witten and Frank 2005); combiner methods such as MAJ, AVG, and WAVG fall into the second category. Recent research within the former stream has used receiver operating characteristic analysis to select dominant classifiers (Provost and Fawcett 2001) and data envelopment analysis to select efficient classifiers (Zheng and Padmanabhan 2007) under various cost and class distributions, and then combine these classifiers' decisions. Fan et al. (1999) have adapted AdaBoost to an online learning environment where new training instances become available continuously or periodically. This paper does not focus on classifier selection and training, but on the combiner methods. Prior research within the combiner method research stream has found that methods that use measurement data are typically more accurate than methods that handle unique labels; methods that require training data typically outperform methods that do not require training (Jain et al. 2000), but that MAJ and AVG, which do not require training, perform either at the same level or significantly better than more complex methods (Duin and Tax 2000).

Another important but largely overlooked aspect of combiner methods is how well they fit with different system architectures. Software agents offer a new paradigm to support decision making (Nissen and Sengupta 2006) where human-driven or autonomous software agents embodying classifiers and other intelligent algorithms can leverage their individual strengths to make collective decisions. The base-classifiers, combiner method, and providers of object features in an MCC can be implemented as software agents in multiagent systems. Research in data mining has implemented MCC agent systems for credit card fraud detection (Stolfo et al. 1997) and network intrusion detection (Lee et al. 2000).

In MCC multiagent systems that are implemented in dynamic real-world settings, the relative performance of base classifiers and the ensemble composition can change over time as agents are retired, added, or temporarily unavailable. Existing combiner methods that require offline training do not take this into consideration and assume that the ensemble composition is static and that individual classifier performance does not change subsequent to training and validation.

3. Information Market-Based Fusion

IMF is theoretically grounded in information markets. More specifically, the IMF aggregation mechanism

[1] In Duin and Tax (2000), AVG is referred to as mean and MAJ is referred to as majority; eight additional combiner methods are evaluated: Bayes rule (two different implementations), nearest mean, nearest neighbor, maximum, median, minimum, and product. When combining the decisions of different base classifiers trained using the same feature set, which is comparable to the MCC architecture that we use, their results (p. 23) show that majority and mean perform on par with or better than the other combiner methods.

used in this paper is based on parimutuel betting markets.

3.1. Information Markets

Information markets are markets specifically designed for the purpose of information aggregation. Equilibrium prices, derived using conventional market mechanisms, provide information based on private and public information maintained by the market participants about a specific situation, future event, or object of interest (Hanson 2003). Although the concept of information markets is fairly recent, the underlying notion of markets being capable of aggregating information is not new (Hayek 1945), and the efficient market hypothesis states that all private and public information is reflected in equilibrium prices (Fama 1970). Empirical research has found support for the efficient market hypothesis and for information aggregation in information markets in general (Berg and Rietz 2003), and parimutuel betting markets in particular (Plott et al. 2003).

Our combiner method is based on parimutuel betting, which originated in horserace gambling in France in 1865, and since then has become a popular betting mechanism in the horseracing world. Pari-mutuel means "wager mutual" and comes from the fact that in parimutuel betting, a winning wager (i.e., bet) receives a share of the total wagers (winning and loosing bets minus a track commission) as a proportion of this winning wager to all winning wagers. The final track odd for a given horse is the total amount bet on all the horses in the race divided by the total amount bet on the given horse. The payout for a winning horse is the product of the amount bet on it and its odd (less track commission). From an MCC perspective, the odd associated with a horse is of great importance, as it represents the aggregated market information about the probability estimate of that horse winning the race. We use parimutuel betting over mechanisms such as continuous double auctions because parimutuel betting does not suffer from liquidity problems that could potentially impact continuous double auction markets when there are large bid-ask spreads or when bid-ask queues are empty (Pennock 2004). Hence, parimutuel mechanisms would work effectively, even when the ensemble of base-classifiers is small.

Plott et al. (2003) experimentally examined information aggregation and different betting behaviors in parimutuel betting markets using two private information models—decision theory private information (DTPI) and competitive equilibrium private information (CEPI)—and one model with belief updating—competitive equilibrium rational expectations. Plott et al. (2003) found that DTPI and CEPI best described the behavior of human participants in their probabilistic information condition experimental parimutuel betting market.

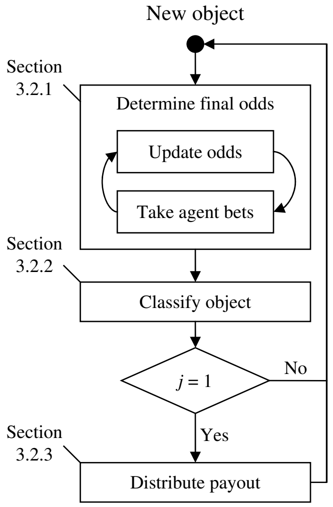
In DTPI, agents only consider their own private information and ignore market prices when deciding on their bets and in forming beliefs. In CEPI, agents base their bets on the current market price, although they do not update their beliefs based on market prices. In both models, agents maximize their conditional expected utility given their private probability estimates and constraints such as available funds. In both DTPI and CEPI, prices are assumed to be in equilibrium; however, as each betting round starts without prices defined, the equilibrium must be obtained before the agents can place their final bets. Assuming no track take, in equilibrium, all potential payouts are equal to the total amount that is bet across all events.

3.2. Information Market-Based Fusion

IMF is a multiclassifier combiner method based on a parimutuel betting information market that can be used in any classification application domain. We present IMF in the context of a fraud-detection application. In this application, object t (i.e., transaction t) can be classified as fraudulent ($j = 1$) or nonfraudulent ($j = 2$) by an ensemble E of agent classifiers. In this application, the set $J = \{1, 2\}$ is the index set of the two classes (i.e., fraudulent and nonfraudulent). The ensemble E has m agents embodying different base classifiers (referred to as agents) represented by indices i in the index set $D = \{1, \ldots, m\}$. While determining the class membership of object t, agent $i \in D$ uses the feature vector associated with t to determine the posterior probability estimate $p_{itj} \in [0, 1]$ that t belongs to class $j \in J$. Agent i bets q_{itj} that object t belongs to class j and is paid according to the parimutuel mechanism based on four factors: (1) the agent's bets, q_{itj}; (2) the total bets on class j, $Q_{tj} = \sum_{i\in D} q_{itj}$; (3) the total bets on all classes, $Q_t = \sum_{j\in J} \sum_{i\in D} q_{itj}$; and (4) the true class of object t. Ensemble E's overall probability estimate that t belongs to $j \in J$ is given by $1/O_{tj} \in [0, 1]$, where O_{tj} is the odd that t belongs to $j \in J$. The odd O_{tj}, which is equal to Q_t/Q_{tj}, is in equilibrium when the potential payouts $Q_{tj}O_{tj}$ for each $j \in J$ and the total bets Q_t are equal (assuming no house commission); i.e., O_{tj} is in equilibrium when $Q_{tj}O_{tj} = Q_t$.

Figure 2 provides an overview of IMF when the true class of objects is only determined for objects classified as positive. When all objects are investigated, Figure 2 is changed by eliminating the decision box, i.e., going straight from *classify object* to *distribute payout*. Investigations are, however, expensive, and in the real world only objects classified as positive are typically investigated. Unless otherwise noted, we will henceforth assume that only objects classified as positive are investigated.

Figure 2 IMF Flowchart



In Figure 2, for each new object t, IMF first determines the final odds O_{tj}^f that are equilibrium or near-equilibrium odds. Establishing equilibrium odds is a nontrivial task because of the recursive relationship between Q_{tj} and O_{tj}, where odds are based on agent bets and agents base their bets on odds. Therefore, multiple rounds of betting are required to determine the final odds that can then be used by agents to make their actual bets. In each round, odds are first updated based on all the agents' prior bets and then agents place new bets based on the current updated odds. After the final odds have been established, ensemble E's overall probability estimate $1/O_{tj}^f$ is compared to a threshold value C_j to determine if t should be classified as belonging to class j. If object t is classified as fraudulent ($j = 1$), then the true class of t is determined and winnings are distributed to the agents.

In addition to establishing ensemble E's probability estimate $1/O_{tj}^f$, IMF facilitates the redistribution of wealth among the agents based on the agents' bets and winnings. From an MCC perspective, IMF produces decisions that are wealth-weighted probability estimates of the occurrence of event j. We next describe the components of IMF in detail as per the major steps depicted in Figure 2.

3.2.1. Determining Final Odds. The problem of determining odds O_{tj} for object t is given by P1:

$$\text{P1:} \qquad Z_1 = \min_{O_{tj} M_j} \sum_{j \in J} M_j \tag{1}$$

$$\text{s.t.} \quad Q_{tj} O_{tj} - M_j = Q_t \quad \forall j \in J \tag{2}$$

$$M_j \geq 0 \text{ and } O_{tj} \geq 1 \quad \forall j \in J. \tag{3}$$

The objective function Z_1 minimizes dummy variables M_j that represent the differences between the total bets by all agents and the total payout for each outcome (2). At equilibrium, M_1 and M_2 are equal to zero.

Figure 3 Binary Search

```
Set search space bounds
$P^l = 0$ and $P^u = 1$
set $O_{t1} = 2/(P^l + P^u)$
Take agent bets
Do
  If $Q_{t1} O_{t1} > Q_t$ then
    set $P^l = 1/O_{t1}$
  else if $Q_{t1} O_{t1} < Q_t$ then
    set $P^u = 1/O_{t1}$
  else if $Q_{t1} O_{t1} = Q_t$ then
    set $P^l$ and $P^u$ to $1/O_{t1}$
  set $O_{t1} = 2/(P^l + P^u)$
  Take agent bets
Until ($P^u - P^l \leq \varepsilon$)
```

Because of the recursive relationship between Q_{tj} and O_{tj}, we solve P1 using binary search (see Figure 3) to determine equilibrium or near-equilibrium odds. Binary search starts with a lower bound $P^l = 0$ and an upper bound $P^u = 1$ for the probability that object t belongs to class $j = 1$. O_{t1} is then computed using $O_{t1} = 2/(P^l + P^u)$. It can be easily verified that $O_{t2} = O_{t1}/(O_{t1} - 1)$ in the case of two class problems. The agents then place bets that maximize their individual utility, given their current wealth and probability estimates and the current odds (Lemmas 1 and 2 describe the optimal bets).

The odds and bets are then used to evaluate whether the current odds are too high or too low. If the potential payout for $j = 1$, i.e., $Q_{t1} O_{t1}$, is greater than the total bets Q_t, then odd O_{t1} is too high, and the lower search space boundary P^l is raised to the reciprocal of O_{t1}; i.e., $P^l = 1/O_{t1}$. However, if the potential payout for $j = 1$ is less than the total bets Q_t, then the odd O_{t1} is too low and the upper search space boundary P^u is lowered to the reciprocal of O_{t1}; i.e., $P^u = 1/O_{t1}$. If the potential payout for $j = 1$ is the same as the total bets Q_t, then the potential payouts for $j = 1$ and $j = 2$ are equal; i.e., $Q_{t1} O_{t1} = Q_{t2} O_{t2}$, the odds are in equilibrium, and the search space is set to this single value $P^l = P^u = 1/O_{t1}$. O_{t1} is then set to the reciprocal of the mean of P^l and P^u and the agents place bets based on these odds. The updating of odds and agent bets continues iteratively until the search space is within tolerance ε, i.e., $P^u - P^l \leq \varepsilon$. When a binary search terminates, it is known that the optimal odds are within bounds $1/P^u$ and $1/P^l$.

3.2.1.1. Determining Agent Bets. Given the current market odds O_{tj}, the agent's probability estimates p_{itj} of object t being in class j, the agent's current wealth w_{it} plus the periodic endowment m, and multiplier k that determines the house-enforced maximum bet km, agent i solves the expected utility maximization problem P2 to determine the amount q_{itj} to bet on classes $j = 1, 2$. The periodic endowment m is given to all the agents to prevent them from running out of funds. Given the utility function U_i of agent i as a function of wealth, problem P2 can be stated as follows:

$$\text{P2:}\quad Z_2 = \max_{q_{itj}} p_{it1} U_i(w_{it}+m-q_{it1}-q_{it2}+q_{it1}O_{t1}) + p_{it2} U_i(w_{it}+m-q_{it1}-q_{it2}+q_{it2}O_{t2}) \quad (4)$$

$$\text{s.t.}\quad q_{it1}+q_{it2} = \begin{cases} w_{it}+m & \text{if } (w_{it}+m)\le km, \\ km & \text{if } (w_{it}+m) > km, \end{cases} \quad (5)$$

$$q_{itj} \ge 0. \quad (6)$$

The objective function in P2 represents the expected utility of agent i when it bets $q_{itj} \ge 0$ on event j. Constraint (5) dictates that the total amount of bets placed by agent i on events $j=1$ and $j=2$ equals the lower of the agents' available funds $m+w_{it}$, and the house-enforced maximum bet km. The house-enforced maximum bet km limits the amount of influence the best performing agents in the ensemble could exert on ensemble decision, due to the need to have all agents, not just the best performing agents, contribute to improving the success of the ensemble (Kittler et al. 1998).

P2 is general enough to incorporate any utility function to model an agent's risk aversion. We utilize a natural logarithm (ln) utility function (henceforth simply referred to as log utility), which has been widely used in prior research (Rubinstein 1976), for the following reasons: (1) log utility enables agents to place bets that yield optimal long-run growth rates (Kelly 1956); (2) it is twice-differentiable and nondecreasing concave, leading to a decreasing absolute risk aversion (Rubinstein 1976); and (3) depending on which betting constraint is binding (see Lemmas 1 and 2 below), log utility bets are either increasing in p_{itj} and $w_{it}+m$ but not a function of O_{tj}, a betting behavior corresponding to DTPI (Plott et al. 2003), or increasing in p_{itj}, $w_{it}+m$, and O_{tj}, a betting behavior corresponding to CEPI (Plott et al. 2003).

Using log utility, problem P2 is transformed to either P3 or P4 depending on the binding constraint in (5) for a given agent. If $w_{it}+m \le km$, then $q_{it1}+q_{it2} = w_{it}+m$ and $w_{it}+m-q_{it1}-q_{it2}=0$, leading to P3:

$$\text{P3:}\quad Z_3 = \max_{q_{itj}} \; p_{it1}\ln(q_{it1}O_{t1}) + p_{it2}\ln(q_{it2}O_{t2}) \quad (7)$$

$$\text{s.t.}\quad q_{it1}+q_{it2} = w_{it}+m \quad (8)$$

$$q_{itj} \ge 0. \quad (9)$$

Lemma 1. *The optimal bet of agent i in P3 while classifying t is* $q^*_{itj} = p_{itj}(w_{it}+m)\ \forall j \in J$.

Proof. See the e-companion.[2]

If $w_{it}+m > km$, then $q_{it1}+q_{it2} = km$ and $w_{it}+m-q_{it1}-q_{it2} = w_{it}+m-km$, which we denote by constant a_{it}. Thus P2 can be transformed to P4:

$$\text{P4:}\quad Z_4 = \max_{q_{itj}} \; p_{it1}\ln(a_{it}+q_{it1}O_{t1}) + p_{it2}\ln(a_{it}+q_{it2}O_{t2}) \quad (10)$$

$$\text{s.t.}\quad q_{it1}+q_{it2} = km \quad (11)$$

$$q_{itj} \ge 0. \quad (12)$$

Lemma 2. *The optimal bets of agent i in P4 while classifying t is as follows*:

Solution a:

$$q^*_{it1} = p_{it1}km + a_{it}\frac{p_{it1}O_{t1}-p_{it2}O_{t2}}{O_{t1}O_{t2}} \quad \textit{and}$$

$$q^*_{it2} = p_{it2}km + a_{it}\frac{p_{it2}O_{t2}-p_{it1}O_{t1}}{O_{t1}O_{t2}}, \quad \textit{when}$$

$$0 < p_{it1}km + a_{it}\frac{p_{it1}O_{t1}-p_{it2}O_{t2}}{O_{t1}O_{t2}} < km \quad \textit{and}$$

$$0 < p_{it2}km + a_{it}\frac{p_{it2}O_{t2}-p_{it1}O_{t1}}{O_{t1}O_{t2}} < km;$$

Solution b:

$$q^*_{it1} = km \quad \textit{and} \quad q^*_{it2} = 0, \quad \textit{when}$$

$$km \le p_{it1}km + a_{it}\frac{p_{it1}O_{t1}-p_{it2}O_{t2}}{O_{t1}O_{t2}}; \quad \textit{and}$$

Solution c:

$$q^*_{it2} = km \quad \textit{and} \quad q^*_{it1} = 0, \quad \textit{when}$$

$$km \le p_{it2}km + a_{it}\frac{p_{it2}O_{t2}-p_{it1}O_{t1}}{O_{t1}O_{t2}}.$$

Proof. See the e-companion.

Equilibrium Odds. The final odds O^f_{tj} are equilibrium odds or near-equilibrium odds, where near equilibrium is defined as being within bounds $1/P^u$ and $1/P^l$, and $P^u - P^l$ is less than or equal to tolerance ε. When binary search terminates, it is known that the optimal odds are within these bounds. As described in binary search, the final odds O^f_{tj} are found for each object t by iteratively updating the odds and requiring agents to place bets using these odds until the odds provided to the agents and their subsequent bets result in $Q_{tj}O^f_{tj} \approx Q_t$, at which time the market closes. The following observations are made. First, bets placed in betting rounds before the final odds have been established are only used for the purpose of updating the odds.[3] Second, if agent bets are discontinuous over O_{tj} then the existence of equilibrium odds cannot be

[2] An electronic companion to this paper is available as part of the online version that can be found at http://mansci.journal.informs.org/.

[3] We make the assumption that agents do not act strategically by attempting to bluff about their private information, i.e., placing bets that do not maximize their utility given the current odds.

guaranteed (Carlsson et al. 2001). Lemma 3 shows that in IMF, when agents $i \in D_1$ bet as per Lemma 1 and agents $i \in D_{2a}$, $i \in D_{2b}$, and $i \in D_{2c}$ bet as per Lemma 2, solutions a, b, and c, respectively, then equilibrium exists. However, even when equilibrium odds do exist, IMF may not always find it because of the recursive nature of O_{tj} and Q_{tj}. Binary search used in IMF nevertheless guarantees a result that is at most ε (a tolerance parameter) from the optimal probability.

Lemma 3. *Given any combination of betting behaviors as per Lemmas* 1 *and* 2, *equilibrium exists, and the equilibrium odd for* $j = 1$ *is*

$$O_{t1} = \frac{\sum_{i\in D_1\cup D_{2a}} p_{it2}(w_{it}+m) + \sum_{i\in D_{2c}}(km)}{\sum_{i\in D_1\cup D_{2a}} p_{it1}(w_{it}+m) + \sum_{i\in D_{2b}}(km)} + 1.$$

Proof. See the e-companion. The e-companion also contains both an empirical evaluation of IMF when agent bets are discontinuous over O_{tj} and an empirical evaluation of the ability of IMF to find the equilibrium odds when the agents bet as per Lemmas 1 and 2.

3.2.2. Classifying Objects. Once the final odds O^f_{t1} and O^f_{t2} are available, we can use the decision rule in (13) to classify[4] object t.

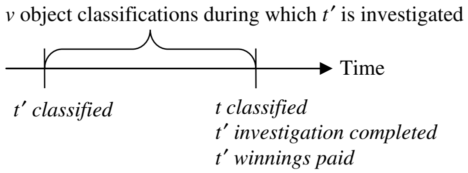
$$\text{If } (1/O^f_{t1} \geq C_1), \quad \text{then classify class of } t \text{ as } j=1; \quad \text{else classify class of } t \text{ as } j=2. \tag{13}$$

In (13), if the reciprocal of final odd for $j = 1$ is higher than the threshold C_1, then object t is classified as a member of the positive class; i.e., $j = 1$ and agent i's wealth is decreased by the amount of i's final bets:

$$w_{it} = w_{it} - \sum_{j\in J} q_{itj}. \tag{14}$$

The true class of t is then investigated, and agent i's wealth is updated with any potential winnings (see §3.2.3). If the object is classified as a member of the negative class, i.e., $j = 2$, then the verification of the object class is not pursued further, as investigations are not typically carried out for negative classifications. In this case, agent wealth is not updated with bets or winnings.

Figure 4 Payout Distribution Time Lag


v object classifications during which t′ is investigated
Time
t′ classified
t classified
t′ investigation completed
t′ winnings paid

3.2.3. Distributing Payout. Whenever object t is classified as belonging to the positive class, detailed investigations are necessary to establish the true class of t. Although final bets are deducted from the agents' wealth immediately, because of the time taken for investigations, there is a time lag corresponding to v elapsed object classifications before winnings can be paid out. This mechanism is similar to sports betting (and other types of futures markets), where bets are collected when bets are placed and winnings are paid out after the game/race has been decided. In Figure 4, t is the current object being classified, t' is the object for which the investigation has just been completed, and v is the number of objects that have been classified since t' was classified. Based on the investigation, if t' is found to be a positive, then agent i's wealth is updated using (15), or else t' is negative and agent i's wealth is updated using (16). This is also followed when the true classes of both positive and negative classifications are investigated.

$$w_{it} = w_{it} + (q_{it'1}/Q_{t'1})Q_{t'}, \tag{15}$$

$$w_{it} = w_{it} + (q_{it'2}/Q_{t'2})Q_{t'}. \tag{16}$$

4. Experimental Setup

4.1. Base Classifiers and Data

Using Weka (version 3.3.6), 22 heterogeneous base-classifiers were created using their default settings (see Table 1). The base classifiers were trained and evaluated using 10-fold cross validation on each of 17 data sets obtained from the UCI Machine Learning Repository (see Table 2). Data sets that included more than two classes were modified by either creating multiple

Table 1 Base Classifiers

ADTree
BayesNet
ConjunctiveRule
DecisionStump
DecisionTable
Ibk
J48
JRip
KStar
LMT
LWL
MultilayerPercep.
NaiveBayes
NBTree
Nnge
OneR
PART
RandomForest
RBFNetwork
Ridor
SimpleLogistic
SMO

This assumption is made to make the utility maximization problem more tractable. In defense of this assumption, the agents do not know when the market closes; i.e., they never know if the current odds are the final odds, and strategic behavior is therefore less likely even if allowed. Furthermore, Plott et al. (2003) found that strategic behavior was negligible among their human subjects in their probabilistic information condition experiment even though the subjects knew that the market would stay open at least until an announced time.

[4] MCC users might prefer rankings or raw probabilities (Saar-Tsechansky and Provost 2004). In these situations the generated ensemble probability estimates can be presented directly to the users.

Table 2 Data Sets

					Base-classifier accuracy (%)			
Data set	Instances	Attributes	Positive rate (%)	Ensemble diversity	Min.	Avg.	Max.	Std.
Adult	32,561	14	24.1	0.847	75.9	82.0	86.0	3.5
Wisconsin breast cancer	699	110	34.5	0.894	76.8	94.0	97.1	4.2
Contraceptive choice	1,473	10	57.3	0.698	60.0	66.1	71.0	2.6
Horse colic	368	22	37.0	0.868	78.0	81.7	86.1	2.3
Covertype (class 1 and 2)	10,000	11	72.9	0.883	78.9	86.6	92.1	3.8
Covertype (class 3 and 4)	10,395	11	6.8	0.954	93.0	95.2	97.5	1.6
Covertype (class 5 and 6)	10,009	11	66.9	0.906	89.9	96.2	99.7	3.5
Australian credit approval	690	15	55.5	0.857	76.2	83.7	85.8	2.6
German credit approval	1,000	20	30.0	0.778	63.7	72.0	75.7	2.9
Pima Indians diabetes	768	8	34.9	0.842	68.8	73.5	77.9	2.5
Thyroid disease	3,772	5	7.7	0.994	89.1	92.7	93.4	1.2
Labor	57	16	64.9	0.487	68.4	80.0	93.0	7.4
Mushrooms	8,124	5	48.2	0.775	77.0	91.2	96.6	7.4
Sick	3,772	12	6.9	0.956	93.9	96.8	98.4	1.2
Spambase	4,601	58	39.4	0.708	78.9	87.6	94.2	5.6
Splice-junction gene sequence	3,190	20	51.9	0.498	53.4	62.3	67.1	3.7
Waveform	3,345	40	49.4	0.790	77.6	86.9	92.7	4.5

subsets with only two classes in each subset or by combining classes. For computationally complex base classifiers to complete the classification using a reasonable amount of resources, data sets with a large number of observations and/or attributes were filtered randomly based on records and/or attributes.[5]

With an average data set size of 5,350 records, a total of 2,000,900 base-classifiers' validation decisions (5,350 records × 17 data sets × 22 base classifiers) were generated from the 10-fold cross validation. These decisions were imported into Microsoft Access, where combiner methods, implemented using Visual Basic, combined the data. Furthermore, because IMF, MAJ, AVG, and dynamic WAVG did not require training, we did not use n-fold cross-validation in the combiner method experiments. Each data set was combined 96 times, as described in §4.2, for a total of 8,745,888 (96 data set combinations × 5,350 average data set size × 17 data sets) ensemble decisions.

4.2. Experimental Design and Factors

The primary purpose of the computational experiments was to compare the effectiveness of IMF against MAJ, AVG, and WAVG methods. As such, the combiner method is our primary factor of interest. Our experiment also included six other independent variables, two factors (cost-to-benefit ratio and number of agents), and four covariates (data set positive ratio, data set size, data set average base-classifier accuracy and ensemble diversity), used to evaluate the sensitivity of our results (see Table 3).

As we were only investigating main effects and second order interactions, and only interactions involving the combiner method factor, we did not need a full factorial design. Instead, we used two factorial block designs (4 combiner methods × 11 sets of agents, and 4 combiner methods × 13 cost-to-benefit ratios), for a total of 96 treatment groups. The cost-to-benefit ratio factor was held constant at 1:10 in the 4 × 11 factorial design. The number of agents factor was held constant at 10 in the 4 × 13 factorial design. Net benefit and the covariates were measured for each of the 17 data sets within each of the 96 treatment cells for a total of 1,632 observations ($4 \times 11 \times 17 + 4 \times 13 \times 17$).

4.2.1. Dependent Measure. Performance measures used in the MCC combiner method research include hit-rate (TP/(TP + FN)) and accuracy ((TP + TN)/(TP + TN + FP + FN)), where TP is true positive, FP is false positive, TN is true negative, and FN is false negative. However, accuracy and hit-rate only provide accurate measures of combiner method effectiveness under one specific scenario—when the number of positive and negative instances is the same and the cost of FP and FN is the same. This is rarely true (Provost et al. 1998). More recently, receiver operating characteristic (ROC) curves and area under the ROC curve (AUC) have gained popularity, partially because they show how well algorithms handle the trade-off between true positive rate (TP/(TP + FN)) and false positive rate (FP/(FP + TN))—i.e., benefits and costs—without having to define a specific class distribution and cost assumption. ROC and AUC do not, however, allow for easy comparisons of combiner methods under specific distribution and

[5] The number of attributes/instances to delete was determined iteratively by first deleting only a few attributes/instances, running the most resource-constraining base-classifier algorithm, and then deleting more attributed/instances if needed. Attributes/instances selected for deletion were determined by assigning a random number to each attribute/instance using Microsoft Excel and then deleting the attributes/instances assigned the lowest number.

Table 3 Experimental Variables

Variable	Function	Description
Net benefit	Dependent variable	FN cost avoidance $*$ number of TP $-$ investigation cost $*$ (number of FP $+$ number of TP)
Combiner method	Main independent variable	IMF, AVG, WAVG, MAJ
Number of agents	Manipulated moderator	$2, 4, 6, \ldots, 22$ agents in the ensemble
Cost-to-benefit ratio	Manipulated moderator	1:100; 1:50, 1:25, 1:10, 1:7.5, 1:5, 1:4, 1:3, 1:2, 1:1.5, 1:1, 1.5:1, and 2:1
Data set size	Measured moderator	Number of data set records
Data set average agent accuracy	Measured moderator	Average data set accuracy of the base classifiers
Data set positive ratio	Measured moderator	Positive records/total number of records in data set
Ensemble diversity	Measured moderator	Data set average pairwise diversity measured using Yule's Q statistic

Note. FN, false negative; TP, true positive.

cost assumptions that we are interested in. ROC also does not provide a single measure that allows us to assess the statistical significance and sensitivity of the relative combiner method performance results to various factors such as the number of base classifiers in the ensemble, cost-to-benefit ratio, and data set size, average agent accuracy, positive ratio, and diversity (Drummond and Holte 2006). Furthermore, ROC curves are created using the true positive rate and false positive rate and therefore cannot be used in situations where TN and FN are not identified, i.e., in domains where negative classifications are not investigated.

Another common performance measure—misclassification cost, which is the total cost of FP and FN classifications—overcomes many of these shortcomings (Lin et al. 2003). However, this measure still requires knowing FN and ignores the costs associated with TP classifications, such as investigation costs. Chan et al. (1999) use cost savings (CS), which takes into account costs associated with TP, FP, and FN but still requires knowing FN. We use a measure very similar to CS[6] that we call net benefit (NB). NB, like CS, allows us to overcome the problems described earlier and, in contrast to CS, does not require knowing FN. NB is calculated as the benefit derived from TP classifications (FN costs avoided) minus costs of investigating positive classifications; see (17). Like ROC curves, NB captures the trade-off between true positive rate and false positive rate. To maximize NB, the classification threshold has to be selected so that it strikes an appropriate balance between net benefit of TP and cost of FP classifications:

$$\begin{aligned}\text{NB} = {} & \text{FN cost avoidance} * \text{number of TP}\\ & - \text{investigation cost}\\ & * (\text{number of FP and TP}). \qquad (17)\end{aligned}$$

In our experiments, we compare the performance of various combiner methods using optimal thresholds for each treatment in order to isolate the treatment effect from noise introduced by using other mechanisms to determine the threshold. To determine the optimal thresholds, we run the MCC experiment 101 times for each treatment, using a different threshold level $(0, 0.01, 0.02, \ldots, 1)$ for each run. The threshold from the run that generates the highest total NB is then labeled as the optimal threshold for that specific treatment. By finding the best threshold for each combiner method, data set, ensemble, and cost-benefit ratio combination, we compare the combiner methods at optimal trade-off levels for that specific combination, which we believe is more relevant than comparing the sensitivity, specificity, hit-rate, etc., of the combiner methods at other suboptimal levels. Furthermore, by comparing the combiner methods at a number of different cost-to-benefit ratios, we improve the generalizability of our results to different domains that have different cost-to-benefit ratios.

4.2.2. Combiner Method Factor. Because our primary objective is to compare the performance of IMF to existing combiner methods, we include the combiner method as a factor that is manipulated at four levels: IMF, AVG, MAJ, and WAVG. IMF is compared to MAJ, AVG, and WAVG, because prior research indicates that AVG and MAJ perform well compared to other existing combiner methods (Duin and Tax 2000). WAVG is included primarily because of its similarity to IMF, because IMF generates a wealth

[6] It is shown in the e-companion that NB is equivalent to CS, given that transaction amount and overhead (Chan et al. 1999) are defined as being equivalent to FN cost avoidance and investigation cost, respectively. Also note that our definition of cost-to-benefit ratio is based on the same idea used in the Chan et al. (1999) rule: only transactions with transaction amounts > overhead should be investigated.

weighted average. In MAJ, each base classifier casts a vote on the class for which the base-classifier's probability estimate is higher than the classification threshold. The class with the most votes is then selected as the ensemble's decision. In AVG the mean of all the base-classifiers probability estimates is compared to the threshold, and the class with a mean probability estimate that is higher than the threshold is selected as the ensemble's decision. In WAVG, different weights are assigned to the different base-classifiers' probability estimates when averaging these estimates. To maintain uniformity while comparing IMF to WAVG, we implement a dynamic version of WAVG, where the weights are updated based on positive classifications only. The weights are determined as the ratio of an individual classifier's precision $(\text{TP}/(\text{TP}+\text{FP}))$ to the total precision of all the classifiers in the ensemble. We also test two alternative weighting schemes, as detailed in §4.2.4.

4.2.3. Sensitivity Analysis

Number of Agents. The number of agents factor is manipulated at 11 levels: $2, 4, 6, \ldots, 22$ agents. This manipulation is done because there is evidence from prior research that the number of agents in an ensemble could impact ensemble classification performance (Lam 2000). The agents are randomly selected at each of the treatment levels, but the selection process is cumulative in nature. For ensembles consisting of two agents, the two agents are randomly selected from the 22 existing base classifiers; for ensembles with four agents, two additional agents are randomly selected from the remaining 20 base classifiers and added to the existing ensemble, and so on. To test the sensitivity of the combiner method performance to the number of agents, we examine whether the relative combiner method performance is moderated by the number of agents, while holding the cost-to-benefit ratio constant at 1:10.

Cost-to-Benefit Ratio. The benefit derived from TP classifications (FN cost avoidance minus investigation cost) and the cost of FP classifications (investigation cost) impacts the net benefit provided by any classification effort. As the cost-to-benefit ratio is domain specific, we use a wide range of cost-to-benefit ratios—13 in total—to explore the generalizability of our results: 1:100; 1:50, 1:25, 1:10, 1:7.5, 1:5, 1:4, 1:3, 1:2, 1:1.5, 1:1, 1.5:1, and 2:1. To clarify, the 1:100 ratio indicates that the net benefit of a TP classification (cost of fraud minus investigation costs of detecting a fraud) is 100 times the cost of investigating a transaction (for example, cost of fraud $=$ \$10,100 versus cost of investigation $=$ \$100). Note that the range of cost-to-benefit ratios used assumes that the net benefit of a TP is always positive; i.e., the FN cost avoided when making a TP classification is always more than the investigation cost. To examine the sensitivity of the combiner method performance to cost-to-benefit ratio, we investigate whether the relative combiner method performance is moderated by cost-to-benefit ratio, holding the number of agents constant at 10 agents.

Data Set Average Accuracy. Average base-classifier accuracy, measured as the percentage of all objects classified correctly for each data set, is included as a possible interaction term, given the possibility that relative combiner method performance could depend on the average accuracy of the base classifiers in a given data set. Thus, this interaction tests whether the relative combiner method performance is moderated by the data set average base-classifier accuracy.

Data Set Size. Data set size refers to the number of records in the data set, which varies from 57 to 32,561. Data set size is included as a covariate primarily to examine the impact of size on the relative performance of IMF and WAVG to the other combiner methods. For example, if the data size is very small, the extent of adjustment of weights in WAVG and redistribution of wealth in IMF are small. We therefore evaluate whether the relative combiner method performance is moderated by the data set size.

Data Set Positive Ratio. The positive ratio of the data set refers to the number of positive class objects divided by all objects in the data set. Positive ratio is included as a covariate to test whether the relative combiner method performance depends on the data set positive ratio. Theoretically, a performance difference, if any among combiner methods should be evident in data sets with positive ratios in the medium range, but not necessarily in data sets with very low (high) positive ratios where any trivial classifier that always predicts the object as negative (positive) does well. Thus, we only expect differences in performance among the combiner methods when the trivial rule is ineffective. The range of data set positive ratios over which the trivial rule is effective is, furthermore, impacted by the cost-to-benefit ratio level. The trivial rule that classifies everything as *positive* is effective over a wider range of data set positive ratios (i.e., medium and high rather than just high) when the cost-to-benefit ratio is low (Witten and Frank 2005). Conversely, the trivial rule that classifies everything as *negative* is effective over a smaller range of data set positive ratios (i.e., just extremely low rather than low) when the cost-to-benefit ratio is low. Considering that the median of the experimental cost-to-benefit manipulations is close to 1:5—i.e., in the low range—we expect combiner method performance differences for low to medium data set positive ratios, but not for medium to high data set positive ratios. We therefore evaluate whether the relative combiner method performance is moderated by the data set positive ratio.

Ensemble Diversity. Base-classifier diversity describes the degree to which the ensemble base classifiers differ in the errors they make. Diversity among the base classifiers is incorporated in the experiment by using different learning algorithms for each base-classifier. Diversity is measured using Yule's Q statistic (Yule 1900) for each data set (see the e-companion). By measuring diversity we can evaluate whether the relative performance of combiner methods is impacted by the level of complimentary information provided by the base classifiers in the different data sets.

4.2.4. Investigating the True Class of All Objects. To evaluate the external validity of our result to domains where the true object class is revealed for all objects, we perform an experiment where the performance of combiner methods is evaluated using both positive and negative classifications. In this experiment, we examine a version of WAVG where wealth is updated for both positive and negative classifications, as well as aWAVG. In aWAVG, the weights are determined based on AdaBoost: ln((1 − error rate)/error rate).

4.3. Time Lag, IMF Parameters, and Base-Classifier Cost-Benefit Retraining

4.3.1. Time Lag and Performance. In the main experiment the true class of t is given instantly after t is classified, but in reality it usually takes some time to determine the true class of t. To determine the performance impacts of such time lags, we perform an experiment where wealth w_{it} is not updated until v additional objects have been classified. v is manipulated at six different levels: 0%, 1%, 5%, 10%, 25%, and 50% of the size of the data set, for each of the 17 data sets, while the main experiment factors are held constant as follows: combiner method = IMF; number of agents = 10; and cost-to-benefit ratio = 1:10. Using these treatments we investigate if the net benefits from 0%-IMF (no time lag) and from 1%-IMF, 5%-IMF, 10%-IMF, 25%-IMF, and 50%-IMF are significantly different.

4.3.2. Selection of IMF Parameters

Binary Search Stopping Parameter ε. The tolerance value ε is used in binary search to determine when to stop the search. To gain a better understanding of how to select an appropriate value for ε and to investigate if this selection is domain dependent, we run an experiment where different values of ε are tested. For a given value of ε (manipulated at $0.01, 0.001, \ldots, 0.00000000001$), we run IMF on each of the data sets while holding other factors constant as follows: number of agents = 10, and cost-to-benefit ratio = 1:10. Also, if no interactions exist, we are still interested in investigating the direct impact of ε on NB.

Maximum Bet Multiplier k. To ensure that the ensemble is not completely dominated by a minority of better performing agents, while at the same time weighing the inputs of better performing agents more heavily, appropriate values of k must be used. For a given value of k (manipulated at 1, 2, 5, 10, 25, 50, 75, 100, 125, 150, 200, 250, 350, 500, and 1,000), we run IMF on all data sets with the number of agents factor set at 10 and the cost-to-benefit ratio set at 1:10. We are also interested in investigating interactions between k and data set characteristics to determine whether the choice of k is domain specific.

4.3.3. Base-Classifier Cost-Benefit Retraining. The ensemble base classifiers in the experiments are not trained using cost-sensitive learning, and they are not retrained for each cost-to-benefit treatment level. The ensemble results are likely to change if the base classifiers are retrained for different cost-to-benefit ratios. However, because all four combiner methods are tested using the same base classifiers, we do not believe that this will systematically bias the relative performance of the combiner methods. Nevertheless, we perform two experiments where the classification performances of various combiner methods are evaluated at five different cost-to-benefit ratios. The first experiment evaluates the combiner methods using an ensemble of five crisp base classifiers used in two different modes: cost-to-benefit ratio both retrained and not retrained. The retrained crisp base classifiers are obtained by hardening measurement level base classifiers at optimal thresholds for the different combinations of data sets and cost-benefit ratios. The base classifiers that are not retrained are obtained by hardening the same base classifiers using a threshold of 0.5. The second experiment evaluates the combiner methods using (a) an ensemble of 10 base classifiers, of which 5 are tree classifiers and (b) an ensemble of 5 base classifiers, in which all 5 are tree classifiers, where the classifiers in (a) and (b) are set in two different modes: cost-to-benefit ratio retrained and not retrained. The retrained base classifiers are obtained using CostSensitiveClassifier in Weka, where each base classifier is retrained at each combination of data set and cost-benefit ratio. In these experiments we evaluate the effect of the interaction between base-classifier mode and combiner method on combiner method performance to evaluate whether the relative performance of the combiner methods depends on whether the base classifiers are retrained.

4.3.4. Base-Classifier Wealth Convergence. To provide initial insight into how fast the relative wealth of individual agents and the total wealth of all the agents in the ensemble converge, we run experiments using select data sets, while holding the number of agents in the ensemble constant at 10 and the cost benefit ratio

Table 4 Statistical Analysis Data

	Combiner method × number of agents		Combiner method × cost-to-benefit ratio		Combiner method	
	Net benefit	ln(benefit)	Net benefit	ln(benefit)	Net benefit	ln(benefit)
Low	350	2.54	30	1.48	30	1.48
High	71,274	4.85	763,879	5.88	763,879	5.88
Mean	18,748	3.82	31,602	3.50	25,946	3.64
Standard deviation	24,471	0.68	96,569	1.00	74,323	0.89
Number of treatments	44		52		96	
N	748		884		1,632	

constant at 1:1. The percentage of the agents' wealth to the total wealth in the ensemble is then output after each classified transaction.

5. Results

5.1. Relative Combiner Method Performance

5.1.1. Overview. Table 4 provides an overview of the results data organized by the three data sets used in our experiments. Two of the statistical analysis data sets are based on the 5 × 11 (combiner method by number of agents) and the 5 × 13 (combiner method by cost-to-benefit) factorial designs, while the third is obtained by pooling the two statistical analysis data sets (possible as the interactions are not significant, as discussed in 5.1.3). We report two-tailed *p*-statistics throughout the paper. For significance testing we use an alpha of 0.05 and 0.1 for marginal significance. To retain an experimentwise error rate of 0.05, while balancing the risk of type II errors, we use a modified Bonferroni procedure (Jaccard and Wan 1996).

5.1.2. Combiner Method Main Effect. The combiner method main effect is tested using the model shown in (18) and the pooled result set described earlier. Note that for each combination of UCI data set ∗ number of agents and UCI data set ∗ cost-to-benefit ratio, the same four combiner methods are tested. We therefore block for the data set, number of agents, cost-to-benefit ratio, data set ∗ number of agents, and data set ∗ cost-to-benefit ratio effects:

$$\ln(\text{net benefit}) = \beta_0 + \beta_1 * \text{combiner method} + \text{block}. \qquad (18)$$

The combiner method main effect is significant ($p < 0.0001$), and the post hoc analysis shows that IMF significantly outperforms AVG ($p = 0.0042$), WAVG ($p = 0.0229$), and MAJ ($p < 0.0001$). See Table 5 for parameter estimates and standard errors.

5.1.3. Sensitivity Analysis. The sensitivity of the relative performance of the combiner methods to the number of agents in the ensemble and the cost-to-benefit ratio are respectively tested using the model shown in (19) for the 5 × 11 factorial design and model (20) for the 5 × 13 factorial design. Thus, each combiner method is tested for all combinations of UCI data set ∗ number of agents and UCI data set ∗ cost-to-benefit ratio in the respective models. We

Table 5 Summary of Primary Results

Effects	Results	Sig.[a]
Net benefit (only true class of objects classified as positive known)	Net $\text{benefit}_{\text{IMF}} >$ net $\text{benefit}_{\text{AVG}}$	$p = 0.0042$
	Net $\text{benefit}_{\text{IMF}} >$ net $\text{benefit}_{\text{WAVG}}$	$p = 0.0229$
	Net $\text{benefit}_{\text{IMF}} >$ net $\text{benefit}_{\text{MAJ}}$	$p < 0.0001$
Parameter estimates (standard errors)		
AVG		−0.00101 (0.00154)
IMF		0.00621 (0.00154)
MAJ		−0.00568 (0.00154)
Sensitivity analysis		
Number of agents	Not sensitive	$p = 0.1407$
Cost-to-benefit ratio	Not sensitive	$p = 0.7552$
Data set size	Not sensitive	$p = 0.7325$
Average agent accuracy	Not sensitive	$p = 0.9803$
Diversity ∗ method		
> without MAJ	Not sensitive	$p = 0.7813$
> without AVG/WAVG	Sensitive	$p = 0.0099$
Net benefit: low diversity	Net $\text{benefit}_{\text{IMF}} >$ net $\text{benefit}_{\text{MAJ}}$	$p = 0.0138$
Net benefit: high diversity	Net $\text{benefit}_{\text{IMF}} >$ net $\text{benefit}_{\text{MAJ}}$	$p < 0.0001$
Net benefit: low positive ratios	Net $\text{benefit}_{\text{IMF}} >$ net $\text{benefit}_{\text{AVG}}$	$p = 0.0005$
	Net $\text{benefit}_{\text{IMF}} >$ net $\text{benefit}_{\text{WAVG}}$	$p = 0.0021$
	Net $\text{benefit}_{\text{IMF}} >$ net $\text{benefit}_{\text{MAJ}}$	$p < 0.0001$
Net benefit: high positive ratios	Net $\text{benefit}_{\text{IMF}} >$ net $\text{benefit}_{\text{AVG}}$	$p = 0.3439$
	Net $\text{benefit}_{\text{IMF}} >$ net $\text{benefit}_{\text{WAVG}}$	$p = 0.6023$
	Net $\text{benefit}_{\text{IMF}} >$ net $\text{benefit}_{\text{MAJ}}$	$p = 0.0139$
Parameter estimates (standard errors)		
AVG		−0.00210 (0.00180)
IMF		0.00799 (0.00180)
MAJ		−0.00509 (0.00180)
Net benefit (true classes of all objects known)	Net $\text{benefit}_{\text{IMF}} >$ net $\text{benefit}_{\text{AVG}}$	$p = 0.1264$
	Net $\text{benefit}_{\text{IMF}} >$ net $\text{benefit}_{\text{WAVG}}$	$p = 0.0767$
	Net $\text{benefit}_{\text{IMF}} >$ net $\text{benefit}_{\text{MAJ}}$	$p = 0.0078$
	Net $\text{benefit}_{\text{IMF}} >$ net $\text{benefit}_{\text{AWAVG}}$	$p < 0.0001$
Sensitivity analysis		
Number of agents	Not sensitive	$p = 0.9848$
Cost-to-benefit ratio	Not sensitive	$p = 0.9944$
Data set size	Not sensitive	$p = 0.8293$
Positive ratio	Not sensitive	$p = 0.0847$
Average agent accuracy	Not sensitive	$p = 0.4030$
Ensemble diversity	Not sensitive	$p = 0.4599$

[a] All *p*-values are two-tailed.

therefore block for these interactions in the respective models:

$$\begin{aligned}\ln(\text{net benefit}) = \beta_0 &+ \beta_1\,\text{combiner method}\\ &+ \beta_2\,\text{number of agents}\\ &+ \beta_3\,\text{combiner method}\\ &* \text{number of agents} + \text{block},\end{aligned} \tag{19}$$

$$\begin{aligned}\ln(\text{net benefit}) = \beta_0 &+ \beta_1\,\text{combiner method}\\ &+ \beta_2\,\text{cost-to-benefit ratio}\\ &+ \beta_3\,\text{combiner method}\\ &* \text{cost-to-benefit ratio} + \text{block}.\end{aligned} \tag{20}$$

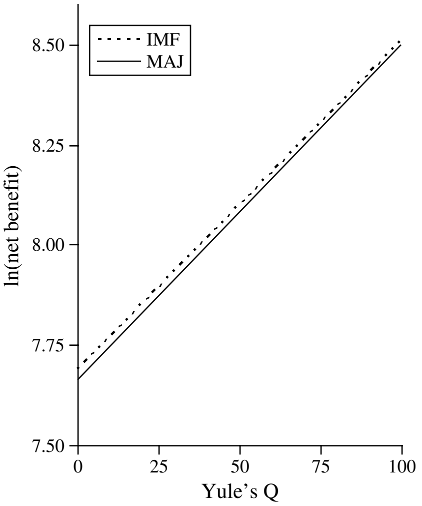
The combiner method $*$ number of agents ($p = 0.1407$) and combiner method $*$ cost-to-benefit ratio ($p = 0.7552$) interactions are insignificant. This indicates that the performance advantages of IMF over AVG, WAVG, and MAJ are not moderated by the number of agents in the ensemble or by the domain dependent cost-to-benefit ratio. Because prior research has not evaluated AVG, MAJ, or WAVG under different cost assumptions, we perform further inspections of the interaction results using scatter plots and find that the performance differences among the combiner methods are stable over the different cost-to-benefit ratios tested. Thus, the earlier conclusion based on the statistical results is corroborated.

The sensitivity of the combiner method performance result to data set average agent accuracy, size, positive ratio, and ensemble diversity are tested using the same blocking factor and result set used for (18):

$$\begin{aligned}&\ln(\text{net benefit})\\ &\quad= \beta_0 + \beta_1\,\text{combiner method}\\ &\qquad+ \beta_2\,\text{data set average agent accuracy}\\ &\qquad+ \beta_3\,\text{data set size} + \beta_4\,\text{data set positive ratio}\\ &\qquad+ \beta_5\,\text{ensemble diversity} + \beta_6\,\text{combiner method}\\ &\qquad* \text{data set average agent accuracy}\\ &\qquad+ \beta_7\,\text{combiner method} * \text{data set size}\\ &\qquad+ \beta_8\,\text{combiner method} * \text{data set positive ratio}\\ &\qquad+ \beta_9\,\text{combiner method} * \text{ensemble diversity}\\ &\qquad+ \text{block}.\end{aligned} \tag{21}$$

The results do not show that relative combiner method performance is sensitive to the data set size ($p = 0.7325$) or data set average agent accuracy ($p = 0.9803$). We do, however, find that the combiner method $*$ data set ensemble diversity ($p = 0.0342$) and combiner method $*$ data set positive ratio ($p < 0.0001$) interactions are significant.

Figure 5 **Combiner Method (MAJ and IMF) × Diversity Interaction**



The interaction involving diversity appears to be driven by MAJ, as MAJ has a significant parameter estimate for the interaction ($p = 0.0039$), whereas AVG ($p = 0.6363$) and IMF ($p = 0.3227$) are insignificant. This is verified by noting that the combiner method $*$ diversity interaction is insignificant ($p = 0.7813$) when MAJ is excluded from the analysis. When only including MAJ and IMF in the analysis, the interaction is significant ($p = 0.0099$). We therefore only perform a detailed analysis of IMF versus MAJ.

Based on visual comparison (Figure 5) it appears that IMF outperforms MAJ at all diversity levels; however, the performance difference is less at low diversity levels (high Yule's Q). However, even at low diversity levels ($Q > 75$), IMF outperforms MAJ ($p < 0.0138$). Thus, at all diversity levels IMF outperforms MAJ as per this test, and AVG and WAVG as per the insignificant interaction and significant main effect.

We explore the significant combiner method $*$ positive ratio interaction ($p = 0.0342$) by dividing the data sets into two groups based on the data set positive ratio, a high group with about half the data sets, positive ratio (>40%) and a low group with the remaining data sets (≤40%). In each group, a model with the combiner method factor and the blocking variables as in (18) are then tested. IMF significantly outperforms AVG ($p = 0.0005$), MAJ ($p < 0.0001$), and WAVG ($p = 0.0021$) in the low group. In the high group, IMF significantly outperforms MAJ ($p = 0.0139$), but the performance advantage is insignificant with respect to AVG ($p = 0.3439$) and WAVG ($p = 0.6023$).

5.1.4. Investigating the True Class of All Objects. The results obtained when the true classes of both positive and negative classifications are revealed are statistically equivalent to the results presented in §§5.1.2. and 5.1.3 with the following exceptions: (1) results are not sensitive to either diversity ($p = 0.4599$) or positive ratio ($p = 0.0847$); (2) IMF still significantly outperforms MAJ ($p = 0.0078$) and WAVG ($p = 0.0767$); and the performance advantage over AVG is now only marginal ($p = 0.1264$). Note that the p-values are two-tailed. The results also show that IMF outperforms aWAVG ($p < 0.0001$). Table 5 summarizes these results.

5.2. Time Lag, IMF Parameters, Cost-Benefit Retraining, and Wealth Convergence

The impact of time lag on net benefit is tested using the model shown in (22) and a statistical analysis data set derived from holding the number of agents and cost-to-benefit ratio constant at 10 and 1:10, respectively. Because all UCI data sets are used for all the treatments in the model (for a total of 102 observations), we block for the data set effect:

$$\ln(\text{net benefit}) = \beta_0 + \beta_1 v + \text{block}. \qquad (22)$$

The lag-level main effect ($p = 0.9962$) is insignificant, thereby indicating that time lag does not impact IMF performance. On the related issue of wealth convergence, readers are referred to the e-companion.

Using the model shown in (23), we do not find any evidence that the value of the binary search-stopping parameter ε, within the tested range $(0.01, 0.001, \ldots, 0.00000000001)$, impacts the performance of IMF ($p = 0.5071$). The impact of ε on the performance of IMF is also not domain dependent. More specifically, while blocking for the data set effect on net benefit, the $\varepsilon *$ data set positive ratio ($p = 0.1248$), $\varepsilon *$ data set size ($p = 0.1856$), $\varepsilon *$ data set average agent accuracy ($p = 0.5989$), and $\varepsilon *$ data set diversity ($p = 0.8897$) interactions are insignificant. Based on the results, in all experiments ε is set to the middle value tested (0.000001):

$$\begin{aligned}
&\ln(\text{net benefit})\\
&\quad = \beta_0 + \beta_1 \varepsilon + \beta_2 \,\text{data set average agent accuracy}\\
&\qquad + \beta_3 \,\text{data set positive ratio} + \beta_4 \,\text{data set size}\\
&\qquad + \beta_5 \,\text{data set diversity}\\
&\qquad + \beta_6 \,\varepsilon * \text{data set average agent accuracy}\\
&\qquad + \beta_7 \,\varepsilon * \text{data set positive ratio} + \beta_8 \,\varepsilon * \text{data set size}\\
&\qquad + \beta_9 \,\varepsilon * \text{data set diversity} + \text{block}. \qquad (23)
\end{aligned}$$

To choose appropriate values for the maximum bet multiplier k and to investigate if the choice of k is domain dependent, we use the model shown in (24), where the block factor is data set:

$$\begin{aligned}
&\ln(\text{net benefit})\\
&\quad = \beta_0 + \beta_1 k + \beta_2 \,\text{data set average agent accuracy}\\
&\qquad + \beta_3 \,\text{data set positive ratio} + \beta_4 \,\text{data set size}\\
&\qquad + \beta_5 \,\text{data set diversity}\\
&\qquad + \beta_6 \,k * \text{data set average agent accuracy}\\
&\qquad + \beta_7 \,k * \text{data set positive ratio} + \beta_8 \,k * \text{data set size}\\
&\qquad + \beta_9 \,k * \text{data set diversity} + \text{block}. \qquad (24)
\end{aligned}$$

The $k *$ data set average agent accuracy ($p = 0.0036$) and $k *$ data set diversity ($p = 0.0012$) interactions are significant, but the $k *$ data set positive ratio ($p = 0.1581$) and $k *$ data set size ($p = 0.1812$) interactions are insignificant. Scatter plots with trend lines and the raw data tables for the standardized log net benefit of the different data sets at the 15 different k values indicate that the significant interactions are driven by extreme values of k. For low k values net benefit decreases as the diversity decreases or the average agent accuracy increases, and vice versa for high k values. However, $k = 50$ consistently provides relatively good results, even when compared to extreme k values at their best performance levels. Furthermore, $k = 25$ and $k = 75$ also perform well. Results show that only when using $k = 25$, $k = 50$, and $k = 75$, the $k *$ data set average agent accuracy ($p = 0.5697$) and $k *$ data set diversity ($p = 0.9212$) interactions are no longer significant. Based on these results, we set $k = 50$ in all experiments.

Using the model shown in (25) and blocking for data set, cost-to-benefit ratio and data set $*$ cost-to-benefit ratio effects, we examine the base-classifier mode and combiner method interactions in the two base-classifier cost-benefit retraining experiments. The results show an insignificant ($p = 0.2037$) interaction when the retrained base-classifiers are obtained by hardening measurement-level classifiers at optimal threshold levels. Similarly, when the retrained base classifiers are obtained using Weka's CostSensitiveClassifier, the interaction is insignificant for the two cases; i.e., $p = 0.9414$ when five tree and five non-tree base classifiers are included in the ensemble, and $p = 0.3638$ when the ensemble consist of five tree base-classifiers. Thus, as expected, we do not find any evidence of relative combiner method performance being moderated by base-classifier mode:[7]

$$\begin{aligned}
\ln(\text{net benefit}) = {}& \beta_0 + \beta_1 \,\text{combiner method}\\
& + \beta_2 \,\text{base-classifier mode}\\
& + \beta_3 \,\text{combiner method}\\
& * \text{base-classifier mode} + \text{block}. \qquad (25)
\end{aligned}$$

[7] Nevertheless, the reader should be aware that some of the crisp base classifiers used in the experiments under extreme cost-benefit ratios could reduce the benefits of MCC. The effect of this potential problem is unknown, but based on the experiment just discussed it does not bias the relative performance results one way or the other.

6. Discussion

6.1. Combiner Method Performance

Based on the β coefficients and standard errors from the main effects test (Table 5), IMF on average provides a 0.72, 1.19, and 0.62 percentage[8] greater impact on net benefit than AVG, MAJ, and WAVG, respectively. Furthermore, compared to the average net benefit of the base classifiers across different ensembles, IMF on average provides a 10.08 percentage[9] greater impact on net benefit. These results are not sensitive to the number of agents in the ensemble, cost-to-benefit ratio, data set size, data set average agent accuracy, or ensemble diversity. However, we do find that the relationship between combiner method and net benefit is moderated by the data set positive ratio when we assume that the true class of objects would only be revealed for objects classified as positive. The results show that IMF outperforms the other combiner methods at low to medium positive ratios, and that there is no significant difference at medium to high positive ratio levels, as theoretically expected. Thus, IMF performs well at all positive ratio levels and outperforms the other combiner methods when it matters the most, i.e., in skewed data sets with low to medium positive ratios for which, given the tested cost-benefit ratios, trivial rules classifying all objects as either positive or negative are likely to be ineffective. For low to medium positive ratios, IMF on average has a 1.01%, 1.31%, and 0.80%[10] greater impact on net benefits than AVG, MAJ, and WAVG, respectively.

To put this into perspective, assume a fraud classification task where the average cost savings from a fraud detection are \$20,000, the average cost of investigation is \$500, the positive rate is 1% (a low positive rate), there are 40,000 transactions per year, and using IMF we accurately classify 50% of the positive instances and 98% of the negative instances. The benefit from this classification is \$4,000,000 (\$20,000 $*$ 200), the cost is \$496,000 ($200 * 500 + 792 * 500$), and the net-benefit is \$3,504,000. In this example, IMF provides an additional benefit per year of \$35,390, \$45,902, and \$24,878 over AVG, MAJ, and WAVG, respectively. However, note that IMF also consumes more resources, an average of 6.21 msec of CPU time,[11] to classify one object, compared to 5.07 msec for WAVG, 1.68 msec for AVG, and 1.69 msec for MAJ. The slightly longer CPU time consumed by IMF is minor given that in most settings IMF classifies on average 579,415 objects per hour of CPU time on an off-the-shelf PC.

IMF also outperforms MAJ, WAVG, aWAVG, and AVG (at a marginal level of significance) when true classes of objects classified as positive as well as negative are revealed. These results are robust within a wide range of cost-to-benefit ratios, number of agents in the ensemble, ensemble diversity, and data set size, positive ratio, and average base-classifier accuracy.

To understand why IMF has superior performance to the other combiner methods, we need to understand the workings of IMF. Because of the log utility function, IMF should perform on par with AVG if all the agents have the same amount of funds available for placing bets in the market, i.e., if the aggregation is not wealth weighted (Wolfers and Zitzewitz 2006). However, as more accurate agents become wealthier, these agents end up influencing the market prices to a greater degree than the less accurate agents, and because the equilibrium prices represent the aggregated probabilities of the ensemble, the more accurate agents have a greater impact on the ensemble's decision than the less accurate agents. Thus, the ensemble decision in IMF is a performance-weighted average, which explains why there is a difference between IMF and AVG and also perhaps why IMF outperforms AVG, and to some extent MAJ, because MAJ is also a nonweighted combiner method with performance similar to AVG.

When comparing IMF to WAVG, we need to examine three major differences between IMF and WAVG: (1) WAVG assigns weights solely based on the precision of the base classifiers relative to the precision of the other base classifiers. In contrast, IMF places progressively greater weight on better performing agents' decisions, as agents with wealth above what they are allowed to bet hedge their bets to a lesser degree than other agents. (2) In IMF, weights are adjusted based on the degree of agents' performance as opposed to WAVG, where the weights are adjusted solely based on the ratio of an individual classifier's precision to the total precision of all the classifiers in the ensemble. To clarify, in IMF, an agent's wealth increases (decreases) to a greater degree the more (less) accurate the agent is in each bet, as agent bets

[8] Based on estimation using β coefficients and standard errors of -0.00101 and 0.00154 for AVG, 0.00621 and 0.00154 for IMF, and -0.00568 and 0.00154 for MAJ (Kennedy 1981).

[9] Based on estimations using β coefficients and standard errors of -0.105823 and 0.027584 for the average base classifier (Kennedy 1981) with ($p < 0.0001$).

[10] Based on estimations using β coefficients and standard errors of -0.002053 and 0.001751 for AVG, 0.0079922 and 0.001751 for IMF, and -0.005092 and 0.001751 for MAJ (Kennedy 1981).

[11] We used GetProcessTimes from Kernel32.lib, which measures CPU time used, rather than actual time to run the algorithms. CPU time excludes time that the process is waiting for other processes to complete. The resource consumption experiment is performed on computers ranging from a desktop computer with a Pentium 4 2.0 GHz processor with 256 MB of RAM to a personal laptop with an AMD Turion 64 X2 Mobile Technology TL-56 processor and 2,048 MB of RAM. The computer is held constant within each treatment group.

are increasing in agent probability estimates. Thus, agents who are correct and more certain receive a higher payout than agents who are correct but less certain, because the bets of more certain agents are higher, and vice versa. (3) The weights in IMF, but not in WAVG, are adjusted based on agents' relative contribution to the ensemble diversity. In IMF, agents with correct bets receive a greater payout if the odds are higher for that class, which occurs when the bets are higher for the other class.

6.2. Time Lag, IMF Parameters, and Base-Classifier Cost-Benefit Retraining

The results do not show that time lag between object classification and object determination impacts the performance of IMF within the range tested (0%–50% of the records in the data set). Thus, there is no evidence to suggest that the performance of IMF deteriorates with time lags between object classifications and object true class determination. The results also do not indicate that the binary search stopping parameter ε and maximum bet parameter k should be set to different values for different classification domains. In our experiments ε was held constant at 0.000001 and k was held constant at 50. We also do not find any evidence that there is a systematic bias in the relative performance of the combiner methods from not retraining the base classifiers for the different cost-to-benefit ratios.

6.3. Combiner Method Design Considerations

For multiagent system MCC implementations, IMF handles changes in ensemble composition and base-classifier performance. The market mechanism used in IMF functions independently of any specific agents that participate in the market. Furthermore, changes in an agent's relative performance impact the agent's wealth and therefore also the weight given to the agent's decisions in the decision fusion process, resulting in online learning. IMF also provides incentives for market participants to truthfully provide their private decisions. This is especially useful in multiagent systems based on competitive agents (Ygge and Akkermans 1999).

7. Conclusions and Future Research Directions

In this paper, we present IMF, a new and novel combiner method based on information markets for multiclassifier combination. We show through extensive experimentation that IMF provides additional utility compared to three benchmark combiner methods AVG, WAVG, and MAJ.

For future research, the effectiveness of IMF can be compared to other combiner methods in other multiclassifier combination architectures, such as bagging and boosting. Other research extensions include investigating the performance impacts of other types of agent behavior using utility functions such as constant absolute risk aversion, constant relative risk aversion, etc; modeling agents to update their beliefs based on market signals or ensemble consensus; mixing agents with different utility functions; and using a combination of human and software agent experts. IMF can also be extended for the more general k-class classification problem using the parimutuel betting mechanism. Finally, future research can explore the possibility of integrating the cost-benefit ratio into IMF itself.

8. Electronic Companion

An electronic companion to this paper is available as part of the online version that can be found at http://mansci.journal.informs.org/.

Acknowledgments

The authors thank the three anonymous reviewers and the associate editor for constructive suggestions that have resulted in substantial improvements to the experimental procedure and theoretical model. This paper has also benefited from the comments of reviewers and participants at the 2007 Winter Business Intelligence Conference in Salt Lake City, International Conference on Information Systems (ICIS) 2006 in Milwaukee, College of Business Research Workshop at the University of South Florida 2006, and Workshop on Information Technology Systems (WITS) 2005 in Las Vegas. This research has been partially supported by the Bear Stearns IT Research Center at the University of South Florida.

References

Berg, J. E., T. A. Rietz. 2003. Prediction markets as decision support systems. *Inform. Systems Frontiers* **5**(1) 79–93.

Carlsson, P., F. Ygge, A. Andersson. 2001. Extending equilibrium markets. *IEEE Intelligent Systems* **16**(4) 18–26.

Chan, P. K., W. Fan, A. L. Prodromidis, S. J. Stolfo. 1999. Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems Their Appl.* **14**(6) 67–74.

Drummond, C., R. C. Holte. 2006. Cost curves: An improved method for visualizing classifier performance. *Machine Learn.* **65**(1) 95–130.

Duin, P. W. R., M. J. D. Tax. 2000. Experiments with classifier combining rules. *Multiple Classifier Systems, Lecture Notes in Computer Science,* Vol. 1857. Springer, Berlin/Heidelberg, 16–29.

Fama, E. 1970. Efficient capital markets: A review of theory and empirical work. *J. Finance* **25**(2) 383–417.

Fan, J., S. Stolfo, J. Zhang. 1999. The application of AdaBoost for distributed, scalable and on-line learning. *Proc. ACM SIGKDD 5th Internat. Conf. Knowledge Discovery and Data Mining*, ACM, New York, 362–366.

Hanson, R. 2003. Combinatorial information market design. *Inform. Systems Frontiers* **5**(1) 107–119.

Hayek, F. A. 1945. The use of knowledge in society. *Amer. Econom. Rev.* **35**(4) 519–530.

Jaccard, J., C. K. Wan. 1996. *LISREL Approaches to Interaction Effects in Multiple Regression*. Sage Publications, Thousand Oaks, CA.

Jain, A. K., R. P. W. Duin, J. Mao. 2000. Statistical pattern recognition: A review. *IEEE Trans. Pattern Anal. Machine Intelligence* **22**(1) 4–37.

Kelly, J. 1956. A new interpretation of information rate. *IEEE Trans. Inform. Theory* **2**(3) 185–189.

Kennedy, P. E. 1981. Estimation with correctly interpreted dummy variables in semilogarithmic equations. *Amer. Econom. Rev.* **71**(4) 801.

Kittler, J., M. Hatef, R. P. W. Duin, J. Matas. 1998. On combining classifiers. *IEEE Trans. Pattern Anal. Machine Intelligence* **20**(3) 226–239.

Lam, L. 2000. Classifier combinations: Implementations and theoretical issues. *Multiple Classifier Systems, Lecture Notes in Computer Science,* Vol. 1857. Springer, Berlin/Heidelberg, 77–86.

Lee, W., S. J. Stolfo, K. W. Mok. 2000. Adaptive intrusion detection: A data mining approach. *Artificial Intelligence Rev.* **14**(6) 533–567.

Lin, J., M. Hwang, J. Becker. 2003. A fuzzy neural network for assessing the risk of fraudulent financial reporting. *Managerial Auditing J.* **18**(8) 657–665.

Newman, D. J., S. Hettich, C. L. Blake, C. J. Merz. 1998. UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html.

Nissen, M. E., K. Sengupta. 2006. Incorporating software agents into supply chains: Experimental investigation with a procurement task. *MIS Quart.* **30**(1) 145–166.

Pennock, M. D. 2004. A dynamic pari-mutuel market for hedging, wagering, and information aggregation. *Proc. 5th ACM Conf. E-Commerce*, ACM, New York.

Plott, C. R., J. Wit, W. C. Yang. 2003. Parimutuel betting markets as information aggregation devices: Experimental results. *Econom. Theory* **22**(2) 311–351.

Provost, F., T. Fawcett. 2001. Robust classification for imprecise environments. *Machine Learn.* **42**(3) 203–231.

Provost, F., T. Fawcett, R. Kohavi. 1998. The case against accuracy estimation for comparing induction algorithms. *Proc. 15th Internat. Conf. Machine Learn.*, Morgan Kaufmann, San Francisco, 445–453.

Rubinstein, M. 1976. The strong case for the generalized logarithmic utility model as the premier model of financial markets. *J. Finance* **31**(2) 551–571.

Saar-Tsechansky, M., F. Provost. 2004. Active sampling for class probability estimation and ranking. *Machine Learn.* **54**(2) 153–178.

Stolfo, S., A. L. Prodromidis, S. Tselepis, W. Lee, D. W. Fan, P. K. Chan. 1997. JAM: Java agents for meta-learning over distributed databases. *Proc. 3rd Internat. Conf. Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, CA, 74–81.

Suen, C. Y., L. Lam. 2000. Multiple classifier combination methodologies for different output levels. *Multiple Classifier Systems, Lecture Notes in Computer Science,* Vol. 1857. Springer, Berlin/Heidelberg, 52–66.

Witten, I. H., E. Frank. 2005. *Data Mining*: *Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco.

Wolfers, J., E. Zitzewitz. 2006. Interpreting prediction market prices as probabilities. *Proc. Allied Soc. Sci. Assoc. Annual Meeting, Boston,* January 6–8.

Ygge, F., J. M. Akkermans. 1999. Decentralized markets versus central control: A comparative study. *J. Artificial Intelligence Res.* **11** 301–333.

Yule, G. U. 1900. On the association of attributes in statistics: With illustrations from the material of the childhood society, etc. *Philos. Trans. Roy. Soc. London. Ser. A, Containing Papers Math. Physical Character* **194** 257–319.

Zheng, Z., B. Padmanabhan. 2007. Constructing ensembles from data envelopment analysis. *INFORMS J. Comput.* **19**(4) 486–496.