Every Team Makes Mistakes: An Initial Report on Predicting Failure in Teamwork

Vaishnavh Nagarajan[1], **Leandro Soriano Marcolino**[2], **Milind Tambe**[2]
[1] Indian Institute of Technology Madras, Chennai, Tamil Nadu, 600036, India
vaish@cse.iitm.ac.in
[2] University of Southern California, Los Angeles, CA, 90089, USA
{sorianom,tambe}@usc.edu

Abstract

Voting among different agents is a powerful tool in problem solving, and it has been widely applied to improve the performance in machine learning. However, the potential of voting has been explored only in improving the ability of finding the correct answer to a complex problem. In this paper we present a novel benefit in voting, that has not been observed before: we show that we can use the voting patterns to assess the performance of a team and predict their final outcome. This prediction can be executed at any moment during problem-solving and it is completely domain independent. We present a preliminary theoretical explanation of why our prediction method works, where we show that the accuracy is better for diverse teams composed by different agents than for uniform teams made of copies of the same agent. We also perform experiments in the Computer Go domain, where we show that we can obtain a high accuracy in predicting the final outcome of the games. We analyze the prediction accuracy for 3 different teams, and we show that the prediction works significantly better for a diverse team. Since our approach is completely domain independent, it can be easily applied to a variety of domains, such as the video games in the Arcade Learning Environment.

Introduction

It is well known that aggregating the opinions of different agents can lead to a great performance when solving complex problems (Marcolino et al. 2014). In particular, voting has been extensively used to improve the performance in machine learning (Polikar 2012). Besides, it is an aggregation technique that does not depend on any domain, being very suited for general competence. However, a team of voting agents will not always be successful in problem-solving. It is fundamental, therefore, to be able to quickly assess the performance of teams, so that a system operator can take actions to recover the situation in time.

Current works in the multi-agent system literature focus on identifying faulty or erroneous behavior (Khalastchi, Kalech, and Rokach 2014; Lindner and Agmon 2014; Tarapore et al. 2013; Bulling, Dastani, and Knobbout 2013), or verifying correctness of systems (Doan et al. 2014). Such approaches are able to identify if a system is not correct, but provide no help if a correct system of agents is failing to solve a complex problem.

Other works focus on team analysis. Raines, Tambe, and Marsella (2000) present a method to automatically analyze the performance of a team. The method, however, only works offline and needs domain knowledge. Other methods for team analysis are heavily tailored for robot-soccer (Ramos and Ayanegui 2008) and focus on identifying opponent tactics (Mirchevska et al. 2014).

In this paper, we show a novel method to predict the final performance (success or failure) of a team of voting agents without using any domain knowledge. Hence, our method can be easily applied in a great variety of scenarios. Moreover, our approach can be quickly applied online at any step of the problem-solving process, allowing a system operator to identify when a team is failing.

The basic idea of our approach is to learn a classification model, based on the frequencies of agreement over all possible subsets of agents. Hence, the feature vector of our prediction model depends uniquely on the coordination method, and has no dependency on the domain. We present a preliminary theoretical model that explains why such an approach is able to make accurate predictions. Moreover, our model indicates that the prediction works better for diverse teams composed by different agents than for uniform teams made by copies of the same agent.

We present experimental results in the Computer Go domain, where we predict the performance of three different teams of voting agents: a diverse, a uniform, and an intermediate team (intermediate with respect to diversity). We show that we can predict win/loss of Go games with around 73% accuracy for the diverse and intermediate team, and 64% for the uniform team. We also study the predictions at every turn of the games, and compare with an analysis performed by using an in-depth search. We show that our method agrees with the analysis, from around the middle of the games, more than 60% of the time for all teams, but is significantly faster. As the technique depends only on the coordination method, the same approach can be applied for predicting the performance of teams of voting agents in any domain. An interesting extension would be to test our predictions in the video games of the Arcade Learning Environment.

Related Work

General competency has received considerable attention recently (Legg 2008; Hutter 2005; Genesereth, Love, and Pell 2005). The main goal is to develop methods that can be applied to a variety of domains (or even any possible domain). Recently, the Arcade Learning Environment was developed, allowing the experimentation of machine learning algorithms across hundreds of different games (Bellemare et al. 2013).

One common general approach to increase the performance of machine learning methods is to aggregate different learners through voting (Polikar 2012). Actually, the applicability of voting goes beyond machine learning, as it has been shown to improve the performance of many different systems (Marcolino et al. 2014; Mao, Procaccia, and Chen 2013). In particular, social choice researchers extensively study voting. Normally, voting is presented under one of two perspectives: as a way to aggregate different opinions, or as a way to discover an optimal choice (List and Goodin 2001; Conitzer and Sandholm 2005). In this work we present a novel view: we show that we can use the voting patterns as a way to assess the performance of a team. Such "side-effect" of voting has not been observed before, and was never explored in social choice theory and/or applications.

Concerning team assessment, the traditional methods rely heavily on tailoring for specific domains. Raines, Tambe, and Marsella (2000) present a method to build automated assistants for post-hoc, offline team analysis, but domain knowledge is necessary for such assistants. Other methods for team analysis are heavily tailored for robot-soccer, such as Ramos and Ayanegui (2008), that present a method to identify the tactical formation of soccer teams (number of defenders, midfielders, and forwards). Mirchevska et al. (2014) present a domain independent approach, but they are still focused on identifying opponent tactics, not on assessing the current performance of a team.

In the multi-agent systems community, we can see many recent works that study how to identify agents that present faulty behavior (Khalastchi, Kalech, and Rokach 2014; Lindner and Agmon 2014; Tarapore et al. 2013). Other works focus on verifying correct agent implementation (Doan et al. 2014) or monitoring the violation of norms in an agent system (Bulling, Dastani, and Knobbout 2013). Some works go beyond the agent-level and verify if the system as a whole conforms to certain specifications (Kouvaros and Lomuscio 2013), or verify properties of an agent system (Hunter et al. 2013). However, a team can still have a poor performance and fail in solving a complex problem, even when the individual agents are correctly implemented, no agent presents faulty behavior, and the system as a whole conforms to all specifications.

This work is also related to multi-agent learning (Zhang and Lesser 2013), but normally multi-agent learning methods are focused on learning how agents should perform, not on team assessment. An interesting approach has recently been presented (Torrey and Taylor 2013), where they have studied how to teach an agent to behave in a way that will make it achieve a high utility. Besides teaching agents, it should also be possible to teach agent teams. During the process of teaching, it is fundamental to identify when the system is leading towards failure. Hence, our approach could be integrated within a team teaching framework.

Finally, it has recently been shown that diverse teams of voting agents are able to outperform uniform teams composed by copies of the best agent (Marcolino, Jiang, and Tambe 2013; Marcolino et al. 2014; Jiang et al. 2014). Here we present an extra benefit in having diverse teams: we show that we can make better predictions of the final performance for diverse teams than for uniform teams.

Identifying when Things Go Wrong

We start by presenting our prediction method, and later in this section we will explain why the method works.

We consider scenarios where agents vote at every step (i.e., world state) of a complex problem, in order to take common decisions at every step towards problem-solving. Formally, let $\mathbf{T}$ be a set of agents t_i, $\mathbf{A}$ be a set of actions a_j and $\mathbf{M}$ be a set of world states m_k. The agents must vote for an action at each world state, and the team takes the action decided by plurality voting rule (we assume that ties are broken randomly). The team obtains a final reward r upon completing all world states. In this paper, we assume two possible final rewards: "success" (1) or "failure" (0).

We define the prediction problem as follows: without using any knowledge of the domain, identify the final reward that will be received by a team. This prediction must be executable at any world state, allowing a system operator to take remedy procedures in time.

We now explain our algorithm. The main idea is to learn a prediction function, given the frequencies of agreements of all possible agent subsets over the chosen actions. Let $\mathcal{P}(\mathbf{T}) = \{\mathbf{T_0}, \mathbf{T_1}, \ldots\}$ be the power set of the set of agents, a_i be the action chosen in world state m_j and $\mathbf{H_j} \subseteq \mathbf{T}$ be the subset of agents that agreed on a_i in that world state.

Consider the feature vector $\vec{\mathbf{x}} = (x_0, x_1, \ldots)$ computed at world state m_j, where each dimension (feature) has a one-to-one mapping with $\mathcal{P}(\mathbf{T})$. We define x_i as the *proportion* of times that the chosen action was agreed upon by the subset of agents $\mathbf{T_i}$. That is,

$$x_i = \sum_{k=0}^{||\mathbf{M_j}||-1} \frac{\mathbb{I}(\mathbf{H_k} = \mathbf{T_i})}{||\mathbf{M_j}||}$$

where $\mathbb{I}$ is the indicator function and $\mathbf{M_j} \subseteq \mathbf{M}$ is the set of world states from m_0 to the current world state m_j.

Hence, given a set $\tilde{\mathbf{X}}$ such that for each feature vector $\vec{\mathbf{x_t}} \in \tilde{\mathbf{X}}$ we have the associated reward r_t, we can estimate a function, $\hat{f}$, that returns an estimated reward between 0 and 1 given an input $\vec{\mathbf{x}}$. We classify estimated rewards above 0.5 as "success", and below 0.5 as "failure". In order to learn the classification model, the features are computed at the final world state.

We use classification by logistic regression, which models $\hat{f}$ as $\hat{f}(\vec{\mathbf{x}}) = \frac{1}{1+e^{-(\alpha+\vec{\boldsymbol{\beta}}^T\vec{\mathbf{x}})}}$, where α and $\vec{\boldsymbol{\beta}}$ are parameters that will be learned given $\tilde{\mathbf{X}}$ and the associated rewards.

$\{t_0\}$	$\{t_1\}$	$\{t_2\}$	$\{t_0, t_1\}$	$\{t_0, t_2\}$	$\{t_1, t_2\}$
0	0	0	1	0	0
0	0	0	1	0	0
0	0	0	2/3	0	1/3

Table 1: Example of the full feature vector after 3 iterations of problem solving.

	1	2
Iteration 0	0	1
Iteration 1	0	1
Iteration 2	0	1

Table 2: Example of the reduced feature vector after 3 iterations of problem solving.

While training, we eliminate two of the features. The feature corresponding to the subset $\emptyset$ is dropped because an action is chosen only if at least one of the agents voted for it. Also, since the rest of the features sum up to 1, and are hence linearly dependent, one of them is also dropped.

We also study a variant of this prediction method, where we use only information about the number of agents that agreed upon the chosen action, but not which agents exactly were involved in the agreement. For that variant, we consider a reduced feature vector $\vec{\mathbf{y}} = (y_0, y_1, \ldots)$, where we define y_i to be the proportion of times that the chosen action was agreed upon by any subset of i agents. Thus,

$$y_i = \sum_{k=0}^{||\mathbf{M_j}||-1} \frac{\mathbb{I}(||\mathbf{H_k}|| = i)}{||\mathbf{M_j}||}$$

where $\mathbb{I}$ is the indicator function and $\mathbf{M_j} \subseteq \mathbf{M}$ is the set of world states from m_0 to the current world state m_j. We compare the two approaches in Section Results.

Example of Features

We give a simple example of our proposed feature vectors. Consider a team of 3 agents: t_0, t_1, t_2. Let's assume two possible actions: a_0, a_1. Consider that, in 3 iterations of the problem solving, the voting profiles were:

Iteration 0: $a_0 a_0 a_1$

Iteration 1: $a_1 a_1 a_0$

Iteration 2: $a_0 a_1 a_1$

where we show which action each agent voted for at each iteration. Based on plurality voting rule, the action chosen for the respective iterations would be a_0, a_1, and a_1. In Table 1 we show an example of how the full feature vector will be defined at each iteration, where each column represents a possible subset of the set of agents, and each row represents one iteration (in increasing order from iteration 0 to iteration 2), and we mark the frequency that each subset agreed in the chosen action.

In Table 2, we show an example of the reduced feature vector, where the column headings define the number of agents involved in an agreement over the chosen action. Note that the reduced representation is more compact, but we have no way to represent the change in which specific agents were involved in the agreements.

Explanation

We present here our preliminary theoretical work that explains why we can use the frequencies of agreement to predict the success or failure of teams. We start with a simple example to show that we can use the outcome of plurality voting to predict the success of a team. Consider a scenario with two agents and two possible actions, a correct and an incorrect one. We assume, for this example, that agents have a probability of 0.6 of voting for the correct action and 0.4 of making a mistake.

If both agents vote for the same action, they are either both correct or both wrong. Hence, the probability of the team being correct is given by $0.6^2/(0.6^2 + 0.4^2) = 0.69$. Hence, if the agents agree, the team is more likely correct than wrong. If they vote for different actions, however, one will be correct and the other one wrong. Given that profile, and assuming that we break ties randomly, the team will have a 0.5 probability of being correct. Hence, the team has a higher probability of taking a correct choice when the agents agree than when they disagree ($0.69 > 0.5$). Therefore, if across multiple iterations these agents agree often, the team has a higher probability of being correct across these iterations, and we can predict that the team is going to be successful. If they disagree often, then the probability of being correct across the iterations is lower, and we can predict that the team will not be successful.

We now present our theoretical development. We base our approach in the view of voting as a way to estimate a ground truth. In social choice, this is modeled in the following way: there is a ground truth (i.e., the correct outcome, for example which action is correct and which one is incorrect), and the agents try to estimate the ground truth. Since the agents are not perfect, they have a noisy estimation of this ground truth. Hence, a noise model is defined as the probability of the agents voting for each action, given the correct outcome (Conitzer and Sandholm 2005).

Therefore, given a voting profile, we can estimate the likelihood of each action being the best, and the optimal decision is given by picking the action with the maximum likelihood estimate (MLE). Hence, a voting rule is going to be optimal if it corresponds to the MLE in any voting profile, given the noise model of the agents. We assume here that the agents are independent, and initially all actions are equally likely to be the best one (i.e., the prior probabilities are uniform over the action set), as usual in the classical voting models.

We consider, in this work, teams that play using plurality voting. Hence, we start by assuming that the agents have a noise model such that the likelihood is maximized by picking the action that received the highest number of votes in a profile. That is, we start by assuming that plurality is the optimal voting rule for the team. In this work, however, we are not interested in finding the best action given a voting profile, but rather in estimating how the probability of picking the correct action changes across different voting pro-

files with different number of agents agreeing on the chosen action. In other words, if the probability of the team being correct (i.e., choosing the best action) is higher in voting profiles where the amount of agreement is higher, then we will be able to predict the success of a team by observing the amount of agreement across the iterations.

We show in the following observation that if plurality is the optimal voting rule (i.e., it corresponds to the maximum likelihood estimate — MLE), we can use the amount of agreement among the agents to predict success.

Observation 1. *The probability that a team is correct increases with the number of agreeing agents m.*

Proof. Let c be the best action (whose identity we do not know). Let $v_1, v_2, \dots v_n$ be the votes of n agents. Let w be the action chosen by the highest number of agents. We want to know the probability of $c = w$:

$$P(c = w|v_1, v_2 \dots v_n) \propto P(v_1, v_2 \dots v_n|c = w)P(c = w)$$

For any noise model where plurality is MLE, we have that $P(v_1, v_2 \dots v_n|c = w)$ is proportional to the number of agents m that voted for c. Therefore, we have that $P(c = w|v_1, v_2 \dots v_n)$ is also proportional to m.

Hence, given two voting profiles $\mathbf{V}_1, \mathbf{V}_2$, with $m_{\mathbf{V}_1} > m_{\mathbf{V}_2}$, we have that $P_{\mathbf{V}_1}(c = w|v_1, v_2 \dots v_n) > P_{\mathbf{V}_2}(c = w|v_1, v_2 \dots v_n)$. Therefore, the team is more likely correct in profiles where a higher number of agents agree. □

In the next observation we show that we can increase the prediction accuracy by knowing not only how many agents agreed, but also which specific agents were involved in the agreement. Basically, we show that the probability of a team being correct depends on the agents involved in the agreement. Therefore, if we know that the best agents are involved in an agreement, we can be more certain of a team success.

Observation 2. *Given two profiles $\boldsymbol{V}_1, \boldsymbol{V}_2$ with the same number of agreeing agents m, the probability that a team is correct is not necessarily equal for the two profiles.*

Proof. We can easily prove by example. Consider a problem with 2 actions. Consider a team of 3 agents, where t_0 and t_1 have a probability of 0.8 of being correct, while t_2 has a probability of 0.6 of being correct. We should always pick the action chosen by the majority of the agents, as the probability of picking the correct action is the highest for all agents (List and Goodin 2001). Hence, plurality is MLE.

However, when only t_0 and t_1 agree, the probability that the team is correct is given by: $0.8^2 * 0.4/(0.8^2 * 0.4 + 0.2^2 * 0.6) = 0.91$. When only t_1 and t_2 agree, the probability that the team is correct is given by: $0.8 * 0.6 * 0.2/(0.8 * 0.6 * 0.2 + 0.2 * 0.4 * 0.8) = 0.59$. Hence, the probability that the team is correct is higher when t_0 and t_1 agree than when t_1 and t_2 agree. □

Based on that, one would expect the prediction for a uniform team to be better than the predictions for a diverse team, if the uniform team is composed by copies of the best agent (since their likelihood of being correct is higher). However, the best agent will not necessarily have noise models where the best action has the highest probability in all world states. Hence, a suboptimal action could have the highest probability, making the agents agree in the same mistakes (Marcolino, Jiang, and Tambe 2013). Therefore, when plurality is not actually a MLE in all world states, we have that Observation 1 will not hold in the world states where this happens. Hence, we will predict that the team made a correct choice, when actually the team was wrong, causing problems in our accuracy.

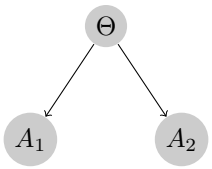


Figure 1: Agents are correlated by the similarities in their implementation Θ

Therefore, we can actually expect to make better predictions for a diverse team than for a uniform team. The basic intuition is that a diverse team is going to be less likely to agree in the same mistakes. For example, let's start by considering an idealized diverse team, where the agents never vote for the same mistakes (that is, never two or more agents vote for the same suboptimal action). In Marcolino et al. (2014) we show that this theoretically holds when the number of available actions goes to infinity. It is very easy to show that we can make better predictions for such idealized team. Basically, if two or more agents agree, the team would be correct with probability 1. A uniform team, however, might still be wrong if two or more agents agree. Hence, we can make better predictions for the idealized diverse team.

Of course, in less idealized scenarios the diverse team might still agree in a suboptimal action (although with a lower probability than in the uniform team case), making the situation more complex. Hence, to better understand how the prediction accuracy might change for different teams, we introduce a new model for diversity, where we do not assume that the agents are independent any more, but correlated by the similarities in their algorithms. We present in this paper our preliminary model, where we consider only two agents and two possible actions.

New Diversity Model We consider a situation with two agents and two possible actions, a correct (C) and an incorrect (I) one. Let A_1 and A_2 be the random variables corresponding to the actions chosen by the two agents in a world state, with probability distribution over the correct and incorrect action as $(s_1, 1 - s_1)$ and $(s_2, 1 - s_2)$ respectively. Contrary to previous models that assume independence (Marcolino, Jiang, and Tambe 2013; Marcolino et al. 2014), we do not assume independence here. We consider that the agents' algorithm might be similar in some account. We can model this dependency, which is not deterministic and moreover, cannot be observed, using a random variable θ. Hence, both A_1 and A_2 depend on this random variable θ. This situation is illustrated by the Bayesian network in Figure 1.

Since the agents are correlated, we define in Table 3 the joint probability distribution of (A_1, A_2), where $0 \leq e \leq 1$

(C,C)	(C,I)	(I,C)	(I,I)
s_1+s_2-e	$-s_2+e$	$-s_1+e$	$1-e$

Table 3: Joint probability table of (A_1, A_2).

is a constant defined by the correlation between the agents. We define $P(A_1 = C)$ and $P(A_2 = C)$ (i.e., s_1 and s_2, respectively) to be the *strength* of the agents, since it indicates how likely the agents will choose the best action. We define the *diversity* of the agents as the probability that they disagree: i.e., $P(A_1 \neq A_2)$, which is $2e - s_1 - s_2$ from our joint probability distribution table. Consequently, $\gamma = 1 + s_1 + s_2 - 2e$ is the similarity between the agents.

We now show that given two teams with the same strength (that is, let s_1' and s_2' be the *strength* of the agents of the second team, we have that $s_1 = s_1'$ and $s_2 = s_2'$), we can make better predictions for the team that has the higher diversity. We assume that the teams play using plurality voting (ties are broken randomly). Although we fix the strength of the agents, we consider that it is not known in advance by a system operator (i.e., for the prediction). This is true in many scenarios, as even though we might know the best agents overall, the actual probability of correctness changes according to each situation/world state (and some agent might be better than the best agents for some fixed world state (Marcolino, Jiang, and Tambe 2013)).

Theorem 1. *Given two teams with the same* strength*, we can make better predictions for the team with higher* diversity *(i.e., lower $\gamma : P(A_1 = A_2)$).*

Proof. Given a profile where the agents agree, the probability of the team being correct is given by:

$$P(C) = \frac{P(A_1 = A_2 = C)}{P(A_1 = A_2)} = \frac{(s_1 + s_2 - e)}{\gamma}$$

Let us make use of the following equalities:

$$P(A_1 = C) = P(A_1 = C, A_2 = C) + P(A_1 = C, A_2 = I) = (s_1 + s_2 - e) + (-s_2 + e)$$

$$P(A_2 = C) = P(A_1 = C, A_2 = C) + P(A_1 = I, A_2 = C) = (s_1 + s_2 - e) + (-s_1 + e) = (s_1 + s_2 - e) + 1 - \gamma - (-s_2 + e)$$

From the above two equalities:

$$P(A_1 = C)+P(A_2 = C) = s_1+s_2 = 2(s_1+s_2-e)+1-\gamma$$

Which can be rewritten as:

$$P(C) = \frac{(s_1 + s_2 - e)}{\gamma} = \frac{s_1 + s_2 - 1}{2\gamma} + \frac{1}{2}$$

Given that $0 \leq \frac{(s_1+s_2-e)}{\gamma} \leq 1$, we have that when $2 \geq s_1+s_2 > 1$, γ lies in $[(s_1+s_2)-1, 1]$. When $0 \leq s_1+s_2 < 1$, γ lies in $[1 - (s_1 + s_2), 1]$. For both cases, as γ increases towards 1 (for fixed s_1 and s_2), $\left|\frac{s_1+s_2-1}{2\gamma}\right|$ decreases. Hence, $P(C)$ gets closer to $\frac{1}{2}$. Therefore, for two different teams with the same s_1 and s_2, the team with higher γ (i.e., lower diversity) will have its probability of being correct, $P(C)$, closer to $\frac{1}{2}$.

However, the closer the probability of the chosen action being correct is to 0.5, *the more difficult it is to predict whether the action is correct or not*. Consider a Bernoulli trial with probability of success $p \approx 1$. In the learning phase, we will see many successes accordingly. In the testing phase, we will predict the majority of the two for every trial, and we will go wrong only with probability $|1-p| \approx 0$. On the other hand, if $p \approx 0.5$, our predictions, whatever they be, will be correct only with about 0.5 probability.

We must also consider the profiles where the agents disagree. In such a case, it is easy to see that regardless of the diversity (i.e., $P(A_1 \neq A_2)$), one cannot make predictions with an accuracy greater than 0.5, as we assume that the strength of the agents is not known. Given a profile where the agents disagree, the tie is going to be randomly broken, and therefore the probability that the chosen action is the correct one is 0.5. As we discuss above, we cannot have accurate predictions for such probability. □

We are currently working in extending this theory to cases with more than two agents and two actions. However, it already gives an idea of why we can have a better prediction for diverse teams than for uniform teams.

In the next section, we show that we can use our method to predict the outcome of Computer Go games at any turn, obtaining a high accuracy from the middle games. We also show that the prediction is better for a diverse team than for a uniform team with statistical significance.

Results

We test our prediction method in the Computer Go domain. We use 4 different Go software: Fuego 1.1 (Enzenberger et al. 2010), GnuGo 3.8, Pachi 9.01 (Baudiš and Gailly 2011), MoGo 4 (Gelly et al. 2006), and two (weaker) variants of Fuego (FuegoΔ and FuegoΘ), in a total of 6 different, publicly available, agents. Fuego is considered the strongest agent among all of them. The description of FuegoΔ and FuegoΘ is available in Marcolino et al. (2014).

We study three different teams: *Diverse*, composed by one copy of each agent; *Uniform*, composed by 6 copies of the original Fuego (initialized with different random seeds); *Intermediate*, composed by 6 random parametrized versions of Fuego (from Jiang et al. (2014)). In all teams, the agents vote together, playing as white, in a series of games against the original Fuego playing as black. The winning rates of the teams can be seen in Figure 2. The difference between *uniform* and *diverse* is not statistically significant

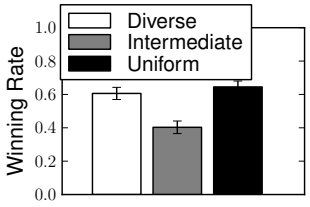


Figure 2: Winning rates of the 3 different teams used in our experiments.

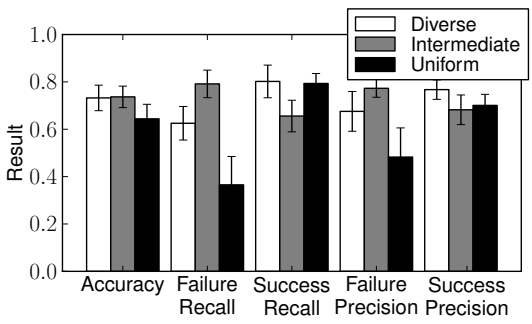


Figure 3: Performance when predicting in the end of games, using the full feature vector.

($p = 0.1492$), and both teams are clearly significantly better than *intermediate* ($p < 6.3 \times 10^{-14}$).

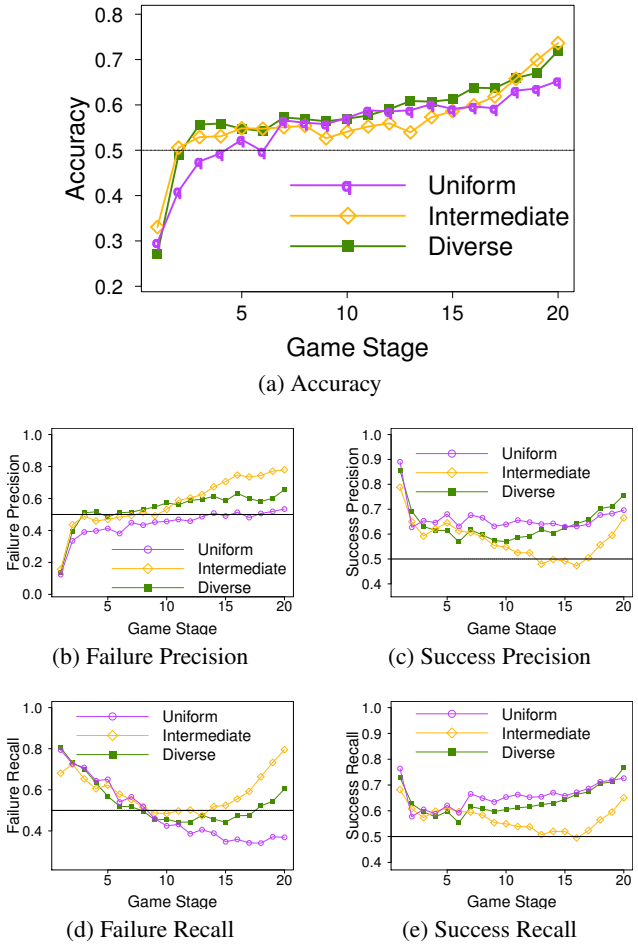
We use a dataset of 691 games for each team. For all results, we used 5-fold cross validation (each fold had approximately the same class ratio as the original distribution). In all graphs, the error bars show the 95% confidence interval.

We start by studying the performance of our prediction in the end of the games (i.e., after the last move). The result is in Figure 3. We could make high-quality predictions for all teams. For *diverse* and *intermediate*, we have around 73% accuracy, while for uniform 64%. This difference is statistically significant, with $p \approx 0.003467$. Concerning failure precision, success precision and failure recall, the prediction for *diverse* is better than for *uniform*, with $p \approx 0.002361$, 3.821×10^{-6} and 3.821×10^{-6}, respectively.

It is also interesting to note that although *intermediate* is significantly weaker than *uniform*, we could achieve a higher accuracy for *intermediate* (with $p \approx 0.00379$). We would expect, however, to make better predictions for *diverse* than *intermediate*, but they have very similar accuracy results in the end. However, by analyzing the other metrics, we can notice that the prediction for "Failure" is better for *intermediate*, while the one for "Success" is better for *diverse*.

As we could see, with absolutely no data about which specific actions were made and which specific world states were encountered, we are able to predict the outcome of the games with high accuracy for all the 3 teams, with better results for *diverse* than *uniform*, even though these two teams have similar winning rates.

We also ran our classifier at every turn of the games. In order to verify the predictions, we used the evaluation of the original Fuego, but we give it a time limit 50× longer. Since this version is approximating a perfect evaluation of a board configuration, we will refer to it as "Perfect". We, then, use Perfect's evaluation of a given board state to estimate its probability of victory, allowing a comparison with our approach. Considering that an evaluation above 0.5 is "success" and below is "failure", we compare our predictions with the ones given by Perfect's evaluation, at each turn of the games. We use this method because a team could be "winning" at a certain stage, but change to "losing" after making a mistake (or vice-versa after an opponent's mistake). Therefore, simply comparing with the final outcome



(a) Accuracy

(b) Failure Precision

(c) Success Precision

(d) Failure Recall

(e) Success Recall

Figure 4: Performance metrics over all turns of 691 games, using the full feature vector.

of the game would not be accurate.

We can see the result in Figure 4. Since the games have different length, we divide all games in 20 stages, and show the average evaluation of each stage. Therefore, a stage is defined as a small set of turns (on average, 2.43 ± 0.5 turns). We were able to obtain a high-accuracy, already crossing the 0.5 line in the 3rd stage. From around the middle of the games (stage 10), the accuracy for *diverse* and *uniform* already gets close to 60% (with *intermediate* only close behind). Although we can see some small drops, overall the accuracy increases with the game stage number, as expected. Moreover, for most of the stages, the accuracy is higher for *diverse* than for *uniform*. The prediction for *diverse* is significantly better than for *uniform* (with $p < 0.1$) in 25% of the stages.

We also run experiments using the reduced feature vector for all teams. In Figure 5 we can see the results when predicting in the end of the games. The accuracy does not change much for *diverse* and *intermediate* (when comparing against the accuracy in both teams using the full feature vector), and the difference is not significant ($p = 0.9929$ and $p = 0.8403$ for *diverse* and *intermediate*). For *uniform* we

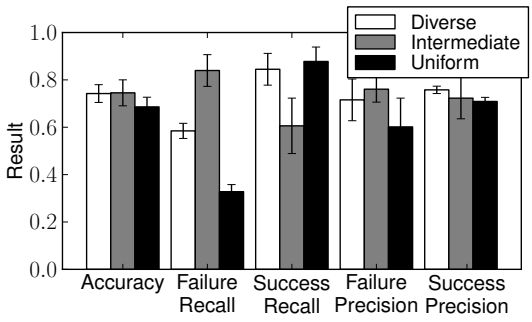


Figure 5: Performance when predicting in the end of games, using the reduced feature vector.

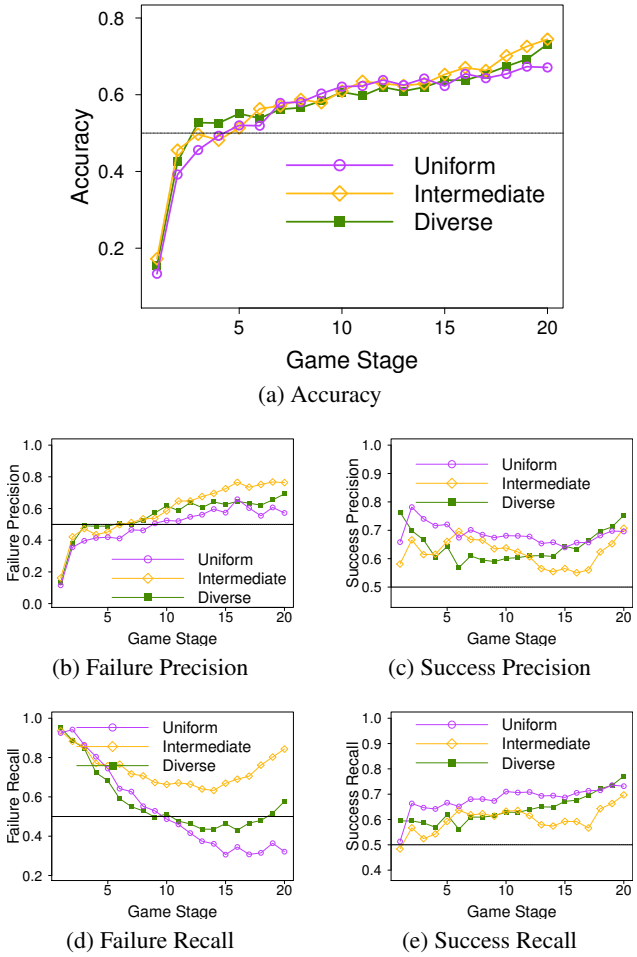
observe an improvement in the accuracy of 4% (which is not statistically significant, $p = 0.2867$). In Figure 6 we see the prediction at each stage of the games, again comparing with Perfect's evaluation. As we can see, we also obtain a high accuracy quickly with the reduced feature vector, reaching 60% again towards the middle of the games. This time, there is less difference in the accuracy between *diverse* and *uniform*, but we can still show that *diverse* is significantly better than *uniform* (with $p < 0.1$) in 15% of the stages (20% including a stage where $p \approx 0.1$). Again, the accuracy for the *intermediate* team is close to the one for *uniform*, even though *intermediate* is a significantly weaker team.

As we can see, for all teams and both feature vectors, our predictions match Perfect's evaluation roughly 60% of the time. However, our method is much faster, since it only requires one linear calculation that takes a few microseconds, while Perfect's evaluation takes a few minutes.

Discussion

We show in this work, both theoretically and experimentally, that we can make high-quality predictions about the performance of a team of voting agents, using only information about the frequency of agreements among agents. We present two kinds of feature vectors, one that includes information about which specific agents were involved in an agreement and one that only uses information about how many agents agreed together. Although the number of features in the former increases exponentially with the number of agents, causing scalability concerns, the latter representation scales better as it increases linearly. Theoretically the full feature vector should have better results, but in our experiments both approaches achieved a high accuracy. Hence, for large teams we can use the reduced feature vector, avoiding scalability problems.

Moreover, in real applications we usually do not have extremely large teams of voting agents. Unless we have an idealized diverse team, the performance is expected to converge after a certain number of agents (Jiang et al. 2014). In Marcolino, Jiang, and Tambe (2013) and Marcolino et al. (2014), significant improvements are already obtained with only 6 agents, while Jiang et al. (2014) shows little improvement as teams grows larger than 15 agents. Therefore, the scalability of the feature vector might not be a real concern.



(a) Accuracy

(b) Failure Precision

(c) Success Precision

(d) Failure Recall

(e) Success Recall

Figure 6: Performance metrics over all turns of 691 games, using the reduced feature vector.

Based on classical voting theory, we would expect the predictions to work better in the *uniform* team (or at least as well as for the *diverse* team). However, we show in our experiments that the prediction works significantly better for the *diverse* team, and we present a preliminary theoretical model to explain this phenomenon. It is also surprising that the prediction for *intermediate* works as well as for the other teams, even though it is significantly weaker (i.e., it has a much lower wining rate). We would expect at least that the prediction in *diverse* should be better than in *intermediate*, since *diverse* is both stronger and has higher diversity. Understanding why this happens is still open, and an immediate next step is to quantify the amount of diversity (for example, as defined in Marcolino, Jiang, and Tambe (2013)) in all teams to better understand this phenomenon.

Although we showed a great performance in prediction, what an operator should actually do as the prediction of failure goes high is not discussed in this paper. Possible remedy procedures vary according to each domain. For example, for a complex problem being solved in a cluster of computers, we could allocate a higher number of resources when

it becomes necessary. Preliminary experiments in Computer Go show that we can actually obtain a high winning rate if games are re-started when the prediction of failure is high. Finding ways to recover the situation in Computer Go games (such as dynamically changing the team or the voting rule) is an interesting avenue for future work.

Finally, as our approach is domain independent, it would be interesting to test the prediction in different domains, such as the variety of games available in the arcade learning environment. For such games, it is also interesting to come up with coping strategies for when the prediction of failure goes high, to see how we can improve the performance of actually playing these games.

Conclusion

Voting is a widely applied domain independent technique in machine learning. We present a novel method to predict the performance of a team of agents that vote together at every step of a complex problem. Our method does not use any domain knowledge and is based only on the frequencies of agreement among the agents of the team. We present a preliminary theoretical work that explains why our method works, besides showing that the prediction should work better in teams composed by different agents (diverse teams). We perform experiments in the Computer Go domain with 3 different teams, where we show that we can achieve a high accuracy in diverse teams, even when doing the prediction in a particular stage of the game (instead of in the end of the problem solving process), allowing an operator to take remedy procedures if the team is not performing well. The accuracy for the uniform team, although significantly lower than for the diverse team at some stages, is also high.

Acknowledgments: This research was supported by MURI grant W911NF-11-1-0332, and by IUSSTF.

References

Baudiš, P., and Gailly, J.-l. 2011. Pachi: State of the Art Open Source Go Program. In *Advances in Computer Games 13*.

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research* 47:253–279.

Bulling, N.; Dastani, M.; and Knobbout, M. 2013. Monitoring norm violations in multi-agent systems. In *AAMAS*.

Conitzer, V., and Sandholm, T. 2005. Common voting rules as maximum likelihood estimators. In *UAI*, 145–152. Morgan Kaufmann Publishers.

Doan, T. T.; Yao, Y.; Alechina, N.; and Logan, B. 2014. Verifying heterogeneous multi-agent programs. In *AAMAS*.

Enzenberger, M.; Müller, M.; Arneson, B.; and Segal, R. 2010. Fuego - An open-source framework for board games and go engine based on Monte Carlo Tree Search. *IEEE Transactions on Computational Intelligence and AI in Games* 2(4):259 –270.

Gelly, S.; Wang, Y.; Munos, R.; and Teytaud, O. 2006. Modification of UCT with patterns in Monte-Carlo Go. Technical report, Institut National de Recherche en Informatique et en Automatique.

Genesereth, M. R.; Love, N.; and Pell, B. 2005. General game playing: Overview of the AAAI competition. *AI Magazine* 26(2):62–72.

Hunter, J.; Raimondi, F.; Rungta, N.; and Stocker, R. 2013. A synergistic and extensible framework for multi-agent system verification. In *AAMAS*.

Hutter, M. 2005. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer.

Jiang, A. X.; Marcolino, L. S.; Procaccia, A. D.; Sandholm, T.; Shah, N.; and Tambe, M. 2014. Diverse randomized agents vote to win. In *NIPS*.

Khalastchi, E.; Kalech, M.; and Rokach, L. 2014. A hybrid approach for fault detection in autonomous physical agents. In *AAMAS*.

Kouvaros, P., and Lomuscio, A. 2013. Automatic verification of parameterised interleaved multi-agent systems. In *AAMAS*.

Legg, S. 2008. *Machine Super Intelligence*. Ph.D. Dissertation, University of Lugano.

Lindner, M. Q., and Agmon, N. 2014. Effective, quantitative, obscured observation-based fault detection in multi-agent systems. In *AAMAS*.

List, C., and Goodin, R. E. 2001. Epistemic democracy: Generalizing the Condorcet Jury Theorem. *Journal of Political Philosophy* 9:277–306.

Mao, A.; Procaccia, A. D.; and Chen, Y. 2013. Better Human Computation Through Principled Voting. In *AAAI*.

Marcolino, L. S.; Xu, H.; Jiang, A. X.; Tambe, M.; and Bowring, E. 2014. Give a hard problem to a diverse team: Exploring large action spaces. In *AAAI*.

Marcolino, L. S.; Jiang, A. X.; and Tambe, M. 2013. Multi-agent team formation: Diversity beats strength? In *IJCAI*.

Mirchevska, V.; Luštrek, M.; Bežek, A.; and Gams, M. 2014. Discovering strategic behaviour of multi-agent systems in adversary settings. *Computing and Informatics*.

Polikar, R. 2012. *Ensemble Machine Learning: Methods and Applications*. Springer. Chapter: Ensemble Learning.

Raines, T.; Tambe, M.; and Marsella, S. 2000. Automated assistants to aid humans in understanding team behaviors. In *AGENTS*.

Ramos, F., and Ayanegui, H. 2008. Discovering tactical behavior patterns supported by topological structures in soccer-agent domains. In *AAMAS*.

Tarapore, D.; Christensen, A. L.; Lima, P. U.; and Carneiro, J. 2013. Abnormality detection in multiagent systems inspired by the adaptive immune system. In *AAMAS*.

Torrey, L., and Taylor, M. E. 2013. Teaching on a budget: Agents advising agents in reinforcement learning. In *AAMAS*.

Zhang, C., and Lesser, V. 2013. Coordinating multi-agent reinforcement learning with limited communication. In *AAMAS*.