Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Every Team Deserves a Second Chance: Identifying When Things Go Wrong (Student Abstract Version)

Vaishnavh Nagarajan[1]**, Leandro Soriano Marcolino**[2]**, Milind Tambe**[2]
[1] Indian Institute of Technology Madras, Chennai, Tamil Nadu, 600036, India, +91-44-22574350
http://www.cse.iitm.ac.in/~vaish/
vaish@cse.iitm.ac.in
[2] University of Southern California, Los Angeles, CA, 90089, USA
{sorianom,tambe}@usc.edu

Abstract

We show that without using any domain knowledge, we can predict the final performance of a team of voting agents, at any step towards solving a complex problem.

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). However, a team will not always be successful. It is fundamental, therefore, to be able to quickly assess the performance of teams, so that an operator can take actions to recover the situation in time.

Raines, Tambe, and Marsella (2000) present a method to analyze the performance of a team. However, it only works offline and needs domain knowledge. Other methods are 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 an operator to identify when the team is failing. We present experimental results in the Computer Go domain, where we predict the performance of three different teams: a diverse, a uniform, and an intermediate team (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. 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.

Prediction Method

We consider scenarios where agents vote at every step of a complex problem, in order to take common decisions towards problem-solving. Hence, let $\mathbf{T}$ be a set of agents t_i, $\mathbf{A}$ a set of actions a_j and $\mathbf{M}$ a set of world states m_k. The agents vote for an action at each world state, and the team takes the action decided by *plurality voting* (ties are broken randomly). The team obtains a final reward r upon completing all world states. 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 an 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 $\tilde{\mathbf{X}}$, where 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 $\hat{r}$ ($0 \leq \hat{r} \leq 1$) 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, we use the features at the final world state. We also study a variant, that uses only information about the number of agents that agreed (see appendix at teamcore.usc.edu/people/sorianom/aaai15-ap.pdf).

We use classification by logistic regression, which models $\hat{f}$ as $\hat{f}(\vec{\mathbf{x}}) = \frac{1}{1+e^{-(\alpha+\vec{\boldsymbol{\beta}}^T\vec{\mathbf{x}})}}$ (α and $\vec{\boldsymbol{\beta}}$ are learned given $\tilde{\mathbf{X}}$ and the associated rewards). We eliminate two features: $\emptyset$, as an action is chosen only if at least one agent voted for it, and $\mathbf{T}$ (all agents agree), since the features are linearly dependent.

Results

We test our prediction method in the Computer Go domain. We use 4 different Go software: Fuego 1.1, GnuGo 3.8, Pachi 9.01, MoGo 4, and two variants of Fuego (FuegoΔ and FuegoΘ), in a total of 6 different agents. As shown in

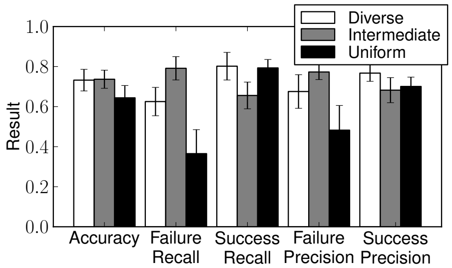


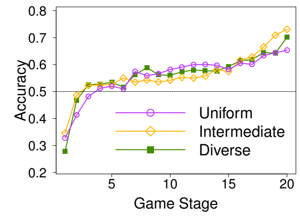
Figure 1: Performance when predicting in the end of games.

Marcolino, Jiang, and Tambe (2013), Fuego is the strongest one. We study three 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 9x9 games against the original Fuego playing as black. *Uniform* is the strongest team (64% winning rate), followed by *diverse* (60%) and *intermediate* (40%). We use a dataset of 691 games for each team, and evaluate the predictions with 5-fold cross validation.

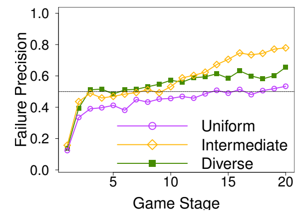
We start by studying our predictions after the end of the games. The result is in Figure 1 (error bars show the 95% confidence interval). We could make high-quality predictions for all teams. For *diverse* and *intermediate*, we have around 73% accuracy, while for *uniform*, 64%. This difference is statistically significant, with $p \approx 0.003467$. It is also interesting to note that although *intermediate* is significantly weaker than *uniform*, we could achieve a higher accuracy for *intermediate* (with $p \approx 0.00379$). Hence, it is not the case that we can make better predictions for stronger teams.

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

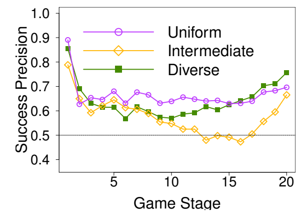
However, predictions made at the end of the problem solving process are not useful. Our objective is to get high-quality predictions at any stage. Therefore, we ran our classifier at every turn of the games. In order to verify the predictions, we used Fuego's evaluation, but we run it $50\times$ longer (we refer to it as "Perfect") to estimate the probability of victory in a board state, allowing a comparison with our approach. Also, since the games have different lengths, we divide them in 20 stages, and show the average evaluation of each one in Figure 2. We obtained a high accuracy quickly, crossing the 0.5 line in the 3^{rd} stage. In fact, the accuracy is significantly higher than the 0.5 mark (with $p < 0.015$) for all teams from around the 5^{th} stage; and from around the middle of the games, our predictions match Perfect's evaluation roughly 60% of the time. Our method, however, is much faster, since it only requires one linear calculation that takes a few microseconds, while Perfect's evaluation takes a few minutes. Therefore, we can easily use our method online, and dynamically take measures to improve the problem solving process when necessary.



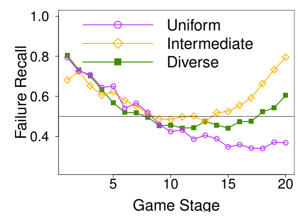
(a) Accuracy



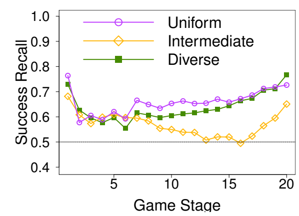
(b) Failure Precision



(c) Success Precision



(d) Failure Recall



(e) Success Recall

Figure 2: Performance metrics over all turns of the games.

Conclusion

We show a novel method to predict the performance of a team of voting agents. Our method does not use domain knowledge and is based only on the frequencies of agreement among the agents. We obtain a high accuracy, even when predicting at each stage of problem solving.

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

References

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*.

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*.

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*.