

Improving Modeling of other Agents using Tentative Stereotypes and Compactification of Observations

Jörg Denzinger and Jasmine Hamdan
Department of Computer Science
University of Calgary
{denzinge,hamdanj}@cpsc.ucalgary.ca

Abstract

We investigate possible improvements to modeling other agents based on observed situation-action pairs and the nearest neighbor rule. Tentative stereotype models allow for good predictions of a modeled agent's behavior even after few observations. Periodic reevaluation of the chosen stereotype and the potential for switching between different stereotypes or to the observation based model aids in dealing with very similar (but not identical) stereotypes and agents that do not conform to any stereotype. Finally, compactification of observations keeps the application of the model efficient by reducing comparisons within the nearest neighbor rule.

Our experiments show that stereotyping significantly improves cases where using just the original method performs badly and that reevaluation and switching fortify stereotyping against the potential risk of using an incorrect stereotype. Compactification shows good potential for improving efficiency, but is sometimes at risk of losing important observations.

1 Introduction

Getting an idea about the decision making processes of others is a very useful and often necessary endeavour for both human beings and computer agents. In a multitude of scenarios, an agent's aptitude at modeling other agents directly effects its chances of successfully achieving its goals.

But modeling another agent is not easy. Since knowing the internal mechanisms of another agent is generally impossible, observing perceivable behavior is the best source of information for an agent trying to model other agents. In [4], combining observed pairs of situations and actions with a similarity measure on situations and the use of the nearest neighbor rule was suggested as a relatively simple method of modeling other agents based purely on observations. But,

compared to the ability of a human being to model others, the method is not very successful or efficient. The first, due to a scarcity of available observations leading to inaccurate predictions, and the latter due to an overabundance of available observations which slows down the search and selection process of the most similar situation and action pair.

In this paper, we evaluate improvements that deal with the problems arising from extreme numbers of observations. For the case of having very few observations we use *reevaluative stereotyping with switching*. *Stereotyping* is the usage of a complete model of another agent (or *stereotype*) to provide a modeling agent with a solid basis on which to base its predictions until more information is available. *Reevaluation* periodically assesses the validity of a chosen stereotype. *Switching* allows for a modeling agent to switch between different stereotypes to deal with similar stereotypes, or to the original method of modeling with observations if no stereotype seems to fit the modeled agent. To keep the number of similarity computations at an acceptable level, we suggest *compactifying* the observations to a smaller collection that guarantees efficiency without greatly compromising accuracy. We attempt to achieve this goal by only selecting observations within a certain vicinity of the situation an agent has to react to, and throwing away all other observations.

We tested these ideas on a simulation of the Pursuit Game, that allows modeling agents to on-line learn cooperative behavior for solving the game. Our experiments showed that when the correct stereotype was available, the performance of the modeling agents was similar to their performance when initially given the correct decision methods of modeled agents. In cases where very similar stereotypes were available, errors in stereotype selection were quickly detected and corrected through reevaluation and switching between stereotypes. Finally, even in cases where only incorrect stereotypes were available, switching to the original method using observations and the nearest neighbor rule enabled the modeling agents to achieve their goals. For compactification, we were able to observe many experiments

which not only showed improved run times but also resulted in better solutions, due to pruning away potentially misleading observations. However, this pruning sometimes came at a cost. Some experiments did not always speed up the modeling agents’ decision making, due to pruning away observations that later did prove essential.

2 Modeling other Agents’ Behaviors

In order to model other agents, it is necessary to discuss what an agent is in general, what information can be used by an agent to make decisions, and what modeling another agent means. After this, we will present modeling using observed situation-action pairs (OSAPs) and the nearest neighbor rule (NNR), a method first suggested in [4] to deal with the modeling of reactive agents. We will also point out the weaknesses of this approach.

2.1 Basic Definitions

In MAS, there is no agreement on the exact definition of an agent. However, most authors agree that an agent $\mathcal{A}g$ can be described as a triplet $\mathcal{A}g = (Sit, Act, Dat)$. Sit is the set of situations $\mathcal{A}g$ can experience, Act is the set of actions $\mathcal{A}g$ can perform and Dat is the set of the possible values for $\mathcal{A}g$ ’s internal data areas. For an agent in a MAS these components can be further structured, for example Dat distinguishes between data areas that contain information about the agent itself (Dat_{own}), areas that contain sure knowledge about other agents (Dat_{sk}) and areas that contain assumed (and therefore unsure) knowledge about other agents (Dat_{ak}). Internally, $\mathcal{A}g$ ’s decision mechanism is a function $f_{\mathcal{A}g}: Sit \times Dat \rightarrow Act$. But for an external observer lacking initial knowledge about the agent, it appears to be a function $g_{\mathcal{A}g}: Sit \rightarrow Act$, i.e. an observing agent can only perceive actions and situations, but not the internal mechanisms that drive an agent’s decision making. The extent of the difference between an actual $f_{\mathcal{A}g}$ and apparent $g_{\mathcal{A}g}$ distinguishes between *reactive* and *proactive* agents. We say an agent $\mathcal{A}g$ is reactive if its Dat has negligible influence on its decision making, and proactive (or knowledge-based) if its Dat has significant influence on its decision making. Finally, the realization of $\mathcal{A}g$ ’s actual $f_{\mathcal{A}g}$ depends on the agent’s architecture.

The process of collecting information about another agent to guess its actions is usually referred to as *modeling* other agents. Many existing modeling approaches rely on observing the behavior $B(\mathcal{A}g, s_0)$ of the agent $\mathcal{A}g$ to be modeled, starting from a situation s_0 . Usually, B can be described as a sequence of situations and the actions taken in these situations, i.e. $B(\mathcal{A}g, s_0) = s_0, a_0, s_1, a_1, \dots$ (with $s_i \in Sit$, $a_i \in Act$). An agent modeling another agent through observed behavior relies on all obser-

vations up to the current situation, i.e. a behavior sequence $s_0, a_0, s_1, a_1, \dots, s_{k-1}, a_{k-1}$.

A major obstacle to modeling another agent is the influence of the modeled agent’s internal data areas - the more influence Dat has on $g_{\mathcal{A}g}$ the more likely that modeling will be based on incorrect assumptions. Therefore, although not impossible, it is extremely difficult to adequately model a proactive agent. On the other hand, modeling a reactive agent is more feasible, due to the negligible influence of its Dat on decision making. Other general problems with modeling agents are different perceptions of modeling and modeled agent and changing behavior of the modeled agent.

2.2 Modeling Reactive Agents using OSAPs and NNR

As stated in the previous section, modeling a reactive agent is a more feasible task than modeling a proactive agent. [4] proposed a simple modeling method that enabled agents to model other reactive agents by using OSAPs and NNR together with some similarity (or distance) measure $dist$ on situations.

$dist$ depends on how situations are represented, therefore a detailed definition will be presented along with our application in Section 4, but in general $dist: Sit \times Sit \rightarrow \mathbb{N}$ measures how similar two situations are, with smaller values implying more similarity between two situations and vice versa. If we want to model agent $\mathcal{A}g$ and have observed its behavior $B(\mathcal{A}g, s_0) = s_0, a_0, s_1, a_1, \dots, s_k, a_k$ from situation s_0 to situation s_k , then the modeling method’s prediction for a situation s is a_j , if $dist(s_j, s) = \min\{dist(s_0, s), \dots, dist(s_k, s)\}$ - this is the well-known nearest neighbor rule. If there are several situations with the same similarity, then newer observations are given priority in the modeling process.

During experimentation, it was seen that when modeling using OSAPs and NNR, a model generated out of few observations often results in inaccurate predictions. On the opposite end of the extreme, a model comprised of many observations decreases the efficiency of the modeling process. Improvements to these problems were the prime goal of this work.

3 Improving Modeling using OSAPs and NNR

In this section, we suggest two improvements to modeling using OSAPs and NNR. The first improvement, *reevaluative stereotyping with switching* targets having only a few OSAPs and was motivated by psychological explanations of human stereotyping. The second improvement, *compactification of observations* tries to reduce observations to only those that are most relevant, using a tree-like structure.

3.1 Reevaluative Stereotyping with Switching

Stereotyping is a well-known concept in human behavior – loosely it is the classification of humans into basic behavior models (or stereotypes). Although generally seen in a negative light, it also may have some benefits – especially if a stereotyped human actually does fit a certain stereotype (see [9]).

If we look at modeling other agents using OSAPs and NNR, stereotyping can improve prediction accuracy when modeling after few observations. If the proper stereotype is available to the modeling agent then only a few observations of the modeled agent can identify it, and using that stereotype ultimately allows for better predictions. On the other hand, selecting the wrong (but similar) stereotype will not help in the long run – so stereotype choices can be continuously reevaluated according to new experiences, with the potential of switching between stereotypes to ensure choice correctness. Finally, if a stereotype even remotely similar to the modeled agent’s decision function f_{A_g} isn’t available, a modeling agent can also switch from using stereotypes to using OSAPs and the NNR. Therefore we do not propose stereotyping as an alternative to using OSAPs and NNR for modeling, we want to combine stereotyping, reevaluation and switching with the use of OSAPs and NNR. We called our improvement *reevaluative stereotyping with switching* due to these reevaluations and because we can switch between different stereotypes or to OSAPs with NNR if necessary. For the sake of brevity we will call the complete process *tentative stereotyping* (or just stereotyping).

We assume that the modeling agent has observed the modeled agent for some small period of time before modeling it. If agent $A_{g_{mod}}$ wants to model agent $A_{g_{obs}}$ using stereotyping and OSAPs with NNR, then $A_{g_{mod}}$ needs a set of stereotypes $Stereo = \{f_1, \dots, f_m\}$, where each $f_i: Sit_{mod} \rightarrow Act_{obs}$ is a possible agent decision function that uses the set of situations Sit_{mod} that $A_{g_{mod}}$ can observe and the set of actions Act_{obs} that $A_{g_{obs}}$ can perform (according to $A_{g_{mod}}$ ’s knowledge about $A_{g_{obs}}$). If the observed behavior is $B(A_{g_{obs}}, s_0) = s_0, a_0, s_1, a_1, \dots, s_k, a_k$, then $A_{g_{mod}}$ does its modeling as follows:

For each $f_i \in Stereo$ its potential behavior $Bst(f_i, B(A_{g_{obs}}, s_0))$ is evaluated as

$$Bst(f_i, B(A_{g_{obs}}, s_0)) = \sum_{j=0}^k eq(f_i(s_j), a_j)$$

where

$$eq(a, a') = \begin{cases} 1, & \text{if } a = a' \\ 0, & \text{else} \end{cases}$$

This means that Bst tells us how often f_i would have predicted the right action of $A_{g_{obs}}$.

Assume a “switch time” p_{OSAP} which defines a minimum number of modeling periods in which stereotyping must take place and a “threshold value” t_{Bst} is given. A stereotype is selected to be used for predicting $A_{g_{obs}}$ ’s behavior if: it has the maximal $Bst > t_{Bst}$ and the p_{OSAP} modeling periods have passed or it just has maximal Bst and p_{OSAP} modeling periods have not yet passed. We use p_{OSAP} to give the stereotyping a chance and to ensure an increase in the number of observations made. We switch to OSAPs and NNR in two cases: p_{OSAP} modeling periods have passed and no stereotype gave $Bst > t_{Bst}$ or p_{OSAP} modeling periods have passed and the performance of $A_{g_{mod}}$ has gotten worse since the last time a stereotype was selected. The definition of “worse performance” is obviously task dependent, therefore a specific definition will be provided in Section 5.

3.2 Compactifying OSAPs Based on kd-Trees

On the other end of the spectrum with regard to number of observations, we have the problem that if a model is based on k OSAPs, then OSAPs and NNR computes the *dist*-value between the current situation and all k situations, which can be time consuming if k is large. A similar problem is faced in case-based reasoning, for finding the most similar case in a case base and is addressed in [14].

Cases are composed of multiple features, where each case has the same number of features as the other cases in the same case-base. For quicker indexing, the cases can be structured into a tree. The general idea behind these *kd-trees* is to build a tree that branches at each level according to a different feature, and a partition value selected according to some partition criteria that would ensure a balanced tree. The leaves of the tree then contain all cases with feature values in the intervals determined by the path to the leaf. Then finding the right leaf requires comparing the current case with the inner nodes of the path matching it (to identify the right path). And finding the most similar case to the case at hand requires comparing it to all cases contained in the found leaf.

If we consider situations and their features as cases, the approach of [14] can be directly applied to our problem with OSAPs and NNR. And although we were majorly inspired by the concept, we chose to use more of a *compactification* of the OSAPs, rather than the structuring provided by a tree. Extremities of the situations’ features are taken to form a virtual boundary around the observed situation space. According to a predefined partition criteria, this virtual boundary is divided into subspaces at the onset of modeling and the subspace to which the current situation belongs is determined. Observed situations outside of that subspace are completely discarded, and the NNR only considers the remaining observed situations for comparison to the current

situation. As we will see, this is not without problems.

4 OLEMAS: Using Agent Modeling for On-line Learning of Cooperative Behavior

Pursuit Games cover a wide range of variants from rather simple to very complex and therefore allow for rather complex behaviors of agents. Due to this range of variants, we have chosen to evaluate the ideas of Section 3 using Pursuit Games, more precisely using the ideas to model prey agents during on-line learning of cooperative behavior for hunter agents in different variants of the game (as suggested in [4]). First we will present the role of modeling other agents in on-line learning of behavior and then we will give a brief description of Pursuit Games and our OLEMAS system.

4.1 On-line Learning of Cooperative Behavior

Learning is an ability that many people want to see in an agent since it allows agents to react to changes and new information and adapt to them. [4] presented a general method of how off-line learning approaches can be used to do on-line learning. The basic idea is to add a special action “learn” to the actions of each learning agent. “Learn” is periodically performed and results in activating a simulation of the environment an agent is in, including all other agents, that is used by the off-line learning method to generate a new agent strategy, i.e. produce a new, improved decision function. The simulation first determines what the situation will be for the agent after the time needed for executing “learn” and starting with this situation the off-line learning method can work as usual, using the simulation to get the feedback or experiences it needs.

In order for this approach to work, the accuracy of the simulation is of quite some importance, because bad simulations will at best slow down task completion and at worst make it impossible. And modeling of the other agents is essential for the quality of the simulations. Therefore, on-line learning of cooperative behavior is a very good testbed for approaches on modeling other agents.

4.2 Pursuit Games and the OLEMAS System

Pursuit Games are scenarios that have long been suggested as testbeds for cooperation approaches for agents (see [1]). They offer a lot of experimental variations and so are very interesting for evaluating learning approaches for agents and agent teams, since different variants require rather different strategies. The general idea of a pursuit game is that a group of hunter agents has to catch one or several prey agents in some primitive (grid) world. Game variants diversify the definition of catch, the actions the agents can do, their shape and speed, the world and much more.

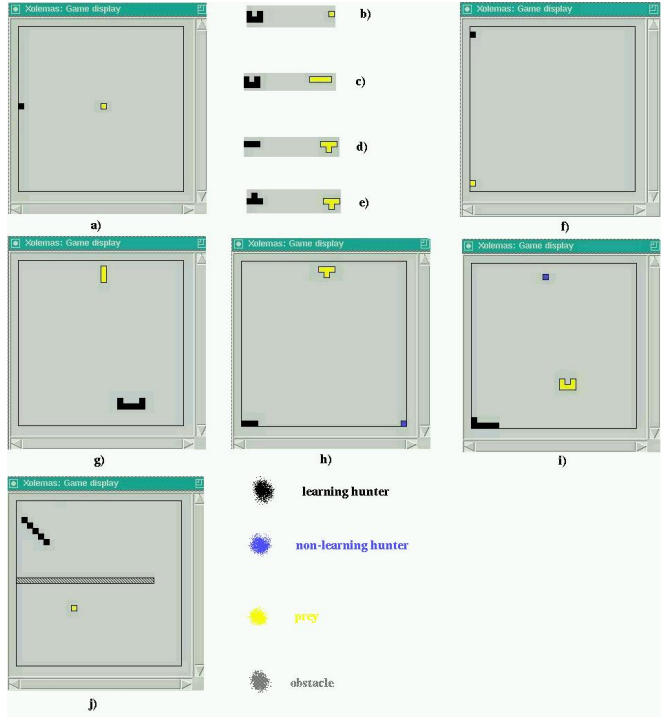


Figure 1. Start situations

The OLEMAS system (**O**n-**L**ine **E**volution of **M**ulti-**A**gent **S**ystems) was developed to evaluate evolutionary learning approaches using Pursuit Games as an application area. The particular learning approach we were interested in used situation-action pairs (SAPs) and NNR as agent architecture and evolved sets of SAPs to be used as agent strategies (see [4] for further details on the learning approach).

A situation in a Pursuit Game in OLEMAS for a particular agent is described as a vector that contains for each other agent and obstacle the coordinates of the other agent relative to the agent and its orientation (measured for predefined centerpoints so that agent shape is not an issue). This allows us to define the function $dist$ for two situations $s = (x_1, y_1, o_1, \dots, x_n, y_n, o_n)$ and $s' = (x'_1, y'_1, o'_1, \dots, x'_n, y'_n, o'_n)$ as follows:

$$dist(s, s') = \sum_{i=1}^n ((x_i - x'_i)^2 + (y_i - y'_i)^2 + ((o_i - o'_i) \bmod 8)).$$

The o_i 's are the orientation of the agent (one of 0 = north, 1 = east, 2 = south, 3 = west), which is also measured with respect to a predefined base orientation for each agent.

5 Experimental Evaluation

We tried to present rather different Pursuit Game variants to show the generality of the improvements, in addition

Table 2. Switching from Stereotyping to OSAPs for Different p_{OSAP} -values

V	OSAPs			Correct Model			Incorrect Stereotype			$p_{OSAP} = 2$			$p_{OSAP} = 3$			$p_{OSAP} = 4$		
	St	T	Su	St	T	Su	St	T	Su	St	T	Su	St	T	Su	St	T	Su
7	70.9	11.6	100	38.6	7.4	100	123.8	18.0	70	83.5	12.6	100	105.3	14.8	100	91.5	13.2	100
8	55.3	9.1	100	42.7	8.5	100	166.4	21.7	50	96.0	13.5	100	107.2	15.5	100	115.9	16.1	100
9	121.8	75.2	100	77.1	25.3	100	200.0	36.3	0	118.7	43.5	100	138.7	101.1	70	160.0	83.6	70
10	178.1	29.3	30	141.1	38.1	40	162.9	43.4	30	160.0	31.9	30	168.8	34.8	30	137.2	30.4	60
11	200.0	102.3	0	60.5	13.3	100	139.2	43.5	60	117.1	41.2	60	147.2	50.3	70	113.2	31.5	90
12	122.0	31.1	80	100.0	24.5	100	177.2	33.6	20	123.7	29.5	90	145.9	30.7	80	149.0	32.8	80

Table 4. Combination of the Ideas

V	Only Incorrect Stereotype			With Switching and Full OSAP Set			With Switching and Compactification			p_{OSAP}
	St	T	Su	St	T	Su	St	T	Su	
21	426.2	53.7	20	188.9	42.1	100	147.7	22.4	100	3
22	416.5	52.7	50	217.3	40.5	90	146.6	22.4	100	2
23	488.6	61.5	40	156.7	25.1	100	180.5	27.8	100	3
24	500.0	62.6	0	167.8	27.2	100	182.1	27.1	100	2
25	464.9	58.3	80	409.0	103.6	30	392.8	73.0	50	2
26	391.8	49.9	40	151.8	28.5	100	164.2	25.1	100	2
27	431.8	94.4	40	252.7	147.3	80	263.6	139.5	100	2

Table 1. Correct Stereotype Available

V	OSAPs			Correct Model			Stereotyping		
	St	T	Su	St	T	Su	St	T	Su
1	96.7	17.3	90	50.1	8.8	100	48.0	10.2	100
2	191.4	36.8	10	61.6	12.8	100	73.0	14.5	100
3	48.9	8.6	100	43.1	8.9	100	43.0	10.0	100
4	100.5	16.2	80	40.0	6.8	100	68.6	10.4	100
5	196.7	36.3	10	78.1	14.4	100	90.0	15.1	90
6	90.0	16.0	100	49.3	15.8	100	49.6	15.9	100

Table 3. Influence of Compactifying

V	Full			Compactified		
	St	T	Su	St	T	Su
13	207.3	53.5	90	139.0	23.7	100
14	413.6	150.2	40	452.2	108.2	20
15	140.5	31.9	100	86.6	14.5	100
16	114.0	22.4	100	99.8	16.7	100
17	130.1	30.9	100	89.8	16.9	100
18	72.0	13.1	100	74.1	11.9	100
19	116.3	21.1	100	142.1	22.3	100
20	53.6	8.8	100	54.3	9.0	100

to the problematic results that will require further improvements. The experiments are divided into four groups: having a correct stereotype available, having no correct stereotype available, compactification of OSAPs, and the combined improvements.

Our analysis concentrates on the average number of steps and average amount of time required for a game (including those of unsuccessful trials), in addition to the rate of successful trials (in percent). In the result tables, these are represented as **St**, **T** and **Su** respectively. The results of each table entry are the average of ten runs.

In all experiments, agents were situated on a 30 by 30 grid. They could move in all eight directions (N, NE, E, SE, S, SW, W, NW) and could rotate 90 degrees to the left or right. Every move took 1 step to accomplish, and a “learn” phase would occur after at least 13 and at most 33 steps. For experiments involving stereotyping, agents had 200 steps to achieve their goal, in order to put a kind of pressure on winning in a short amount of time. As stereotypes, we used 5 strategy functions provided by OLEMAS and 10 sets of

SAPs. For compactification experiments agents had 500 steps, in order to effectively show the impact of compactification at the end stages of the game. A median dividing the x axis of the observed situations was used as the single partition criteria. Note that in the following, improved *hunter performance* refers to a decrease in the number of steps necessary for goal achievement and/or an increase in success rate.

The following variants had as game goal *kill* (i.e. occupying the same square as the prey): 1-9 and 13-26. All other variants had to *immobilize* the prey. Variants 1, 2, 6, 13-17, and 21-25 started from a) in Figure 1. Variant 3 substituted b) for the agents in a), 4 and 11 substituted c) and variants 8 and 20 substituted e). Variants 5, 18, and 19 used the agents of a) but placed them randomly. Variant 7 also placed the hunter from a) and the prey from e) randomly. Variant 9 started with j), 10 with h), 12 with i), 26 with f) and 27 with g). The preys in variants 1, 13 and 21 tried to evade the nearest hunter, the preys in variants 2, 4, 5, 10, 11, 14,

17, 19, 23, 25 and 26 tried to avoid all hunters and all grid borders (measuring distances in various ways), in variants 9, 12, 15, 16, 18, 22, 24, and 27 only the hunters counted. Finally, the preys in variants 3, 6-8 and 20 used some sets of SAPs as strategies.

The first group of experiments evaluates hunter performance when a correct stereotype for the prey is available (see Table 1). Our objective here was to show that using a correct stereotype can significantly improve hunter performance in cases where using OSAPs yields poor results (which, as Table 1 shows, is the case). On the other hand, we do not expect much improvement when using OSAPs already yields good results (as shown by variant 3). We also expected stereotyping to result in slightly worse performance compared to when a hunter knows the prey’s actual strategy – also shown in Table 1. The utility of reevaluation becomes especially evident in variant 6, although it is not apparent from the results. In variant 6, the available stereotypes were very similar and the first selected was not the prey’s actual strategy. But as more observations were made, and periodic reevaluation ensued, the correct stereotype was eventually chosen. As the result shows, this reevaluation and switching to another stereotype process does not adversely effect performance.

The second group of experiments shows the results of using an incorrect prey stereotype with switching to OSAPs at different values of p_{OSAP} – recall that p_{OSAP} denotes the minimum number of learning periods in which stereotyping must occur (see Table 2). The main objective of these experiments was to assess the adverse effects of stereotyping without good knowledge, and to evaluate how the switch to OSAPs and NNR limits potential damage. As the table shows, switching from using an incorrect stereotype to modeling with OSAPs significantly improved the performance of all variants when compared to sticking with the wrong stereotype. We did not expect wrong stereotypes to yield better results than using OSAPs and NNR or a correct prey stereotype. But surprisingly variants 9-11 showed that wrong stereotypes might not always be more misleading than OSAPs. And for variant 10, we actually got better results (at $p_{OSAP}=4$) than with using the correct model, which suggests that sometimes a detour might even have positive effects. With regard to determining a good p_{OSAP} -value, the table shows that it depends on the variant, although a value of 2 is often good. Note that in the context of when to switch, worse performance is defined as an increased distance between the hunter and prey in two consecutive modeling periods.

The third group of experiments examined the results of a hunter using a compactified set of OSAPs for prey modeling (see Table 3). Our objective was to show that using this set is more efficient than using the full set of OSAPs. This can be seen in a decrease of the time per step ratio

T/St , which we had for every variant, except for 20, where T/St was unchanged. We did expect that in some cases performance may slightly degrade when compactifying because of the removal of potentially relevant OSAPs, leading to inaccurate predictions. But as a very positive result, some cases actually showed a decrease in the number of steps for goal achievement. This indicates an increase in prediction accuracy for some scenarios, which can be attributed to the compactification process removing a number of OSAPs that may have incorrectly influenced the hunter’s predictions without compactification. However, the other side of the coin is also visible, a (not only slight) increase in steps (and consequently time) or even a decrease in success. The concept of compactification seems promising, but the particular approach tested requires improvement.

The final group of experiments presents the results of reevaluative stereotyping with switching to OSAPs including compactification of those OSAPs (see Table 4). Again it is clear that switching from an incorrect stereotype to modeling with OSAPs significantly improved the hunter’s success in goal achievement compared to using an incorrect stereotype. In addition, using the compactified set of OSAPs following switching decreased T/St in most variants, verifying that the ideas can be combined (as expected).

6 Related Work

Modeling other agents has been researched from several perspectives. For example, [6] describes the Recursive Modeling Method, which incorporates concepts from decision and game theory into agent modeling. In [8], agents are assumed to be homogeneous and predictions can be based on that assumption. [15] assumes some theoretical optimal behavior as the model for an agent. [3] uses observations to produce finite automata as models for other agents, while [13] represents models of other agents as influence diagrams. In [10], coordinating agents use case-based decision theory and communicated meta-level information for learning.

We see connections between our improvements to OSAPs and NNR and to [5] and [11]. The approach in [5] can be seen as producing probability values for combinations of features that together constitute an agent model. Our stereotypes can be seen as a subset of the set of all feature combinations, thus having a much smaller number of predefined models to handle, while being able to switch to OSAPs and NNR still gives us the full flexibility of this agent architecture (and all possible features that can be observed). In [11], agents used observed data to classify opponent agents into predefined opponent teams for simulated robotic soccer, under the assumption that opponent teams did not change strategies. No reevaluation of these classifications is done, nor is it possible to deal with opponents

that are outside of the classification.

Our approach to modeling other agents also has some connections to the area of plan recognition (see [2] for an overview), especially case-based plan recognition. In [7], for example, observations are used to fill holes in partial plans, based on similarity of situations, very similar to [4]. The use of partial plans has some resemblance to stereotypes, but because holes in those plans are required to be filled, the necessity of having a complete model for prediction after just a few observations (which is achieved by our approach) is not addressed. For an overview of prototype selection and generation techniques in the area of case-based reasoning, see [16].

Finally, modeling another agent can be seen as a very special case of Belief Revision. But most of the work in this area, due to the need to be more general than just modeling a reactive agent, is based on Truth Maintenance Systems and does not make use of the features of our special case (see [12] for a brief overview of this area).

7 Conclusion

We presented two possible improvements to modeling of other agents based on OSAPs and NNR. By starting off with tentative stereotyping, a stereotype can provide a complete model of another agent and can be determined after just a few observations. In the case that the agent fits the stereotype, the modeling agent's performance is usually greatly improved. When multiple stereotypes are very similar, periodic reevaluation ensures that the most correct stereotype is always in use. If no correct stereotype is available, the ability to switch to the basic observation based method prevents the modeling agent from poor performance. Compactifying observations through simple feature-based partitioning not only offers a more efficient use of a model, it can also eliminate misleading observations. But our experiments show that it can also pose the risk of losing important observations. In this area, more research is necessary to find better partitioning criteria to minimize this risk.

References

- [1] M. Benda, V. Jagannathan, and R. Dodhiawalla. An Optimal Cooperation of Knowledge Sources, Technical Report BCS-G201e-28, Boeing AI Center, 1985.
- [2] S. Carberry. Techniques for Plan Recognition, *User Modeling and User-Adapted Interaction*, 2001, 11(1-2), pp. 31–48.
- [3] D. Carmel and S. Markovitch. Opponent modeling in multi-agent systems, In: Weiss, Sen, (eds), *Adaptation and Learning in Multi-Agent Systems*, Springer, 1995, pp. 40–52.
- [4] J. Denzinger and M. Kordt. Evolutionary On-Line Learning of Cooperative Behavior with Situation-Action Pairs, Proc. ICMAS'00, IEEE Press, 2000, pp. 103–110.
- [5] L. Garrido, R. Brena, and K. Sycara. Towards Modeling other Agents: A Simulation-Based Study, Proc. MABS 1998, Springer, 1998, pp. 210–225.
- [6] P. Gmytrasiewicz and E. Durfee. Reasoning about Other Agents: Philosophy, Theory and Implementation, Proc. 12th WS on DAI, 1993, pp. 143–153.
- [7] B. Kerkez and M. Cox. Case-Based Plan Recognition with Incomplete Plan Libraries, Proc. AAAI Fall 2002 Symposium on Intent Inference, 2002, pp. 52–54.
- [8] J. Kok and N. Vlassis. Mutual Modeling of Teammate Behavior, Technical Report IAS-UVA-02-04, University of Amsterdam, 2002.
- [9] Macrae et al. Stereotypes as Energy-Saving Devices: A Peek Inside the Cognitive Toolbox, *Journal of Personality and Social Psychology* 66(1), 1994, pp. 37–47.
- [10] M. V. N. Prasad and V. R. Lesser. Learning Situation-Specific Coordination in Cooperative Multi-agent Systems, In *Autonomous Agents and Multi-Agent Systems*, 1999, pp. 173–207.
- [11] P. Riley and M. Veloso. On Behavior Classification in Adversarial Environments, Proc. DARS 2000, Springer-Verlag, 2000, pp. 371–380.
- [12] S. Shapiro. Belief Revision and Truth Maintenance Systems: An Overview and a Proposal, Internal report, State University of New York at Buffalo, 1998.
- [13] D. Suraydi and P. Gmytrasiewicz. Learning Models of Other Agents Using Influence Diagrams. Proc. 7th Intern. Conf. on User Modeling (UM-99), 1999, pp. 223–232.
- [14] S. Wess, K.-D. Althoff, and G. Derwand. Using kd-trees to improve the retrieval step in casebased reasoning, Proc. EWCBR-93, Springer, 1994, pp. 167–181.
- [15] P. Stone, P. Riley and M. Veloso. Defining and Using Ideal Teammate and Opponent Agent Models, Proc. AAAI-2000, AAAI Press, 2000, pp. 1040–1045.
- [16] R. Wilson and T. Martinez. Reduction Techniques for Instance-Based Learning Algorithms, *Machine Learning*:38, 2000, pp. 257–286.