In this paper we present Information Market based Fusion (IMF), a novel multi-classifier combiner method for decision fusion that is based on information markets. We compare the effectiveness of IMF to Majority, Average and Weighted Average schemes using computational experiments involving 17 datasets from the UCI Machine Learning Repository and 22 different base-classifiers from Weka. Collectively, the experimental results show that IMF outperforms the other combiner methods. IMF furthermore adjusts to changes in base-classifier accuracy, provides incentives for the base-classifiers to present truthful information, and does not require training or a static ensemble composition. Additionally, in this paper we also introduce and test a novel dynamic cutoff selection algorithm.

Key words: multi-classifier combination, decision fusion, information markets, agents

1. Introduction

In many decision-making scenarios, decisions of multiple human experts or multiple classifiers are fused to determine the overall decision. Examples include: a group of investment experts making investment decisions for an organization; an ensemble of classifiers in a fraud detection application making decisions on whether or not a transaction is fraudulent; a group of accounting experts and classifiers making going-concern decisions. Multi-classifier 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 an object’s class label.

The basic idea behind MCC is that different classifiers in an ensemble have different strengths and weaknesses, and therefore provide complementary information 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, and can be categorized based on whether they require training or not. For example, Naive
Bayes, Decision Templates and Weighted Average (WAVG) are combiner methods requiring training
while Average (AVG), Majority (MAJ) and Product do not require any training. Details on various
combiner methods in the literature can be found in (Suen and Lam 2000). Results from the literature
indicate that MCC overall provides performance benefits, and that MAJ and AVG perform either at a
similar level or significantly better than trained methods (Kittler, et al. 1998). Existing combiner methods
that require training however assume that ensemble base-classifier composition remains constant and that
training performance is a good proxy for subsequent actual performance.

The development of software agent technology offers a new framework and infrastructure to support
decision-making (Nissen and Sengupta 2006) where human experts driving software agents as well as
autonomous software agents embodying classifiers and other intelligent algorithms can leverage their
individual strengths to make collective decisions. In such decision-making endeavors it is important that
humans as well as software agent experts are provided incentives to truthfully provide their actual
classification decisions. Existing combiner methods were not designed with agents-based decision-
making platforms in mind and therefore do not provide this feature.

**Figure 1:** Generic Classifier Combiner Architecture
In this paper we propose an information market based fusion approach for multi-classifier combination that 1) overall outperforms MAJ, AVG, and WAVG, 2) does not require training, 3) can adapt to changes in ensemble composition and base-classifier accuracy, and 4) does not assume that the base-classifiers are cooperative agents. In evaluating the effectiveness of our proposed approach, we compared IMF against three combiner methods, AVG, MAJ and WAVG, that have performed relatively well in prior research. The computational experiments involved 17 datasets that were obtained from the UCI Machine Learning Repository (Newman et al. 1998) and 22 different base-classifiers from Weka (Witten and Frank 2005). Although our information market based combiner method could also be used for fusing decisions by human experts, in this paper, we focus on multi-classifier decision fusion. Along with IMF, we also propose and evaluate a dynamic cutoff selection algorithm that can be used in real-world applications where the true object class is only determined for objects that are classified as positive.

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

2. Related Research

2.1. Decision Fusion Overview

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 tested in order to identify the best classifier for a given problem domain, but it is generally difficult to determine which classifier(s) will perform well in subsequent classification. Classifier comparisons have revealed that although a specific classifier may provide an overall relative performance advantage for a given
classification problem, the classification errors it makes for certain cases may be avoided by an “inferior”
classifier (Kittler, et al. 1998). Thus, by combining the decisions of diverse classifiers it is possible to
improve the overall prediction accuracy and to avoid having to select a single classifier. This is the basis
for decision fusion.

MCC research has primarily focused on two areas: (1) what classifiers to include in an ensemble and
how to train these classifiers; and (2) how to combine base-classifier decisions, the focus of this paper.
Methods such as bagging, boosting and stacking fall into the first category, see (Witten and Frank 2005)
for details on these methods, while combiner methods such as MAJ, AVG and WAVG fall into the
second category. Bagging and boosting can use any combiner method, but normally use MAJ or AVG
depENDING on the base-classifier output. More recent research within the former stream has for example
used Data Envelopment Analysis to select efficient classifiers and then combine these classifiers’
decisions (Zheng and Padmanabhan forthcoming), and Zhao and Ram (2004) investigated how to select
appropriate cascading depth in cascade generalization, a variation of method stacking. As stated earlier,
our research does not focus on classifier selection and training, instead we focus on the combiner method.
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
typically outperform methods that do not require training (Suen and Lam 2000); and simple combiner
methods MAJ and AVG perform either at the same level or significantly better than more complex trained
methods (Kittler, et al. 1998).

2.2. Combiner Method Design Considerations

Another important, but largely overlooked, aspect of combiner methods is how well they fit with different
system architectures. MCC problems are distributed in nature, but existing MCC designs view base-
classifiers and combiner-methods as systems with centralized control. On the other hand, distributed
problem solving (DPS) is of primary focus in multi-agent system (MAS) research, and one of the main
benefits of MAS architectures is the support they provide for DPS. Given the increasing popularity of
MAS for various applications (Nissen and Sengupta 2006) and the support MAS provides for DPS, MAS is a suitable architecture for MCC where the base-classifiers, combiner method and providers of object features can be implemented as agents.

Existing combiner methods however assume that the agents, implementing classifiers, are cooperative. A variety of successful MAS systems are based on competitive agents, for example routing of delivery trucks (Bohte and Poutré 2006) and climate control in large buildings (Ygge and Akkermans 1999), while other systems are in the research stage, for example dynamic scheduling of patient treatment in hospitals (Bohte and Poutré 2006). In such systems, the competitive nature of the agents allows groups with different goals to implement agents and the competition provides a mechanism for efficient utilization of resources. Given that existing combiner methods do not provide any incentives for the agents to provide their private information truthfully, it is not clear how existing combiner methods would get accurate information from the base-classifiers embodied in agents or human experts driving agents.

Furthermore, in MAS implemented in dynamic real world domains, the ensemble composition could change over time as agents (embodying base-classifiers or representing human experts) are retired, added or temporarily not available. Additionally, as the availability and the cost of gathering different object features, and the usefulness of these features change overtime, the relative accuracy of the base-classifiers could also change. Existing trained combiner methods, as opposed to methods that did not require training, were not designed with this in mind as they assumed that the ensemble composition was static, and that individual classifier performance did not change subsequent to training and validation. We next introduce IMF, a combiner method that overcomes the problems discussed in this section and at the same time outperforms AVG, MAJ and WAVG.

3. Information Market Based Fusion

The combiner method proposed in this paper is theoretically grounded in information markets. More specifically, IMF’s aggregation mechanism is based on pari-mutuel betting markets. We next provide an overview of information markets and pari-mutuel betting, followed by a detailed description of IMF.
3.1. Information Markets

Information markets (IM) are markets designed specifically 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 argued in the 1940s that prices in a competitive market efficiently aggregate information held by market participants (Hayek 1945). The efficient market hypothesis states that all private and public information is reflected in equilibrium prices, this even includes insider information under strong efficiency (Fama 1970). These concepts can even be traced back to Adam Smith’s invisible hand.

More recently, the idea of using markets for the specific purpose of aggregating information has received more attention. Experimental information market research has generally found support for the efficient market hypothesis (Forsythe and Lundholm 1990). Empirical research has found evidence of information aggregation in information markets (Berg and Rietz 2003), and research focusing on pari-mutuel betting has found support for the efficient market hypothesis (Ali 1979). Based on these theoretical and empirical findings we believe that an information market can be designed to effectively combine the decisions of base-classifiers in an ensemble.

3.2. Pari-Mutuel Betting

Pari-mutuel betting originated in horserace gambling in France in 1865 and has since become a very popular betting mechanism in the horseracing world. Pari-mutuel means “wager mutual” and comes from the fact that in pari-mutuel betting, a winning wager (i.e., bet) receives a share of the total wagers (winning and loosing bets less track commission) as a proportion of this winning wager to all winning wagers. In a given pari-mutuel betting horserace, bettors place bets on different race outcomes and are paid based on their wager, the other bettors’ wagers and the actual outcome of the race as described above. The final track odd for a given horse is the total amount bet on all horses in the race divided by
the total amount bet on the given horse i.e., the payout equals the amount bet on the winning horse times
the odds for the winning horse, less track commission. From a MCC perspective these odds are of great
importance as they represent aggregated information where the reciprocal of the odd for a specific horse
can be viewed as the probability estimate of that horse winning the race.

Plott, et al. (2003) experimentally examined information aggregation and different betting behaviors
in pari-mutuel betting markets using two private information models, Decision Theory Private
Information (DTPI) and Competitive Equilibrium Private Information (CEPI), and one model with
updated beliefs, Competitive Equilibrium Rational Expectations (CERE). Plott, et al. (2003) found that
DTPI, followed by CEPI overall best described the behavior of participants in a pari-mutuel market as
observed in a laboratory setting.

In DTPI, agents only consider their own private information and ignore market prices when deciding
on their bets as well as in forming beliefs. In CEPI, agents also do update their beliefs based on market
prices, but do base their bets on the current market price. In both DTPI and CEPI models, agents
maximize their conditional expected utility given their private probability estimates and available funds
(and market prices in CEPI, i.e., odds), subject to constraints. The total bets and corresponding odds for
each event determine the potential payout for each event. At equilibrium, all the potential payouts are
equal to the total amount bet on all events, assuming no track take. In CERE, agents again take the
market prices as given, but in CERE these prices incorporate all available private and public information,
and an agent’s behavior is determined by both their prior experience and current information. Thus,
agents include current market prices in addition to their private information in updating their beliefs.

In equilibrium, agents are indifferent to which outcome to bet on as their conditional expected utility
is the same for all outcomes. This also implies that in equilibrium, all agents have incorporated all
relevant information into their beliefs. In both CEPI and CERE, the prices are assumed to be in
equilibrium; however, as each betting round starts without any prices defined, the equilibrium must be
obtained before the agents can place their bets.
3.3. Information Market Based Fusion

IMF is a multi-classifier combiner method based on pari-mutuel 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 an object \( t \) (i.e., a transaction) can be classified as fraudulent \((j=1)\) or non-fraudulent \((j=2)\) by an ensemble \( E \) of agent classifiers, with the index set of classes \( J=\{1,2\} \). The ensemble \( E \) has \( m \) agents embodying different base-classifiers (and henceforth simply referred to as agents) represented by indices \( i \) in the index set \( D = \{1,..,m\} \). While determining the class membership of \( t \), agent \( i \in D \) uses the feature vector associated with \( t \) to determine the posterior probability estimate \( p_{ij} \) in \([0, 1]\) that \( t \) belongs to class \( j \in J \). Agent \( i \) bets \( q_{ij} \) that object \( t \) belongs to class \( j \) and is paid based on \( q_{ij} \), the total bets placed by all the agents on class \( j \), \( Q_{ij} = \sum_{i \in D} q_{ij} \), the total bets placed by all the agents on all class, \( Q_t = \sum_{j \in J} \sum_{i \in D} q_{ij} \), and the true class of object \( t \), according to pari-mutuel mechanism.

Ensemble \( E \)'s overall probability estimate that \( t \) belongs to \( j \in J \) is given by \( 1/O_{ij} \) in \([0,1]\), where \( O_{ij} \) is the odd that \( t \) belongs to \( j \in J \). \( O_{ij} \), which is equal to \( Q_t/Q_{ij} \), is in equilibrium when the potential payouts \( Q_{ij}O_{ij} \) for each \( j \in J \) and the total amount bet on all events \( Q_t \) are equal, assuming house commission to be zero. At this point, \( q_{ij}O_{ij} \) is also equal to the potential payout to agent \( i \) for event \( j \) computed using the pari-mutuel mechanism as \( (q_{ij}/Q_t)Q_t \).

The ensemble’s overall probability estimate \( 1/O_{ij} \) is compared to a threshold (cut-off) value \( C_j \) to determine if \( t \) should be classified as belonging to class \( j \). Establishing equilibrium odds is however a nontrivial task, given that there is dependency between \( Q_{ij} \) and \( O_{ij} \). Therefore multiple rounds of betting are required to reach equilibrium (if it exists) or near equilibrium, and to determine final odds (equilibrium or near equilibrium odds) that can then be used by agents to make their actual bets. In each of the rounds prior to reaching the final odds, odds are first updated based on various agents’ prior bets and then agents place new bets based on the current updated odds.
IMF, in addition to providing a means to establish odds, also facilitates the redistribution of wealth among various agents based on agents’ bets and winnings. From a MCC perspective, IMF combines decisions based on a wealth-weighted probability of event $j$ occurring. We next describe the IMF method in detail as per the major steps depicted in Figure 2.

### 3.3.1. Determining Final Odds

Recall that final odds are equilibrium odds, if equilibrium exits. Otherwise, final odds represent near-equilibrium odds. Equilibrium odds are found for each object $t$ by iteratively updating the odds, and agents placing bets using these odds until the odds provided to the agents and their subsequent bets result in $Q_t O_j = Q_t$, at which time the market closes. The following observations are made. First, bets placed in betting rounds, when the market is in disequilibrium, are only used for the purpose of updating the odds. Second, if agent bets are discontinuous over $O_j$ then the existence of equilibrium odds cannot be guaranteed (Carlsson, et al. 2001). To overcome oscillation of odds over multiple iterations, and to find a good solution when no optimal solution exists, we use a combination of binary search and optimization.

Note that even if the equilibrium odds are not successfully established, the impact on the ensemble classification should be minimal considering that the final ensemble probabilities are within a narrow bound $\varepsilon$ of the equilibrium probability or the best probability (when equilibrium does not exist) (see below).

**Binary Search** - The binary search algorithm (see Figure 3) starts with setting lower bound $P^l=0$ and upper bound $P^u=1$ for the probability that object $t$ belongs to class $j=1$. $O_{i1}$ is then computed using (1), note that $O_{i2} = O_{i1}/(O_{i1}-1)$ in the case of two class problems:

![IMF Flowchart](image-url)
Set search space bounds, \( P^l = 0 \) and \( P^u = 1 \)

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 \( P^u \) and \( P^l \) to \( 1/O_{t1} \)

Take agent bets

Until \((P^u - P^l) \leq \varepsilon\)

\[ O_{t1} = \frac{2}{P^l + P^u} \]  \( \quad (1) \)

The agents then place bets by solving problem P2 in Section 3.3.3. The odds and bets are then used to evaluate if the current market odds are too high or too low as follows. If the potential payout for \( j = 1 \) is greater than the total bets \( Q_t \), then the odds \( O_{t1} \) is too high and the search space is therefore bounded below by the reciprocal of \( O_{t1} \) as in (2).

\[ P^l = 1/O_{t1} \]  \( \quad (2) \)

On the other hand, if the potential payout for \( j = 1 \) is less than the total bets \( Q_t \), then the odds \( O_{t1} \) is too low and the search space is therefore bounded above by the reciprocal of \( O_{t1} \) as in (3).

\[ P^u = 1/O_{t1} \]  \( \quad (3) \)

If the potential payouts for \( j = 1 \), \( Q_{t1}O_{t1} \) is the same as the total bets \( Q_t \), then the potential payouts for \( j = 1 \) is equal the potential payouts for \( j = 2 \), i.e., \( Q_{t1}O_{t1} = Q_{t2}O_{t2} \), the odds are in equilibrium and the search space is limited to this single value given by (4).

\[ P^l = P^u = 1/O_{t1} \]  \( \quad (4) \)

\( O_{t1} \) 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 continues iteratively until the search space is deemed narrow enough \((P^u - P^l) \leq \varepsilon\), where \( \varepsilon \) is a tolerance value.

\[ Z1 = \min_{O_{t1,M_j}} \sum_{j \in J} M_j \]

S.T.

- \( Q_{t1}O_{t1} - M_1 = Q_t \)
- \( Q_{t2}O_{t2} - M_2 = Q_t \)
- \( M_j \geq 0 \)
- \( 1/P^l \geq O_{t1} \geq 1/P^u \)
- \( O_{t2} = O_{t1}/(O_{t1} - 1) \)

**Figure 4:** P1 Optimization
are found by binary search, then $P^l = P^u$ and problem P1 need not be solved.

Problem P1 is used to determine the optimal odds within the bounds $P^l$ and $P^u$ obtained from binary search. The objective function minimizes dummy variables $M_1$ and $M_2$, which represent differences between the total amount bet and the total payout. At perfect equilibrium, $M_1$ and $M_2$ are equal to zero.

### 3.3.2. Classifying the Object

The cutoff and the reciprocal of final odds both have the same dimensions as that of probability and can be compared to classify an object according to the decision rule in (5).

$$\text{If } (1/O_f^t \geq C_1) \text{ then classify class of } t \text{ as } j=1; \text{ else classify class of } t \text{ as } j=2$$

As seen from (5), if the reciprocal of final odds for $j=1$ is higher than the cutoff then object $t$ is classified as being a member of the positive class, i.e., $j=1$. If an object is classified as a member of the positive class then the agents place their final bets (see Section 3.3.3) and agent $i$’s wealth is decreased by the amount of final bets:

$$w_{it} = w_{it} - \sum_{j \in J} q_{ij} \quad (6)$$

The true class of $t$ is then investigated and after the true class of $t$ has been determined agent $i$’s wealth is updated with any potential winnings, see Section 3.3.4 below.

Note that if the object is classified as a member of the negative class i.e., $j=2$, then the classification of the object is not pursued further, since investigations in the real world are typically not carried out to determine the actual class of an object when that object is classified as negative. In this case agent wealth is not updated with bets or winnings.

### 3.3.3. Determining Agent Bets

Agent $i$ solves the expected utility maximization problem P2 (see below) to determine the amount to bet $q_{ij}$ on classes $j=1,2$ and $s_{it}$, the amount not to bet that the agent could have bet, given the current market odds $O_{ij}$, the agent’s probability estimates $p_{ij}$ of object $t$ being in class $j=1,2$ (recall that agent $i$ embodying a classifier uses the feature vector of object $t$ to generate $p_{ij}$), the agent’s current wealth $w_{it}$...
plus the periodic endowment $m$, and multiplier $k$ that determines the house enforced maximum bet $km$.

The periodic endowment $m$ is given to all agents in order to prevent agents from running out of funds.

Given the utility function of agent $i$, $U_i$ as a function of wealth, problem P2 can be stated as follows,

\[
P2: \quad Z_2 = \max_{q_{itj}, s_{it}} \ p_{itj} U_i(w_{it} + m - q_{it1} - q_{it2} + q_{it1}O_{it1}) + p_{it2} U_i(w_{it} + m - q_{it1} - q_{it2} + q_{it2}O_{it2}) \tag{7}
\]

\[
s.t. \quad q_{it1} + q_{it2} + s_{it} = \begin{cases} w_{it} + m & \text{if } (w_{it} + m) \leq km \\ km & \text{if } (w_{it} + m) > km \end{cases} \tag{8}
\]

\[
q_{itj}, s_{it} \geq 0 \tag{9}
\]

The objective function in P2 represents the expected utility of agent $i$ when it bets $q_{itj}$ on event $j$ and does not bet $s_{it}$ on any event. The total amount of bets and non-bets cannot exceed the lower of the agents’ available funds $m + w_{it}$, and the house enforced maximum bet $km$ as indicated by (8). Further, bets $q_{itj}$ must be equal to or greater than zero (9). The house enforced maximum bet $km$ limits the amount of influence of the best performing agents in the ensemble, since in MCC, all agents, not only the best performing agents, need to contribute in order to ensure success of the ensemble (Kittler, et al. 1998), see Section 4.3.3 for further discussion.

P2 is general enough to enable the use of any utility function to model the agents’ risk aversion. We utilize a natural logarithm ($\ln$) utility function (hence forth simply referred to as log utility) that is 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) use of log utility enables the aggregated equilibrium supply and demand to reveal the wealth weighted mean beliefs (Wolfers and Zitzewitz 2006); (3) log utility is twice-differentiable and non-decreasing concave, leading to a decreasing absolute risk aversion (i.e., diminishing utility of wealth) without requiring the proportional risk aversion to be decreasing, constant or increasing (Rubinstein 1976); and (4) depending on which betting constraints is binding (see Lemma 1 and 2 below) log utility bets are either increasing in $p_{itj}$ and $w_{it} + m$ but not a
function of $O_{ij}$, a betting behavior corresponding to DTPI (Plott, et al. 2003), or increasing in $p_{ij}, w_i + m$
and $O_{ij}$, a betting behavior corresponding to CEPI (Plott, et al. 2003).

Using the log utility and changing the constraint so that the agents must bet everything that they can,
i.e., $s_i = 0$, but can hedge their bets by betting on both $j=1$ and $j=2$, P2 is transformed to P3 as follows:

P3: $Z_3 = \max_{q_{itj}} p_{itj} \ln(w_{it} + m - q_{it1} - q_{it2} + q_{it1}O_{1it}) + p_{it2} \ln(w_{it} + m - q_{it1} - q_{it2} + q_{it2}O_{2it})$ (10)

s.t. $q_{it1} + q_{it2} = w_{it} + m$ if $w_{it} \leq (k-1)m$
$q_{it2} = \begin{cases} w_{it} + m & \text{if } w_{it} \leq (k-1)m \\ km & \text{if } w_{it} > (k-1)m \end{cases}$ (11)

$q_{ij} \geq 0$ (12)

If $w_{it} \leq (k-1)m$ then $q_{it1} + q_{it2} = w_{it} + m$ and $w_{it} + m - q_{it1} - q_{it2} = 0$, leading to P4.

P4: $Z_4 = \max_{q_{itj}} p_{itj} \ln(q_{it1}O_{1it}) + p_{it2} \ln(q_{it2}O_{2it})$ (13)

s.t. $q_{it1} + q_{it2} = w_{it} + m$ (14)

$q_{ij} \geq 0$ (15)

**Lemma 1**: The optimal bets of agent $i$ in P4 while classifying $t$ is: $q_{it}^* = p_{ij}(w_{it}+m) \forall j \in J$.

Proof: See on-line supplement.

If $w_{it} > (k-1)m$ 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 P3 can be transformed to P5.

P5: $Z_5 = \max_{q_{itj}} p_{itj} \ln(a_{it} + q_{it1}O_{1it}) + p_{it2} \ln(a_{it} + q_{it2}O_{2it})$ (16)

s.t. $q_{it1} + q_{it2} = km$ (17)

$q_{ij} \geq 0$ (18)

**Lemma 2**: The optimal bets of agent $i$ in P5 while classifying $t$ is:

$q_{it1}^* = p_{itj} km + a_{it} \frac{p_{itj}O_{1it} - p_{it2}O_{2it}}{O_{1it}O_{2it}}$, and $q_{it2}^* = p_{itj} km + a_{it} \frac{p_{it2}O_{2it} - p_{it1}O_{1it}}{O_{1it}O_{2it}}$
Problem P3 is solved in two steps: 1) the agents determine which constraint is binding, (recall that the amount the agents can bet is either constrained by their available funds $w_{it} + m$ or by the house enforced maximum bet $km$); and 2) the agents determine the optimal bets given the binding constraints, i.e., the agents solve either P4 or P5.

### 3.3.4. Distribute Payout

Whenever an object is classified as belonging to the positive class, detailed investigations are necessary to establish the true class of object $t$. There are two possibilities: 1) the true class of $t$ is positive $j=1$, (i.e., TP) or 2) the true class of $t$ is negative $j=2$, (i.e., FP). While final bets are deducted from the agents’ wealth immediately, as described in Section 3.3.2, there is a time lag corresponding to $v$ transactions before the winnings can be paid out to the agents because of the time taken by the investigations. This is the same mechanism used in sports betting (and other types of future markets) where bets are collected when bets are placed and winnings are paid out after the game/race has been decided.

If on investigating transaction $t'$ that occurred $v$ transactions before the current transaction $t$, if $t'$ is found to be a member of the positive class $j=1$, then agent $i$’s wealth is updated using (19), else in the case of negative class $j=2$, agent $i$’s wealth is updated using (20):

\[
    w_{it} = w_{it} + Q_{t'}(q_{i,v,1}/Q_{t,v,1})
\]  
\[
    w_{it} = w_{it} + Q_{t'}(q_{i,v,2}/Q_{t,v,2})
\]

### 3.3.5. Determining Cutoff

The cutoff $C_j$ is used as a threshold to classify object $t$ into $j$ when $P_{jt} \geq C_j$, where $P_{jt}$ is ensemble $E$’s overall probability estimate that $t$ belongs to $j \in J$ and is given by $1/O^t_{jt}$, where $O^t_{jt}$ is the final odds that $t$ belongs to $j$. The choice of $C_j$ impacts the ratio of False Positives (FP) to False Negatives (FN), and therefore impacts the net benefits of a fraud detection program. When a fraudulent transaction is not investigated, then the cost of this FN can be significant compared to the cost of investigating a non-
fraudulent transaction classified as fraudulent, i.e., a FP. However we still want to keep the investigation costs as low as possible, and therefore need to limit both the number of FNs and FPs. $C_j$ should therefore be set at an appropriate level that maximizes the net-benefits of a fraud detection program by reducing the total cost of FN and FP across multiple transactions. Since FN is not investigated in many “real world” applications, including fraud detection, an alternate measure for determining the cutoff level is to maximize the net-benefits accrued by investigating True Positives (TP) and FP.

Let $c^i$ be the cost of investigating a transaction that has been classified as positive (i.e., FP or TP), and $c'$ be the benefits accrued when a transaction that is positive is classified as positive (i.e., TP). Furthermore, given $n_k$ prior transactions where the $k^{th}$ cutoff value $C_{jk}$ was used for class $j=1$, and the number of TP $t_k$ and FP $f_k$, the conditional probability of TP, $P(TP/C_{jk})= t_k/n_k$. The total net-benefit from using $C_{jk}$ is then given by $B_{jk} = c'P(TP/C_{jk})n_k - c'in_k$. Thus for future $n$ transactions, the optimal cutoff $C_j$ could be obtained by maximizing the net-benefits $B$ as follows.

$$B = \max_{C_j \geq 0} P(TP/C_j)ncc' - c'n$$ (21)

We use a dynamic approach to determine $C_j$ where not all objects have to be investigated. The central idea behind this approach is to perturb the current (i.e., $k^{th}$) cutoff $C_{jk}$ by a small amount $\Delta$ such that $k+1^{th}$ cutoff $C_{jk+1} = C_{jk} + \Delta$ and the $k+2^{th}$ cutoff $C_{jk+2} = C_{jk} - \Delta$. Using $C_{jk+1}$ the next $n_{k+1}$ objects are classified followed by using $C_{jk+2}$ to classify the next $n_{k+2}$ objects. The new cutoff $C_{jk+3}$ is the cutoff corresponding to the maximum of ($B_k$, $B_{k+1}$, $B_{k+2}$). We present an evaluation of our dynamic cutoff selection algorithm in Section 4.3.

4. **Experiment**

In this section we describe the computational experiments undertaken to test the effectiveness of IMF. We first describe the base-classifiers and the datasets used in the computational experiments. We then discuss the experimental design and the experimental factors, and provide a brief description of the experimental procedure. This is then followed by a discussion of two additional experiments used to gain insight into:
1) the appropriateness of cutoffs $C_j$ selected using the dynamic
cutoff selection algorithm, and 2) the performance impact of time-
lag $v$. This section is concluded with discussions on two
experiments used to gain insights into the sensitivity of IMF
performance to different values of parameters, $r$ and $k$.

4.1. Base-Classifiers and Data

Using Weka (version 3.4.6) 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 the 17 datasets that were
obtained from the UCI Machine Learning Repository (see Table 2). Datasets that included more than two
classes were modified by either creating multiple subsets with only two classes in each subset or by
combining classes. Furthermore, in order for computationally complex base-classifiers to complete the
classification using a reasonable amount of resources (primarily memory and CPU time), datasets with a
large number of observation and/or attributes were filtered randomly based on records and/or attributes.

<table>
<thead>
<tr>
<th>Table 1: Base-Classifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADTree</td>
</tr>
<tr>
<td>BayesNet</td>
</tr>
<tr>
<td>ConjunctiveRule</td>
</tr>
<tr>
<td>DecisionStump</td>
</tr>
<tr>
<td>DecisionTable</td>
</tr>
<tr>
<td>Ibk</td>
</tr>
<tr>
<td>J48</td>
</tr>
<tr>
<td>JRip</td>
</tr>
<tr>
<td>KStar</td>
</tr>
<tr>
<td>LMT</td>
</tr>
<tr>
<td>LWL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>Adult</td>
</tr>
<tr>
<td>Wisconsin Breast Cancer</td>
</tr>
<tr>
<td>Contraceptive Method Choice</td>
</tr>
<tr>
<td>Horse Colic</td>
</tr>
<tr>
<td>Covertype (class 1 &amp; 2)</td>
</tr>
<tr>
<td>Covertype (class 3 &amp; 4)</td>
</tr>
<tr>
<td>Covertype (class 5 &amp; 6)</td>
</tr>
<tr>
<td>Australian Credit Approval</td>
</tr>
<tr>
<td>German Credit</td>
</tr>
<tr>
<td>Pima Indians Diabetes</td>
</tr>
<tr>
<td>Thyroid Disease</td>
</tr>
<tr>
<td>Labor</td>
</tr>
<tr>
<td>Mushrooms</td>
</tr>
<tr>
<td>Sick</td>
</tr>
<tr>
<td>Spambase</td>
</tr>
<tr>
<td>Splice-junction Gene Sequences</td>
</tr>
<tr>
<td>Waveform</td>
</tr>
</tbody>
</table>
With an average dataset size of 5,350 records, a total of 2,000,900 base-classifiers decisions (5,350 records \(\times\) 17 datasets \(\times\) 22 base-classifiers) were generated. These decisions, grouped based on dataset, were imported into Microsoft Access where combiner methods implemented using Visual Basic and LINGO combined the data. Each dataset was combined 100 times (see Section 4.2) for a total of 9,095,000 (100 dataset combinations \(\times\) 5,350 average dataset size \(\times\) 17 datasets) aggregated 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. As such, combiner method is our primary factor of interest (see Section 4.2.2). To evaluate the sensitivity of our results (see Section 4.2.3) the experiment also included two additional factors (cost-to-benefit ratio and number-of-agents) and three additional covariates (dataset positive ratio, dataset size and dataset average base-classifier accuracy). Table 3 summarizes these variables.

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. We instead used two factorial block designs (4 combiner methods \(\times\) 11 number of agents, and 4 combiner methods \(\times\) 14 cost-to-benefit ratios), for a total of 100 treatment groups, where the cost-to-benefit ratio factor and the number of agents factor were held constant in the 4\(\times\)11 and 4\(\times\)14 factorial designs respectively.

Net benefit, the dependent variable and the covariates were measured for each of the 17 datasets within each of the 100 treatment cells for a total of 1,700 observations with 748

<table>
<thead>
<tr>
<th>Table 3: Experimental Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variable</strong></td>
</tr>
<tr>
<td>Net Benefit</td>
</tr>
<tr>
<td>Combiner Method</td>
</tr>
<tr>
<td>Number of Agents</td>
</tr>
<tr>
<td>Cost-Benefit Ratio</td>
</tr>
<tr>
<td>Dataset Size</td>
</tr>
<tr>
<td>Dataset Agent Accuracy</td>
</tr>
<tr>
<td>Dataset Positive Ratio</td>
</tr>
</tbody>
</table>
4.2.1. Dependent Measure

Most combiner method research use hit-rate, i.e., the percentage of objects correctly classified, as the dependent measure. However, hit-rate only provides an accurate measure of the combiner method effectiveness under one specific scenario: when the benefits from TP and true negatives (TN), and the cost of FP and FN are the same in absolute terms, which often is not the case. To compare combiner method performance we instead examine the net-benefit that each of the methods provide, given various benefits and costs associated with true and false positives. Details on how net-benefit was computed can be found in Section 4.2.3.

4.2.2. Combiner Method Factor

As mentioned earlier, the primary purpose of the experiment was to compare the performance of IMF to existing combiner methods, and therefore we included combiner method as a factor that was manipulated at four levels, IMF, AVG, MAJ and WAVG.

IMF was compared to MAJ, AVG and WAVG given that prior research indicated that AVG and MAJ performed well compared to other existing combiner methods, as discussed earlier, while WAVG was included primarily because of its similarity to IMF, given that IMF generated a wealth weighted average. Recall that IMF only learns from objects that are classified as positive, i.e., \( j = 1 \). By applying the same assumption to WAVG we could not implement the commonly used WAVG implementation, which requires training data. We therefore implemented a dynamic version of WAVG where the weights were determined as the ratio of the individual classifiers’ hit rate to the total hit-rate of all the classifiers in the ensemble, here the hit-rate was updated for each object classified as positive.

The effectiveness of IMF was evaluated against MAJ, AVG and WAVG by examining if the net benefit from IMF was significantly greater than the net benefit from MAJ, AVG and WAVG.
4.2.3. Sensitivity Analysis

Number of Agents – The number of agents factor was manipulated at 11 levels: 2, 4, 6,…, and 22 agents, and was included because prior research did find that the number of agents in an ensemble impacted ensemble classification performance (Lam 2000). The agents were randomly selected at each of the treatment levels, but the selection process was cumulative in nature. To clarify, for ensembles consisting of 2 agents, the two agents were randomly selected from the 22 existing base-classifiers, for ensembles with 4 agents, two additional agents were 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 examined if the net-benefit from the combiner methods was moderated by number-of-agents.

Cost-to-Benefit Ratio - The cost of investigations, and the benefits derived from TP classifications impact the net-benefit provided by any classification effort. As the cost-to-benefit ratio is domain specific, we include a wide range of cost-to-benefit ratios, 14 in total, to explore the generalizability of our results: 1:100; 1:50, 1:25, 1:15, 1:10, 1:7.5, 1:5, 1:4, 1:3, 1:2, 1:1.5, 1:1, 1.5:1, and 2:1, holding the number-of-agents factor constant at 10 agents. To clarify, consider the 1:100 ratio in the case of a fraud audit, the ratio can for example represent that the average benefit (or cost avoidance) of detecting a fraud is $10,000 while the average investigation cost is $100. Say that we investigate 100 cases and 5 of these cases are fraudulent, then the net benefit from these investigations is $40,000 (10,000*5-100*100). To examine the sensitivity of the combiner method performance to the cost-to-benefit ratio, we investigated whether or not the net benefit from the combiner methods was moderated by the cost-to-benefit ratio.

Dataset Average Accuracy - Average base-classifier accuracy, measured as hit-rate for each dataset, is included as a possible interaction term. We believe that it is possible that some combiner methods perform relatively well when the average accuracy is low, while others perform better when the average accuracy is higher. To examine this interaction we tested if the net benefit from the combiner methods was moderated by the dataset average base-classifier accuracy.
**Dataset Size** - Dataset size refers to the number of records in the dataset and varies from 57 to 32,561 records. Dataset size was included as a covariate primarily to examine if and how size impacts the relative performance of IMF and WAVG, as these combiner methods learn. For example, if the data size is very small, the extent of redistribution of wealth in IMF is small. We therefore evaluated if the net benefits of the combiner methods were moderated by the dataset size.

**Dataset Positive Ratio** - The positive ratio of the dataset refers to the number of positive objects divided by all objects in the dataset. Positive ratio was included as a covariate to test if the relative combiner method performance depends on the level of the dataset positive ratio. Theoretically, if there is a difference in combiner method performance, then this difference will only be evident in datasets with positive ratios that are in the medium range as for very low (high) positive ratios, any trivial classifier that always predicts the object as negative (positive) can always do well. For extreme values of positive ratios, it is unlikely that any of the combiner methods can improve the results from the trivial classifier, since various combiner methods are likely to implement the trivial rule of classifying all objects as positive or negative, depending on if the positive ratio is high or low, by setting the cutoff close to 0 or 1. The range of probabilities for which the trivial rule is effective is impacted by cost-to-benefit considerations. The experimental cost-benefit ratio manipulation (low to medium) makes the trivial rule that classifies everything as positive effective over a wider range of positive ratios, i.e., medium to high rather than just high positive ratios, while it makes the trivial negative rule effective over a smaller range, i.e., only extremely low positive ratios (Witten and Frank 2005). We therefore evaluated if the net benefit from the combiner methods was moderated by the dataset positive ratio.

### 4.3. Investigation of Dynamic Cutoff, Time Lag and IMF Parameters

#### 4.3.1. Dynamic Cutoff Selection

In our experiments we compare the performance of the combiner methods using the optimal cutoffs for each treatment rather than the performance of the combiner methods using dynamically determined cutoff in order to isolate the treatment effect from noise introduced by using the dynamic cutoff selection.
algorithm. In order to determine the optimal cutoffs, we did run MCC 101 times for each treatment using a different cutoff level (0, 0.01, 0.02, … 1) for each run. The cutoff from the run that generated the highest total net-benefit was then labeled as the optimal cutoff for that specific treatment. While the optimal cutoff approach described above allows us to evaluate the relative performance of the combiner methods in our experimental setting, this approach is not useful in many “real world” applications as the true class of all objects is required to be investigated. A dynamic cutoff selection algorithm was therefore introduced in Section 3.3.5, but this algorithm does not guarantee the selection of optimal cutoff levels.

When evaluating the appropriateness of the dynamic cutoff selection algorithm we were only interested in the effectiveness of the algorithm itself. To reduce the amount of noise in the experiment we therefore only used one combiner method. We used MAJ because MAJ takes less time to run. To evaluate the effectiveness of the dynamic cutoff algorithm, the net benefit generated using MAJ with dynamically set cutoffs (DYN-MAJ) was compared to net benefits generated using MAJ with optimal cutoffs (OPT-MAJ) and randomly generated cutoffs (RAN-MAJ). Net benefits for these three treatments were measured from runs using each of the 17 datasets. The net benefit generated using RAN-MAJ was used as a minimum performance level that DYN-MAJ should perform significantly better than, i.e., we evaluated if the net benefit from DYN-MAJ is significantly higher than the net benefit from RAN-MAJ. The utility of DYN-MAJ was also evaluated by comparing net benefits generated using DYN-MAJ to net benefits generated using OPT-MAJ. An insignificant result for this test would not let us conclude that OPT-MAJ performs better than DYN-MAJ.

4.3.2. Time Lag and Performance

In the main experiment the true class of t is given instantly after t has been classified, but it might in reality take some time to determine the true class of t, as described in Section 3.3.4. To determine the performance impacts of such time lags we performed an experiment where \( w_v \) was not updated until \( v \) additional objects had been classified. \( v \) was manipulated at six different levels: 0, 1, 5, 10, 25 and 50 percent of the size of the dataset, for each of the 17 datasets while the main experiment factors were held
constant as follows: combiner method = IMF; number of agents = 10; and cost-to-benefit ratio = 1:10. Using these treatments we investigated if the net benefit from 0%-IMF is significantly different to the net benefit from 1%-IMF, 5%-IMF, 10%-IMF, 25%-IMF and 50%-IMF.

### 4.3.3. Selection of IMF Parameters

**Binary Search Stopping Parameter** $\varepsilon$ - The tolerance value $\varepsilon$ is used in binary search (see Section 3.3.1) to determine when to stop the search, given by $P^u - P^l \leq \varepsilon$. To gain a better understanding of how to select an appropriate value for $\varepsilon$ and if this selection is domain dependent we ran a sub-experiment where different values of $\varepsilon$ were tested. For a given value of $\varepsilon$ (manipulated at 0.1, 0.01, 0.001, 0.0001 and 0.00001) we ran IMF on fourteen randomly select datasets while holding the other factors constant as follows: number of agents = 10; and cost-to-benefit ratio = 1:10. In our tests we were interested in investigating interactions between $\varepsilon$ and the different data set characteristics to understand if $\varepsilon$ were to be set to different values depending on the specific domain, and if no interactions existed, the direct impact of $\varepsilon$ on net benefit to determine if some values of $\varepsilon$ were better than others.

**Maximum Bet Multiplier** $k$ - To ensure that the ensemble was not completely dominated by a minority of the agents the maximum bet $km$ was required to be set sufficiently low. However, the ensemble could also benefit from weighting the input of better performing agents more heavily than the input from poorer performing agents. This required not setting the value of $k$ too low. Thus, there was perhaps an optimal value of $k$ that led to maximum net benefits while using IMF. For a given value of $k$ (manipulated at 1, 2, 5, 10, 25, 50, 75, 100, 125, 150, 200, 250, 350, 500, and 1000), we ran IMF on all datasets with the number-of-agents factor set at 10 and the cost-to-benefit ratio set at 1:10. We were interested in investigating interactions between $k$ and data set characteristics (average agent accuracy, size and positive ratio) in order to determine whether the optimal values of $k$ set were domain specific, and determine the direct effects of $k$ on the net benefits if no interactions were significant in order to understand if some values of $k$ were better than others.
5. Results

In this section we first describe the experimental results from the main experiment, including descriptive statistics, evaluation of ANCOVA assumptions and statistical results, including sensitivity analyses. We then evaluate the performance of the dynamic cutoff selection algorithm, and how different time lags and the selection of IMF parameters impact the performance of IMF.

5.1. Relative Combiner Method Performance

5.1.1. Overview

Table 4 provides an overview of the result data organized by the three statistical analysis datasets used in our statistical analysis, the two statistical analysis datasets based on the 5x11 and 5x14 factorial designs (combiner method by number-of-agents and combiner method by cost-to-benefit respectively) and the pooling of these two statistical analysis datasets (the pooling was possible as neither interaction was significant, as discussed in 5.1.3).

After log transformations of the dependent variable the normality assumption: Shapiro-Wilks (p>0.08); Kologmorov-Smirnov (p>0.15); Cramer von Mises (p>0.25); and Anderson-Darling (p>0.25), and equal variance assumption: Levene’s (p=0.99), were satisfied. Furthermore, we did not observe any influential outliers since all observations had Cook’s D values less than 1 (D<0.6).

---

1 We report actual p-statistics throughout the paper but for significance testing we use an alpha of 0.05. Note that we performed a large number of tests. To retain an overall type I error rate of 0.05 for these tests, while at the same time balancing the risk of committing type II errors, we used a modified Bonferroni procedure (Jaccard and Wan 1996).
5.1.2. Combiner Method Main Effect

The combiner method main effect was tested using the model shown in (22) and the pooled result set described earlier. Note that for each UCI dataset*number-of-agents and UCI dataset*cost-to-benefit ratio combination the same four combiner methods were tested, we therefore block for the dataset, number-of-agents, cost-to-benefit ratio, dataset*number-of-agents and dataset*cost-to-benefit ratio effects.

\[
\ln(\text{net benefit}) = \beta_0 + \beta_1 \text{combiner method} + \text{block} \quad (22)
\]

The combiner method main effect was significant (p<0.001) and the follow-up analysis showed that IMF significantly outperformed AVG (p=0.0030), WAVG (p=0.0011) and MAJ(p=0.0001).

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 were respectively tested using the model shown in (23) and the 5x11 factorial design, and model (24) and the 5x14 factorial designs. Note that each combiner method was tested for all combinations of UCI dataset*number-of-agents and UCI dataset*cost-to-benefit ratio in respective models, we therefore block for these interactions in respective models.

\[
\ln(\text{net benefit}) = \beta_0 + \beta_1 \text{combiner method} + \beta_2 \text{number-of-agents} + \beta_3 \text{combiner method*number-of-agents} + \text{block} \quad (23)
\]

\[
\ln(\text{net benefit}) = \beta_0 + \beta_1 \text{combiner method} + \beta_2 \text{cost-to-benefit ratio} + \beta_3 \text{combiner method*cost-to-benefit ratio} + \text{block} \quad (24)
\]

The combiner method*number-of-agents (p=0.7393) and combiner method*cost-to-benefit ratio (p=0.8209) interactions were insignificant. The results as such do not support that the performance advantage of IMF over AVG, WAVG and MAJ is moderated by the number of agents in the ensemble or by the domain dependent cost-to-benefit ratio.
The sensitivity of the relative combiner method performance result to the dataset average agent accuracy, size and positive ratio was tested using the same blocking factor and result set used for (22):

$$\ln(\text{net benefit}) = \beta_0 + \beta_1 \text{combiner method} + \beta_2 \text{dataset average agent accuracy} + \beta_3 \text{dataset size} + \beta_4 \text{dataset positive ratio} + \beta_5 \text{combiner method*dataset average agent accuracy} + \beta_6 \text{combiner method*dataset size} + \beta_7 \text{combiner method*dataset positive ratio} + \text{block}$$  (25)

The results did not show that relative combiner method performance results were sensitive to either the dataset size ($p=0.9223$) or the dataset average agent accuracy ($p=0.3040$). We did however find that the combiner method*dataset positive ratio interaction was significant ($p<0.0001$). To gain a better understanding of the nature of this interaction we plotted the interaction against log net benefit (see Figure 5). Note in Figure 5 that for medium to low dataset positive ratios log net-benefit is higher for IMF than for the other combiner methods (supported by the intercept of IMF being significantly higher than that of the other combiner methods, $p<0.0001$), and at higher positive ratios the lines merge (supported by the slope of IMF being significantly lower than that of the other combiner methods, $p<0.0001$).

To gain further insight into this interaction we divided the datasets into three groups based on the dataset positive ratio, a high group with the three datasets with the highest positive ratio ($>60\%$), low group with the three datasets with the lowest positive ratios ($<10\%$) and a middle group with the middle 11 datasets. Each group was then used for ANOVAs with a single factor, combiner method, using the same blocking variables as (22). In the low group IMF

![Figure 5: Combiner Method x Positive Ratio Interaction](image)
significantly outperformed the other combiner methods (p=0.0010), while in the middle and high groups the results were insignificant (p=0.1843) and (p=0.2849) respectively (see Figure 6).

5.1.4. Investigation of Dynamic Cutoff, Time Lag and IMF Parameters

The Dynamic Cutoff Selection algorithm was tested using the model shown in (26), and a similar dataset as the pooled result, where instead of the combiner method factor, three different MAJ implementations were used to generate the result set (for a total of 1275 observations). We therefore blocked for the dataset, number-of-agents, cost-to-benefit, dataset*number-of-agents and dataset*cost-to-benefit effects:

\[
\ln(\text{net benefit}) = \beta_0 + \beta_1 \text{ cut-off selection type} + \text{block}
\]  

(26)

The cutoff selection type main effect (p=0.0006) was significant and follow-up analysis, see Figure 7, showed that DYN-MAJ significantly outperforms RAN-MAJ (p =0.0037), while there was no significant difference between DYN-MAJ and OPT-MAJ (p=0.8575).

The impact of Time Lag on Net Benefit was tested using the model shown in (27), and statistical analysis datasets derived from holding the number-of-agents constant at 10. All UCI datasets were used for all treatments in model (for a total of 102 observations), we therefore blocked for the dataset factor. The lag-level

![Figure 6: Combiner Method x Dataset Positive Groups Interaction](image1)

![Figure 7: Cut-Off Selection Comparison](image2)
The main effect (p=0.8908) was insignificant. Thus the results showed that time lag did not impact IMF performance.

\[
\ln(\text{net benefit}) = \beta_0 + \beta_1 v + \text{block} \quad (27)
\]

We did not find any evidence that the value of the binary search stopping parameter \(\epsilon\), within the tested range (0.1, 0.01, 0.001, 0.0001, 0.00001), impacted the performance of IMF nor the way \(\epsilon\) impacted performance of IMF was domain dependent. More specifically, while blocking for the dataset effect on net benefit, the \(\epsilon \times \text{dataset positive ratio} \ (p=0.1190)\), \(\epsilon \times \text{dataset size} \ (p=0.7869)\) and \(\epsilon \times \text{dataset average agent accuracy} \ (p=0.8838)\) interactions, and the \(\epsilon\) main effect (0.3234) were insignificant.

Finally, in order to choose appropriate values for \(k\), and to investigate if the choice of \(k\) was domain dependent, we used the model shown in (28), while manipulating \(k\) at 15 different levels while holding the number-of-agents at 10 and the cost-to-benefit ratio constant at 1:10. All UCI datasets were used for all treatments (for a total of 255 observations). We therefore blocked for the dataset effect in (28).

\[
\ln(\text{net benefit}) = \beta_0 + \beta_1 k + \beta_2 \text{dataset average agent accuracy} + \beta_3 \text{dataset positive ratio} + \\
\beta_4 \text{dataset size} + \beta_5 k \times \text{dataset average agent accuracy} + \\
\beta_6 k \times \text{dataset positive ratio} + \beta_7 k \times \text{dataset size} + \text{block} \quad (28)
\]

The \(k \times \text{dataset average agent accuracy} \) interaction was significant (p=0.0296), while the \(k \times \text{dataset positive ratio} \) (p=0.9227) and \(k \times \text{dataset size} \) (p=0.0521) interactions were insignificant. Scatter plots with trend lines and the raw data tables for the standardized log net benefit of the different datasets at the 15 different \(k\) values indicated three average agent accuracy groupings: under 80\%, between 80 and 90\%, and above 90\%, see (see Figure 8). Based on Figure 8, within these groupings net benefits appear higher: 1) for low \(k\) values (treatment levels 1 to 4) in the average agent accuracy under 80\% group, 2) middle to low \(k\) values (treatment levels 2 to 6) in the between 80 and 90\% group, and 3) high \(k\) values (treatment levels 10 to 13) in the above 90\% group. We utilized these rules in the main experiment and set \(k\) to 2, 10 or 200 depending on the average agent accuracy of the dataset processed.
6. Discussion

We next provide a discussion of the results presented above.

6.1. Combiner Method Performance

6.1.1. Overall Performance of IMF

To understand why IMF significantly outperforms 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 base-classifiers have the same amount of funds available for placing bets in the market, i.e., if the aggregation was not wealth weighted (Wolfers and Zitzewitz 2006). However, as more accurate agents become wealthier these agents end up dictating the market prices to a greater degree than the less accurate agents, and as the equilibrium prices represent the aggregated probabilities of the ensemble, the more

* y-axis: z-value of log net benefit
** x-axis: treatment level of \( k \) factor
*** markers: average-agent accuracy (percentage shown in label box)

Figure 8: \( k \) by, high, medium and low groupings
accurate agents have a greater impact on the ensemble’s decision than the less accurate agents. Thus, in IMF the ensemble decision is an accuracy-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 since MAJ is also a non-weighted combiner method with performance similar to AVG.

When comparing IMF to WAVG, we need to examine the major difference between IMF and WAVG, which is that WAVG assigns weights solely based on the accuracy of the base-classifiers relative to the accuracy of the other base-classifiers, while IMF places progressively greater weight on better performing agents’ decisions as agents with wealth above what they are allowed to bet, become more aggressive, i.e., Lemma 2 provides a more aggressive betting behavior than Lemma 1. Furthermore, in IMF the weights are also adjusted based on the degree of the agents’ accuracy as opposed to WAVG where the weights are adjusted only based on the confusion matrix. In IMF the agents’ wealth increases (decreases) to a greater degree the more (less) accurate the agent is, as the agents’ bets are increasing in the agents’ probability estimates. Thus, agents that are correct and more certain will receive a higher payout than agents that are correct but less certain since the bets of the more certain agents will be higher, and vice versa. Finally, the weights in IMF are also adjusted based on the diversity of the agents as in IMF the wealth of different agents is a factor of the agents’ relative level of diversity. An agent with a correct bet will receive a greater payout if the odds are higher for that event, and these odds increase with increases in the amount bet by “disagreeing” agents.

6.1.2. Sensitivity Analysis

The results discussed in Section 6.1.1 were not sensitive to the number of agents in the ensemble, the cost-to-benefit ratio, dataset size and the dataset average agent accuracy. We did however find that the relationship between combiner method and net benefit was moderated by the dataset positive ratio. The results showed that IMF outperformed the other combiner methods at low positive ratios, and that there was no significant difference at higher positive ratio levels as theoretically expected, see discussion in Section 4.2.3. This interaction shows that IMF performs well at all positive ratio levels and that IMF
outperforms the other combiner methods when there is room for improvement, i.e., when the trivial rule is not effective. In other words, IMF outperforms the other combiner methods when it matters and otherwise performs on par with the other combiner methods. Note that our experimental datasets only had positive ratios in the low to medium range. As such we only tested cost-to-benefit and positive ratio combinations with low or medium cost-to-benefit ratios, as the combiner methods would just classify everything as negative for high cost-to-benefit ratios.

6.1.3. Dynamic Cutoff, Time Lag and IMF Parameters

As the true object class is not revealed for objects classified as negative in many real world applications the optimal cutoffs cannot always be determined. The results show that the proposed dynamic cutoff selection algorithm is useful in these situations as it outperforms randomly selected cutoff and its performance is not significantly different from the optimal cutoffs. The dynamic cutoff algorithm can also be useful in situations where the optimal cutoff changes over time, recall that the algorithm explores cutoffs both below and above the current dynamically determined optimal cutoff and adjusts this optimal cutoff to the best cutoff of the three cutoffs.

The results do not show that time lag between object classification and object determination impacts the performance of IMF within the range tested (0 to 50% of the records in the dataset). Thus, we do not have any evidence that the performance of IMF would deteriorate if implemented in environments with long time lags between object classification and object determination. Furthermore, if IMF was, for whatever reason, implemented in a domain where true object class is not determined for any objects, then the performance of IMF would be similar to AVG.

Finally, when implementing IMF two parameters have to be set, the binary search stopping parameter $\epsilon$ and the maximum bet multiplier $k$. Within the range of $\epsilon$ tested (0.1, 0.01, 0.001, 0.0001, 0.00001) the results did not show any significant interactions or main effects.

The results also show that $k$ should be set to different values for different datasets depending on the average accuracy of the agents for the specific dataset. When the average agent accuracy is under 80%,
between 80 and 90%, and above 90%, good values for $k$ were 2, 10 or 200 respectively. These results indicate that lower the average agent accuracy, it is important that one or a few agents do not dominate the ensemble.

7. Conclusion and Future Research Directions

In this article we have combined three diverse research streams, MCC, MAS and IM, to design a new and novel MCC combiner method, IMF. We have showed, through an extensive experiment using 17 different datasets and 22 base-classifiers with multiple sensitivity analyses, that IMF overall outperforms the three benchmark combiner methods AVG, WAVG and MAJ. We have furthermore described how the design of IMF allows it to more readily integrate with MAS architectures, which we believe will become a dominant paradigm for distributed problem solving. We have also introduced and evaluated a dynamic cutoff selection algorithm that is both novel and provides utility in domain where it is not feasible to investigate all objects.

7.1. Future Research Opportunities

From an MCC perspective this research was positioned within the combiner method research stream and did not consider different types of MCC architectures. Future MCC research is therefore needed to test the relative effectiveness of IMF compared to other combiner methods in different MCC architectures, for example bagging and boosting.

Finally, this research only investigates one type of agent behavior, i.e., using only one (though widely accepted and commonly used) utility function. Future research can investigate the performance impact of other types of agent behavior, for example using other utility functions (Constant Absolute Risk Aversion, Constant Relative Risk Aversion, etc), modeling the agents to update their beliefs based on market signals (perhaps using game theory), mixing agents with different utility functions, using a combination of human and software agents, etc.
8. Acknowledgments

This paper benefited from comments of reviewers and participants at the International Conference on Information Systems conference 2006, the COBA Research Workshop at the University of South Florida 2006 and the Workshop on Information Technologies & Systems 2005.

9. References


- 33 -