Every Team Makes Mistakes, in Large Action Spaces

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

Abstract

Voting is applied to better estimate an optimal answer to complex problems in many domains. We recently presented a novel benefit of voting, that has not been observed before: we can use the voting patterns to assess the performance of a team and predict whether it will be successful or not in problem-solving. Our prediction technique is completely domain independent, and it can be executed at any time during problem solving. In this paper we present a novel result about our technique: we show that the prediction quality increases with the size of the action space. We present a theoretical explanation for such phenomenon, and experiments in Computer Go with a variety of board sizes.

1 Introduction

Voting has been applied in many important domains, such as machine learning [Polikar, 2012], crowdsourcing [Mao *et al.*, 2013; Bachrach *et al.*, 2012], and even board games [Marcolino *et al.*, 2014; Obata *et al.*, 2011]. Voting provides theoretical guarantees, and it is an aggregation approach that is very suited for wide applicability. However, a team of voting agents will not always be successful in problem-solving. It is very important, hence, to be able to assess quickly the performance of teams, in order to be able to take actions to recover the situation in time. Moreover, complex problems are generally characterized by a large action space, and hence methods that work well in such situations are of particular interest.

Current works in the multi-agent system literature focus on identifying faulty or erroneous behavior [Khalastchi *et al.*, 2014; Lindner and Agmon, 2014], or verifying correctness [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 *et al.* [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].

Many works in robotics propose monitoring a team by detecting differences in the internal state of the agents (or disagreements), mostly caused by malfunction of the sensors/actuators [Kalech and Kaminka, 2007; 2011]. In a system of voting agents, however, disagreements are inherent in the coordination process and do not necessarily mean that an erroneous situation has occurred due to such malfunction. Meanwhile, the works in social choice are mostly focused on studying the guarantees of finding the optimal choice given a noise model for the agents [Conitzer and Sandholm, 2005], but provide no help in assessing the performance of a team.

We recently introduced a novel method to predict the final performance (success or failure) of a team of voting agents, without using any domain knowledge [Nagarajan *et al.*, 2015]. Our method can be applied in a great variety of scenarios, and it can be quickly used online at any step of problem-solving. This is fundamental in many applications. For example, consider a problem being solved in a cluster of computers. It is undesirable to allocate more resources than necessary, but if we notice that a team is failing, we can increase the allocation of resources. Or consider a team playing together a game against an opponent (such as board games, or poker). Different teams might play better against different opponents. Hence, if we notice that a team is having issues, we could dynamically change it. Under time constraints, however, such prediction must be executed quickly.

However, Nagarajan *et al.* [2015] only presented results with a fixed action space. Hence, it was not clear how the prediction quality would change for different problems. Moreover, Nagarajan *et al.* [2015] only considered a fixed threshold for the classification, and the impact of using different ones was never studied. Hence, in this paper we present: (i) a novel theoretical study that shows that we can make better predictions about the team performance in large action spaces; (ii) extensive new experiments covering 4 different board sizes in Computer Go and 3 different teams; (iii) new experimental evaluations using ROC curves that not only show experimentally how the performance of our prediction changes in larger action spaces, but also shows better (than what was done in Nagarajan *et al.* [2015]) the difference in prediction quality for diverse and uniform teams.

2 Related Work

Voting is a technique that can be applied in many different domains, such as: crowdsourcing [Mao *et al.*, 2013; Bachrach *et al.*, 2012], board games [Marcolino *et al.*, 2013;

2014; Obata *et al.*, 2011], machine learning [Polikar, 2012], forecasting systems [Isa *et al.*, 2010], etc. It is fundamental, hence, to be able to assess the performance of a voting team.

Traditional methods of team assessment rely heavily on tailoring for specific domains. Raines *et al.* [2000] present a method to build 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.

In the multi-agent systems community, we can see many recent works that study how to identify agents that present faulty behavior [Khalastchi *et al.*, 2014; Lindner and Agmon, 2014]. Other works focus on verifying correct agent implementation [Doan *et al.*, 2014] or monitoring the violation of norms in an agent system [Bulling *et al.*, 2013]. However, a team can still have a poor performance and fail in solving a problem, even when the individual agents are correctly implemented and no agent presents faulty behavior.

Sometimes even correct agents might fail to solve a task, especially embodied agents (robots) that could suffer sensing or actuating problems. Kaminka and Tambe [1998] present a method to detect clear failures in an agent team by social comparison (i.e., each agent compares its state with its peers). Such an approach is fundamentally different than our work, as we are detecting a tendency towards failure for a team of voting agents (caused, for example, by simple lack of ability, or processing power, to solve the problem), not a clearly problematic situation that could be caused by imprecision/failure of the sensors or actuators of an agent/robot. Later, Kalech and Kaminka [2011] study the detection of failures by identifying disagreement among the agents. In our case, however, disagreements are inherent in the voting process. They are easy to detect but they do not necessarily mean that a team is immediately failing, or that an agent presents faulty behavior/perception of the current state.

Finally, it has recently been shown that diverse teams of voting agents are able to outperform uniform teams composed of copies of the best agent [Marcolino *et al.*, 2013; 2014; Jiang *et al.*, 2014]. In Nagarajan *et al.* [2015] we presented an extra benefit of having diverse teams: we showed that we can make better predictions of the final performance for diverse teams than for uniform teams. In this paper we study such claim more extensively in Computer Go experiments than what was done before. Moreover, Marcolino *et al.* [2014] showed that the performance of diverse teams increases as the action space grows. Here we build on the model of Marcolino *et al.* [2014] to show that the prediction quality for such teams also increases with the action space.

3 Prediction Method

Before introducing our novel theoretical work, we start by revisiting the prediction method proposed in Nagarajan *et al.* [2015], in order for this paper to be fully comprehensible.

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 $\mathbf{S}$ be a set of world states s_k. The agents must vote for an action at each world state, and the team takes the action decided by the *plurality voting rule*, that picks the action that received the highest number of votes (we assume 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 remedial 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 $\mathcal{P}(\mathbf{T}) = \{\mathbf{T_1}, \mathbf{T_2}, \ldots\}$ be the power set of the set of agents, a_i be the action chosen in world state s_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 $\vec{\mathbf{x}} = (x_1, x_2, \ldots)$ computed at world state s_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, $x_i = \sum_{k=1}^{|\mathbf{S_j}|} \frac{\mathbb{I}(\mathbf{H_k} = \mathbf{T_i})}{|\mathbf{S_j}|}$, where $\mathbb{I}$ is the indicator function and $\mathbf{S_j} \subseteq \mathbf{S}$ is the set of world states from s_1 to the current world state s_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 a certain threshold as "success", and below it 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. 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, we also drop the feature corresponding to all agents agreeing on the chosen action.

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 $\vec{\mathbf{y}} = (y_1, y_2, \ldots)$, where we define y_i to be the proportion of times that the chosen action was agreed upon by any subset of i agents:$y_i = \sum_{k=1}^{|\mathbf{S_j}|} \frac{\mathbb{I}(|\mathbf{H_k}| = i)}{|\mathbf{S_j}|}$, where $\mathbb{I}$ is the indicator function and $\mathbf{S_j} \subseteq \mathbf{S}$ is the set of world states from s_1 to the current world state s_j. We compare the two approaches in Section 5.

4 Theory

4.1 Prediction Theory

We first present the main theoretical model of Nagarajan *et al.* [2015], as we build over it to develop our new result in the

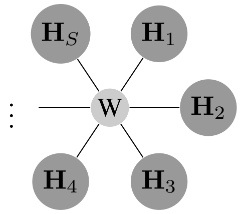


Figure 1: Graphical model of the problem solving process across a series of voting iterations.

next section. We consider agents voting across multiple world states. We assume that all iterations equally influence the final outcome, and that they are all independent. Let the final reward of the team be defined by a random variable W, and let the number of world states be S. We model the problem solving process by the graphical model in Figure 1, where $\mathbf{H}_j$ is the instance of a random variable that represents the subset of agents that agreed on the chosen action at world state s_j.

Of course a specific problem (for example, Go games where the next state will depend on the action taken in the current one) would call for more complex models to be completely represented. Our model is a simplification of the problem solving process, abstracting away the details of specific problems for a greater *generality*.

For any subset $\mathbf{H}$, let $P(\mathbf{H})$ be the probability that the chosen action was correct given the subset of agreeing agents. $P(\mathbf{H})$ depends on both the team and the world state. However, we marginalize the probabilities to produce a value that is an average over all world states. We consider that, for a team to be successful, there exists a unique δ such that:

$$\left\{ \prod_{j=1}^{S} P(\mathbf{H}_j) \right\}^{1/S} > \delta \qquad (1)$$

We use the exponent $1/S$ in order to maintain a uniform scale across all problems. Each problem may have a different number of world states; and for one with many world states, it is likely that the incurred product of probabilities is sufficiently low to fail the above test, independent of the actual subsets of agents that agreed upon the actions. However, the final reward is not dependent on the number of world states.

We show, then, that we can use a linear classification model (such as logistic regression) that is equivalent to Equation 1, to predict the final reward of a team.

Theorem 1 *Given the model in Equation 1, the final outcome of a team can be predicted by a linear model.*

Proof Getting the log in both sides of Equation 1, we have $\sum_{j=1}^{S} \frac{1}{S} \log(P(\mathbf{H}_j)) > \log(\delta)$. The sum over the steps (world states) of the problem-solving process can be transformed to a sum over all possible subset of agents that can be encountered, $\mathcal{P}$: $\sum_{\mathbf{H}\in\mathcal{P}} \frac{n_{\mathbf{H}}}{S} \log(P(\mathbf{H})) > \log(\delta)$, where $n_{\mathbf{H}}$ is the number of times the subset of agreeing agents $\mathbf{H}$ was encountered during problem solving. Hence, $\frac{n_{\mathbf{H}}}{S}$ is the frequency of seeing the subset $\mathbf{H}$, which we denote by $f_{\mathbf{H}}$.

Recall that $\mathbf{T}$ is the set of all agents. Hence, $f_{\mathbf{T}}$ (which is the frequency of all agents agreeing on the same action), is equal to $1 - \sum_{\mathbf{H}\in\mathcal{P}\setminus\{\mathbf{T}\}} f_{\mathbf{H}}$. Also, note that $n_{\emptyset} = 0$, since at least one agent must pick the chosen action. The above equation can, hence, be rewritten as $\log(P(\mathbf{T})) + \sum_{\mathbf{H}\in\mathcal{P}\setminus\mathbf{T}} f_{\mathbf{H}} \log\left(\frac{P(\mathbf{H})}{P(\mathbf{T})}\right) > \log(\delta)$. Hence, our final model will be:

$$\sum_{\mathbf{H}\in\mathcal{P}\setminus\mathbf{T}} \log\left(\frac{P(\mathbf{H})}{P(\mathbf{T})}\right) f_{\mathbf{H}} > \log\left(\frac{\delta}{P(\mathbf{T})}\right) \qquad (2)$$

Note that $\log(\frac{\delta}{P(\mathbf{T})})$ and the "coefficients" $\log(\frac{P(\mathbf{H})}{P(\mathbf{T})})$ are all *constants* with respect to a given team, as we have discussed earlier. Considering the set of all $f_{\mathbf{H}}$ (for each possible subset of agreeing agents $\mathbf{H}$) to be the characteristic features of a single problem, the coefficients can now be *learned* from training data that contains many problems represented using these features. Further, the outcome of a team can be estimated through a linear model. ■

4.2 Action Space Size

We present now our novel result concerning the quality of the predictions over large action space sizes. In order to perform such study, we assume the *spreading tail* (*ST*) agent model, presented in Marcolino *et al.* [2014]. The basic assumption is that the pdf of each member of the team has a non-zero probability over an increasingly larger number of suboptimal actions as the action space grows, while the probability of voting for the optimal action remains unchanged. Let the size of the action space $|\mathbf{A}| = \varrho$, and $p_{i,j}$ be the probability that agent i votes for action with rank j. Marcolino *et al.* [2014] shows that when $\varrho \rightarrow \infty$, the probability that a team of n *ST* agents will play the optimal action converges to:

$$\tilde{p}_{best} = 1 - \prod_{i=1}^{n}(1-p_{i,0}) - \sum_{i=1}^{n}(p_{i,0}\prod_{j=1,j\neq i}^{n}(1-p_{j,0}))\frac{n-1}{n}, \quad (3)$$

that is, the probability of two or more agents agreeing over suboptimal actions converges to zero, and the agents can only agree over the optimal choice (note that a suboptimal action may still be taken when no agent agrees).

Before proceeding to our study, we are going to make a few definitions and then two weak assumptions. We consider now here any action space size. Let α be the probability of a team taking the optimal action when all agents disagree. Since we can only take the optimal action if one agent votes for that action, α is a function of the probability of each agent voting for the optimal action. That is, we may have situations where all agents disagree and no agent voted for the optimal action, or where all agents disagree, but there is one agent that voted for the optimal action (and, hence, we may still take the optimal action due to random tie braking).

Let β be the probability of a team taking the optimal action when there is some agreement on the voting profile. β may be different according to each voting profile, but we assume that we always have that $\beta < 1$ if $\varrho < \infty$, and $\beta = 1$ if

$\varrho \rightarrow \infty$, according to the *ST* agent model. That is, if two or more agents agree, there is always some probability $q > 0$ that they are agreeing over a suboptimal action, and $q \rightarrow 0$ as $\varrho \rightarrow \infty$.

We will make the following weak assumptions: (i) If there is no agreement, the team is more likely to take a suboptimal action than an optimal action. I.e., $\alpha < 1 - \alpha$; (ii) If there is agreement, there is at least one voting profile where the team is more likely to take an optimal action than a suboptimal action. That is, there is at least one β such that $\beta > 1 - \beta$.

Assumption (i) is weak, since $\alpha < 1/n$ (as we break ties randomly and there may be cases where no agent votes for the optimal action). Clearly $1/n < 1 - 1/n$ for $n > 2$. Assumption (ii) is also weak, because if we are given a team that is always more likely to take suboptimal actions than an optimal action for *any* voting profile, then a trivial predictor that always outputs "failure" would be optimal (and, hence, we would not need a prediction at all).

Theorem 2 *The quality of our prediction about the performance of a set of* ST *agents* $\mathbf{T}$ *is the highest as* $\varrho \rightarrow \infty$.

Proof Let's fix the problem to predicting performance at one world state. Hence, as we consider a single decision, there is a single $\mathbf{H}^i$ such that $f_{\mathbf{H}^i} = 1$, and $f_{\mathbf{H}^j} = 0 \, \forall j \neq i$. In order to simplify the notation, we denote by $\mathcal{H}$ the subset $\mathbf{H}^i$ corresponding to $f_{\mathbf{H}^i} = 1$. We also consider the performance of the team as "success" on that fixed world state if they take the optimal action, and as "failure" otherwise.

Let a *voting event* be the process of querying the agents for the vote, obtaining the voting profile and the corresponding final decision. Hence, it has a unique correct label ("success" or "failure"). A voting event ξ will be mapped to a point χ in the feature space, according to the subset of agents that agreed on the chosen action. Multiple voting events, however, will be mapped to the same point χ (as exactly the same subset can agree in different situations). Hence, given a point χ, there is a certain probability that the team was successful, and a certain probability that the team failed. Therefore, by assigning a label to that point, our predictor will also be correct with a certain probability. With enough data, the predictor will output the most likely of the two events. That is, if given a profile, the team has a probability p of taking the optimal action, the probability of the prediction being correct will be $\max(p, 1 - p)$.

We first study the probability of making a correct prediction across the whole feature space, for different action space sizes, and after that we will focus on what happens with the specific voting events as the action space changes.

Let us start by considering the case when $\varrho \rightarrow \infty$. By Equation 3, we know that every time two or more agents agree on the same action, that action will be the optimal one. Note that this is a very clear division of the feature space, as for every single point where $|\mathcal{H}| \geq 2$ the team will be successful with probability 1. Therefore, on this subspace we can make perfect predictions. The only points in the feature space where a team may still take a suboptimal action are the ones where a single agent agrees on the chosen action, i.e., $|\mathcal{H}| = 1$. Hence, for such points we will make a correct prediction with probability $\max(\alpha, 1 - \alpha)$.

Let's now consider cases with $\varrho < \infty$. Let's first consider the subspace $|\mathcal{H}| \geq 2$. Before, our predictor was correct with probability 1. Now, given a voting event where there is an agreement, there will be a probability $\beta < 1$ of the team taking the optimal action. Hence, the predictor will be correct with probability $\max(\beta,1 - \beta)$, but $\max(\beta, 1 - \beta) < 1$.

Let's consider now the subspace $|\mathcal{H}| = 1$. Here the quality of the prediction depends on α, which is a function of the probability of each agent playing the best action. On the *ST* agent model, however, the probability of one agent voting for the best action is independent of ϱ [Marcolino *et al.*, 2014]. Hence, α does not depend on the action space size, and for these cases the quality of our prediction will be the same as before. Therefore, for all points in the feature space, the probability of making a correct prediction is either the same or worse when $\varrho < \infty$ than when $\varrho \rightarrow \infty$.

However, that does not complete the proof yet, because a voting event ξ may map to a different point χ when the action space changes. For instance, the number of agents that agree over a suboptimal action may overpass the number of agents that agree on the optimal action as the action spaces changes from $\varrho \rightarrow \infty$ to $\varrho < \infty$. Therefore, we need to show that our prediction will be strictly better when $\varrho \rightarrow \infty$ irrespective of such mapping. Hence, let us now study the voting events. As the number of actions decrease, a certain voting event ξ when $\varrho \rightarrow \infty$, will map to a voting event ξ' when $\varrho < \infty$. Let χ and χ' be the corresponding points in the feature space for ξ and ξ'. Also, let $\mathcal{H}$ and $\mathcal{H}'$ be the respective subset of agreeing agents. Let's consider now the four possible cases:

(i) $|\mathcal{H}| = |\mathcal{H}'| = 1$. For such events, the performance of the predictor will remain the same, that is, for both cases we will make a correct prediction with probability $\max(\alpha, 1-\alpha)$. Note that this case will not happen for all events, as $p_{i,0} \not\rightarrow 0$ when $\varrho \rightarrow \infty$, hence there will be at least one event where $|\mathcal{H}| \geq 2$.

(ii) $|\mathcal{H}| \geq 2$, $|\mathcal{H}'| \geq 2$. For such events the performance of the predictor will be higher when $\varrho \rightarrow \infty$, as we can make a correct prediction for a point χ with probability 1, while for a point χ' with probability $\max(\beta, 1 - \beta) < 1$.

(iii) $|\mathcal{H}| \geq 2$, $|\mathcal{H}'| = 1$. This case will not happen under the *ST* agent model. If there was a certain subset $\mathbf{H}$ of agreeing agents when $\varrho \rightarrow \infty$, when we decrease the number of actions the new subset of agreeing agents $\mathbf{H}'$ will either have the same size or will be larger. This follows from the fact that we may have a larger subset agreeing over some suboptimal action when the action space decreases, but the original subset that voted for the optimal action will not change.

(iv) $|\mathcal{H}| = 1$, $|\mathcal{H}'| \geq 2$. We know that in this case ξ' is an event where the team fails (otherwise the same subset would also have agreed when $\varrho \rightarrow \infty$). Hence, for $1 - \alpha > \alpha$ (weak assumption (i)), we make a correct prediction for such case when $\varrho \rightarrow \infty$. When $\varrho < \infty$, we make a correct prediction if $1 - \beta > \beta$. β, however, depends on the voting profile of the event ξ'. By weak assumption (ii), there will be at least one event where the team is more likely to be correct than wrong (that is, $\beta > 1 - \beta$). Hence, there will be at least one event where our predictor changes from making a correct prediction (when $\varrho \rightarrow \infty$) to making an incorrect prediction.

Hence, for all voting events, the probability of making a correct prediction will either be the same or worse when $\varrho < \infty$ than when $\varrho \to \infty$, and there will be at least one voting event where it will be worse, completing the proof. Hence, $\varrho \to \infty$ is strictly the best case for our prediction. As we assume that all world states are independent, if $\varrho \to \infty$ is the best case for a single world state, it will also be the best case for a set of world states. ■

5 Results

We test our prediction method in the Computer Go domain. We use four different Go software: Fuego 1.1 [Enzenberger *et al.*, 2010], GnuGo 3.8 [Free Software Foundation, 2009], 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 six different, publicly available, agents. Fuego is the strongest agent among all of them [Marcolino *et al.*, 2013]. The description of FuegoΔ and FuegoΘ is available in Marcolino *et al.* [2014]. *All the results we present here are novel, and were not shown at Nagarajan* et al. *[2015].*

We study three different teams: *Diverse*, composed of one copy of each agent; *Uniform*, composed of six copies of the original Fuego (initialized with different random seeds); *Intermediate*, composed of six random parametrized versions of Fuego (from Jiang *et al.* [2014]). In all teams, the agents vote together, playing as white, in a series of Go games against the original Fuego playing as black. We study four different board sizes for *diverse* and *uniform*: 9x9, 13x13, 17x17 and 21x21. For *intermediate*, we study only 9x9, since the random parametrizations of Fuego do not work on larger boards.

In order to evaluate our predictions, we use a dataset of 1000 games for each team and board size combination (in a total of 9000 games). For all results, we used repeated random sub-sampling validation. We randomly assign 20% of the games for the testing set (and the rest for the training set), keeping approximately the same ratio as the original distribution. The whole process is repeated 100 times. Hence, in all graphs we show the average results, and the error bars show the 99% confidence interval ($p = 0.01$), according to a *t-test*. Moreover, when we say that a certain result is significantly better than another, we mean statistically significantly better, according to a *t-test* where $p < 0.01$, unless we explicitly give a p value.

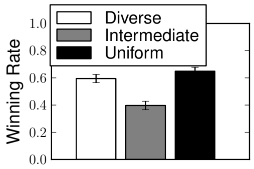
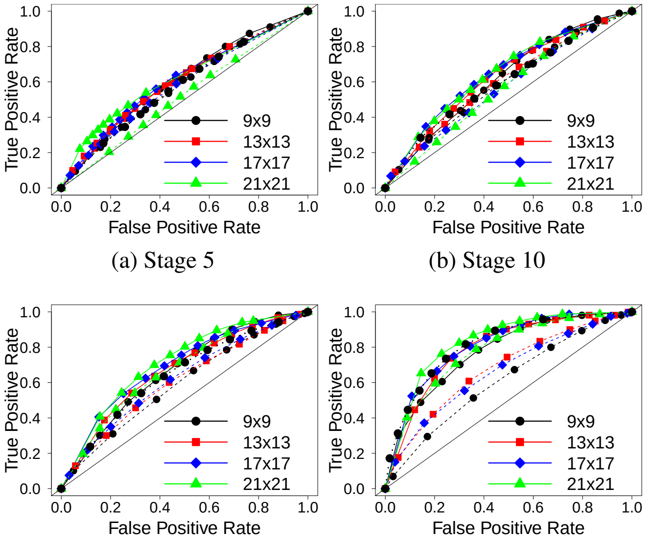


Figure 2: Winning rates of the three teams.

First, we show the winning rates of the teams in Figure 2, in 9x9 Go. *Uniform* is better than *Diverse* with statistical significance ($p = 0.014$), and both teams are clearly significantly better than *Intermediate* ($p < 2.2 \times 10^{-16}$).

In order to verify our online predictions, we used Fuego's evaluation, but we give it a time limit 50× longer. Since this approximates 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 (since the likelihood of victory changes dynamically during a game), 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.



(a) Stage 5

(b) Stage 10

(c) Stage 15

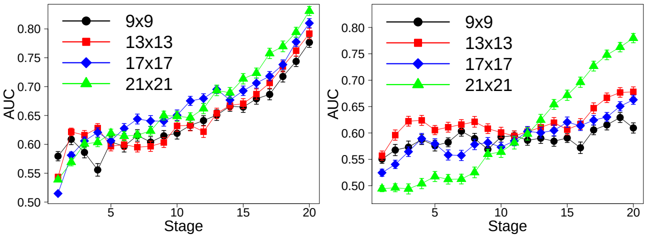
(d) Stage 20

Figure 3: ROC curves for the *diverse* (continuous line), *intermediate* (dotted line), and *uniform* team (dashed line), over a variety of board sizes.

Since the games have different lengths, 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, 1.35 ± 0.32 turns in 9×9; 2.76 ± 0.53 in 13×13; 4.70 ± 0.79 in 17×17; 7.85 ± 0.87 in 21×21). For all games, we also skip the first 4 moves, since our baseline (Perfect) returns corrupted information in the beginning of the games.

We measure our results using receiver operating characteristic (ROC) curves. ROC curves shows the true positive and the false positive rates of a binary classifier at different thresholds (that is, the value above which the output of our prediction function $\hat{f}$ will be considered "success"). We also study the area under the ROC curve (AUC), as a way to synthesize the quality information from the curve into a single number, and compare the different situations. We start by showing the ROC curves for all teams and board sizes in Figure 3.

In Figure 4 we can see the AUC results, with one graph per team, in order to more clearly observe the effect of increasing the size of the action space for each team (we do not show *intermediate* here as it only works for 9x9 Go). For the *diverse* team, we start observing the effect of increasing the action space after stage 5, when the curves for 17×17 and 21×21 tend to dominate the other curves. In fact, the AUC for 17×17 is significantly better than smaller boards in 60% of the stages, and in 80% of the stages after stage 5. Moreover, after stage 5 no smaller board is significantly better than 17×17. Concerning 21×21, we can see that from stage 14, its curve completely dominates all the other curves. In all stages from 14 to 20 the result for 21×21 is significantly better than for all other smaller boards. Hence, we can note



(a) Diverse Team (b) Uniform Team

Figure 4: AUC for different teams and board sizes, by teams.

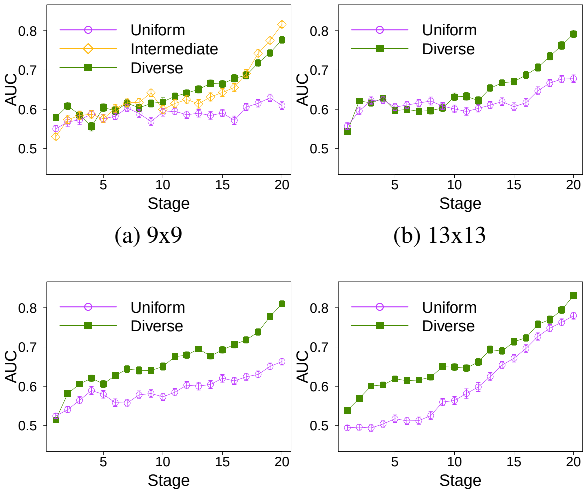
that the effect of increasing the action space seems to depend on the stage of the game.

Concerning the *uniform* team, up to 17×17 we cannot observe a positive impact of the action space size on the prediction quality; but for 21×21 there is clearly an improvement from the middle game when compared with smaller boards. On all 8 stages from stage 13 to stage 20, the result for 21×21 is significantly better than for other board sizes. In terms of percentage of stages where the result for 21×21 is significantly better than for 9×9, we find that it is 40% for the *uniform* team, while it is 85% for the *diverse* team. Hence, the impact of increasing the action space occurs for *diverse* earlier in the game, and over a larger number of stages.

Now, in order to compare the performance for *diverse* and *uniform* under different board sizes, we show the AUCs in Figure 5 organized by the size of the board. It is interesting to observe that the quality of the predictions for *diverse* is better than for *uniform*, irrespective of the size of the action space. Moreover, while for 9×9 and 13×13 the prediction for *diverse* is only always significantly better than for *uniform* after around stage 10, we can notice that for 17×17 and 21×21, the prediction for *diverse* is always significantly better than for *uniform*, irrespective of the stage (except for stage 1 in 17×17). In fact, we can also show that the difference between the teams is greater on larger boards. In Figure 6 we can see the difference between *diverse* and *uniform*, in terms of area under the AUC graph, and also in terms of percentage of stages where *diverse* is significantly better than *uniform*, for 9×9 and 21×21. The difference between the areas in 9×9 and 21×21 is statistically significant, with $p = 0.0003337$.

We also evaluate the accuracy (as it is more intuitively understandable) in the last stage, with a prediction threshold of 0.5. We obtain 71% for *diverse* in 9×9, and 81% in 21×21. For *uniform*, we obtain 62% in 9×9, and 75% in 21×21.

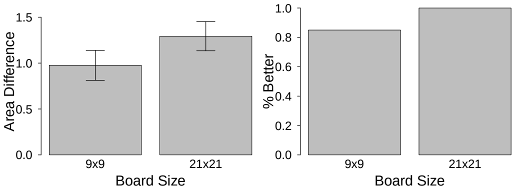
We evaluate the reduced feature vector as well. The results are similar to the ones using the full feature vector, but with a much more scalable representation. In fact, in Figure 7 we study the area under the AUC graphs of the full and reduced representations. The reduced representation is actually statistically significantly better for all teams on the 9×9 and 13×13 boards. For *diverse*, however, the importance of the full representation increases as the action space gets larger. On 17×17, the difference between the representations is not statistically significant ($p = 0.9156$), while on 21×21 the full representation is significantly better than the reduced one.



(a) 9x9 (b) 13x13

(c) 17x17 (d) 21x21

Figure 5: AUC for different teams and board sizes, by board sizes.



(a) By area (b) By percentage of stages

Figure 6: Differences in prediction quality for the *diverse* and *uniform* teams.

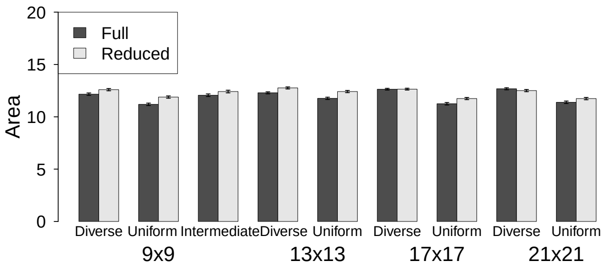


Figure 7: Comparison of prediction quality with the full and reduced representation.

6 Conclusion

We study the effect of increasing the action space size on the team prediction technique shown in Nagarajan *et al.* [2015]. Our theory shows that the prediction quality increases in larger action spaces, and the experimental results confirm such phenomenon. Moreover, we present a detailed study on the prediction quality for different kinds of teams, using ROC curves to analyze different thresholds.

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

References

[Bachrach *et al.*, 2012] Yoram Bachrach, Thore Graepel, Gjergji Kasneci, Michal Kosinski, and Jurgen Van Gael. Crowd IQ: aggregating opinions to boost performance. In *AAMAS*, pages 535–542, 2012.

[Baudiš and Gailly, 2011] Petr Baudiš and Jean-loup Gailly. Pachi: State of the Art Open Source Go Program. In *Advances in Computer Games 13*, November 2011.

[Bulling *et al.*, 2013] Nils Bulling, Mehdi Dastani, and Max Knobbout. Monitoring norm violations in multi-agent systems. In *AAMAS*, 2013.

[Conitzer and Sandholm, 2005] Vincent Conitzer and Tuomas Sandholm. Common voting rules as maximum likelihood estimators. In *UAI*, pages 145–152. Morgan Kaufmann Publishers, 2005.

[Doan *et al.*, 2014] Thu Trang Doan, Yuan Yao, Natasha Alechina, and Brian Logan. Verifying heterogeneous multi-agent programs. In *AAMAS*, 2014.

[Enzenberger *et al.*, 2010] M. Enzenberger, M. Müller, B. Arneson, and R. Segal. 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, Dec. 2010.

[Free Software Foundation, 2009] Free Software Foundation. Gnugo. `http://www.gnu.org/software/gnugo/`, 2009.

[Gelly *et al.*, 2006] Sylvain Gelly, Yizao Wang, Rémi Munos, and Olivier Teytaud. Modification of UCT with patterns in Monte-Carlo Go. Technical report, Institut National de Recherche en Informatique et en Automatique, 2006.

[Isa *et al.*, 2010] Iza Sazanita Isa, S. Omar, Z. Saad, N.M. Noor, and M.K. Osman. Weather forecasting using photovoltaic system and neural network. In *Proceedings of the Second International Conference on Computational Intelligence, Communication Systems and Networks*, CICSyN, pages 96–100, July 2010.

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

[Kalech and Kaminka, 2007] Meir Kalech and Gal A. Kaminka. On the design of coordination diagnosis algorithms for teams of situated agents. *Artificial Intelligence*, 171:491–513, 2007.

[Kalech and Kaminka, 2011] Meir Kalech and Gal A. Kaminka. Coordination diagnostic algorithms for teams of situated agents: Scaling-up. *Computational Intelligence*, 27(3):393–421, 2011.

[Kaminka and Tambe, 1998] Gal A. Kaminka and Milind Tambe. What is wrong with us? Improving robustness through social diagnosis. In *AAAI*, 1998.

[Khalastchi *et al.*, 2014] Eliahu Khalastchi, Meir Kalech, and Lior Rokach. A hybrid approach for fault detection in autonomous physical agents. In *AAMAS*, 2014.

[Lindner and Agmon, 2014] Michael Q. Lindner and Noa Agmon. Effective, quantitative, obscured observation-based fault detection in multi-agent systems. In *AAMAS*, 2014.

[Mao *et al.*, 2013] Andrew Mao, Ariel D. Procaccia, and Yiling Chen. Better Human Computation Through Principled Voting. In *AAAI*, 2013.

[Marcolino *et al.*, 2013] Leandro Soriano Marcolino, Albert Xin Jiang, and Milind Tambe. Multi-agent team formation: Diversity beats strength? In *IJCAI*, 2013.

[Marcolino *et al.*, 2014] Leandro Soriano Marcolino, Haifeng Xu, Albert Xin Jiang, Milind Tambe, and Emma Bowring. Give a hard problem to a diverse team: Exploring large action spaces. In *AAAI*, 2014.

[Nagarajan *et al.*, 2015] V. Nagarajan, L. S. Marcolino, and M. Tambe. Every team deserves a second chance: Identifying when things go wrong. In *AAMAS*, 2015.

[Obata *et al.*, 2011] Takuya Obata, Takuya Sugiyama, Kunihito Hoki, and Takeshi Ito. Consultation algorithm for Computer Shogi: Move decisions by majority. In *Computer and Games'10*, volume 6515 of *Lecture Notes in Computer Science*, pages 156–165. Springer, 2011.

[Polikar, 2012] Robi Polikar. *Ensemble Machine Learning: Methods and Applications*, chapter Ensemble Learning. Springer, 2012.

[Raines *et al.*, 2000] T Raines, Milind Tambe, and S Marsella. Automated assistants to aid humans in understanding team behaviors. In *AGENTS*, 2000.

[Ramos and Ayanegui, 2008] Fernando Ramos and Huberto Ayanegui. Discovering tactical behavior patterns supported by topological structures in soccer-agent domains. In *AAMAS*, 2008.