A Detailed Analysis of a Multi-agent Diverse Team

Leandro Soriano Marcolino, Chao Zhang, Albert Xin Jiang, and Milind Tambe

University of Southern California, Los Angeles, CA, 90089, USA
{sorianom, zhan661, jiangx, tambe}@usc.edu

Appendix

A Diversity Beats Strength

In Section 3 we presented an example with non-deterministic agents that showed that a diverse team can play better than a uniform team made of copies of the strongest agent. The full description of the agents used in the example can be seen in Table 1, where we show the pdf of the agents for each world state. We considered the utility vector $< 1, 0, 0 >$ for all world states.

Agent	State 1	State 2	State 3	State 4	Strength
Agent 1	$< 0.99, 0.01, 0 >$	$< 0, 0.99, 0.01 >$	$< 0.99, 0, 0.01 >$	$< 0.99, 0.01, 0 >$	0.7425
Agent 2	$< 0, 0.99, 0.01 >$	$< 0.99, 0.01, 0 >$	$< 0.99, 0, 0.01 >$	$< 0, 0.01, 0.99 >$	0.4950
Agent 3	$< 0.99, 0.005, 0.005 >$	$< 0.99, 0.005, 0.005 >$	$< 0, 0.5, 0.5 >$	$< 0, 0.5, 0.5 >$	0.4950
Agent 4	$< 0.99, 0.01, 0 >$	$< 0.99, 0.004, 0.006 >$	$< 0, 0.4, 0.6 >$	$< 0.99, 0.003, 0.007 >$	0.7425
Agent 5	$< 0, 0.3, 0.7 >$	$< 0, 0.7, 0.3 >$	$< 0.99, 0.005, 0.005 >$	$< 0.99, 0.002, 0.008 >$	0.4950

Table 1. A team of non-deterministic agents that can overcome copies of the best agent.

B Optimal Voting Rules

We present the derivation of the optimal voting rule, stated in Section 3. If an agent ϕ_i votes for a certain action a_x, the probability of a_x being the action with rank r will be given by p^i_r, the probability that ϕ_i voted for the action with rank r. This will only be true if we assume a uniform prior probability for the ranking of all actions.

Given a certain voting pattern, each possible ranking for the actions in the voting pattern is a mutually exclusive event. Therefore, we sum over all possible ranking combinations where a_x is the best action. The votes of each agent are independent, hence the multiplication of the probabilities, as presented in the paper.

Now, we present here the full proof of Theorem 2, also stated in Section 3.

Proof of Theorem 2 By *Assumption 2* we know that we are looking for a tie-breaking rule, as the action chosen by most of the votes should always be taken. Let's consider the sets and the voting result described in the *Assumption 1*. Let $< p_1, ..., p_k >$ be the pdf of agent $\phi'_{best,j}$, and the pdf of the other agents of the subset be $< p_1 - \epsilon_i, p_2 +$

$\gamma_{i2}, ..., p_k + \gamma_{ik} >$, $\gamma_{il} \geq 0 \; \forall l \in (2, k)$ and $\sum_{l=2}^{k} \gamma_{il} = \epsilon_i$. Let b be a rank in $(2, k)$. The probability of a_x being the best action is given by:

$$P_1 = (p_1) \prod_{i \in Weak} (p_1 - \epsilon_i) \prod_{t \in Strong} (p_b + \gamma_{tb})$$

While the probability that a_y is the best action is given by:

$$P_2 = (p_b) \prod_{i \in Weak} (p_b + \gamma_{ib}) \prod_{t \in Strong} (p_1 - \epsilon_t)$$

By *Assumption 1*, we have that $P_1 > P_2$. We can generate another voting pattern by making one agent ϕ_{weak} in $Weak$ vote for a_y and one agent ϕ_{strong} in $Strong$ vote for a_x. The probability of a_x being the best action will change to:

$$P'_1 = P_1 * \frac{(p_1 - \epsilon_{strong})(p_b + \gamma_{weak,b})}{(p_1 - \epsilon_{weak})(p_b + \gamma_{strong,b})}$$

While the probability of a_y being the best action will change to:

$$P'_2 = P_2 * \frac{(p_1 - \epsilon_{weak})(p_b + \gamma_{strong,b})}{(p_1 - \epsilon_{strong})(p_b + \gamma_{weak,b})}$$

As $(p_1 - \epsilon_{strong}) > (p_1 - \epsilon_{weak})$ and $(p_b + \gamma_{weak,b}) > (p_b + \gamma_{strong,b})$ by the *Assumption 1*, we have that $P'_1 > P_1$. Similarly, as $(p_1 - \epsilon_{weak}) < (p_1 - \epsilon_{strong})$ and $(p_b + \gamma_{strong,b}) < (p_b + \gamma_{weak,b})$ by the *Assumption 1*, we have that $P'_2 < P_2$.

Therefore, assuming that $P_1 > P_2$, we have that $P'_1 > P'_2$. Hence, for all modifications that can be generated by switching one of the agents, it is better to break ties in favor of the strongest agent. We can use all these voting patterns as a base and apply the same process recursively, to generate all possible voting patterns with a tie. Therefore, it will always be better to break ties in favor of the strongest agent.

Now we consider voting patterns with a tie between more than two options. Let's suppose that in this case breaking ties in favor of the strongest agent ($\phi'_{best,j}$) is not the optimal voting rule. Therefore, we should break the tie in favor of some option a_y. This implies that a_y has a higher probability of being the best action than a_x, the option chosen by the best agent. Now let's remove the agents that voted in all other options except a_x and a_y. This affects the probability of a_x and a_y being the best action in the same way. Therefore, we should still break ties in favor of option a_y. However, we already showed that when there are two options we should break ties in favor of the strongest agent. Hence, we should break the tie in favor of option a_x. So, by contradiction, we see that if there is a tie between more than two options we should still break ties in favor of the strongest agent.

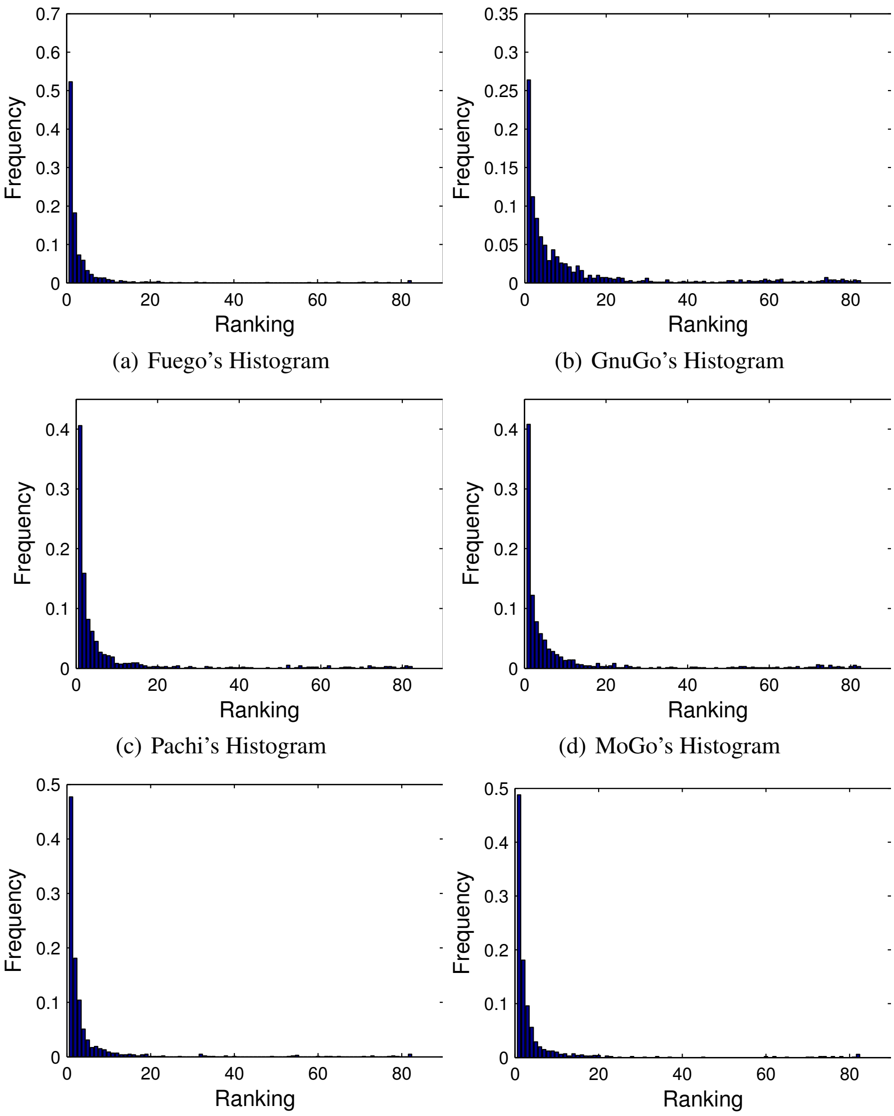
If the strongest agent of the team is not one of the agents involved in the tie, we can ignore the opinion of the strongest agent according to *Assumption 2*, and break the tie in favor of the strongest agent from the ones involved in the tie, because *Assumption 1* applies to any subset of the agents.

C Definition of FuegoΔ and FuegoΘ

In Section 4.2 we used agents FuegoΔ and FuegoΘ in our experiments. We present here the description of these agents. Fuego follows an UCT Monte Carlo Go algorithm, so it uses heuristics to simulate games during the Monte Carlo Simulations. There are mainly 5 possible heuristics in Fuego's code. These heuristics have a hierarchical order, and the original Fuego agent follows the order $<$Atari Capture, Atari Defend, Lowlib, Pattern$>$ (The heuristic called Nakade is not enabled by default). We created a variation of Fuego, that will be called FuegoΔ, that follows the order $<$Atari Defend, Atari Capture, Pattern, Nakade, Lowlib$>$. We also created FuegoΘ, that follows the order $<$Atari Defend, Nakade, Pattern, Atari Capture, Lowlib$>$. The memory available for FuegoΔ and FuegoΘ is half of the memory available for Fuego.

D Histogram of the Agents

In Section 4.2, we presented the histograms of Fuego and GnuGo, estimated over 1000 board states. Figure 1 shows the histograms of all agents.



(a) Fuego's Histogram

(b) GnuGo's Histogram

(c) Pachi's Histogram

(d) MoGo's Histogram

(e) FuegoΔ's Histogram

(f) FuegoΘ's Histogram

Fig. 1. Histogram of the agents, using real data.