Using Stereotypes and Compactification of Observations to Improve Modeling of other Agents

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

April 26, 2004

Abstract

This paper investigates improvements to modeling other agents based on observed situation-action pairs and the nearest-neighbor rule, which suffers when dealing with very few or very many observations. Stereotype models allow for good predictions of a modeled agent's behavior even after few observations. To handle any adverse effects of stereotyping, periodic reevaluation of the chosen stereotype and the potential to switch between different stereotypes aids in dealing with very similar, but not identical, stereotypes. Also, periodic reevaluation and the potential to switch from a chosen stereotype to the original observation based modeling method aids in dealing with agents that do not conform to any stereotype. Finally, compactification of observations keeps the application of the original modeling method efficient by reducing comparisons within the nearest neighbor rule.

Our experiments within the OLEMAS system show that stereotyping significantly improves cases where just using the original modeling performs poorly. In addition, reevaluation and switching fortify stereotyping against the potential risk of using an incorrect stereotype. In many cases, compactification both increases the efficiency of using observed situation-action pairs and improves predictions by filtering out misleading observations. At times however, this filtering comes at the cost of prediction accuracy by removing relevant observations.

1 Introduction

Getting an idea about the decision making criteria and processes of others is a very useful and often necessary endeavour for both human beings and computer agents. Whether it be a cooperative agent scenario where communication is limited, expensive or risky, or a competitive agent scenario, where the gain of one agent requires the loss of another – the better an agent models other agents, the better are its chances of successfully achieving its goals.

Since it is usually impossible to know the internal and private decision mechanisms of other agents, observed data is the best source of information for an agent trying to model othes. But what does it do to predict the behavior of the other agents in situations that were never observed? In [DK00], combining observed pairs of situations and actions, a similarity measure on situations and the use of the nearest neighbor rule was suggested as a primitive way to model other agents based purely on observations. However, compared to the ability of a human being to model others, the method is not very successful or efficient. Poor success rates seem to arise if few observations are available, leading to inaccurate predictions. At the opposite end of the extreme, inefficiency is a side effect of having very many observations, and therefore many comparisons within the nearest neighbor rule, which slows down the search and selection process that ultimately yields a prediction of the modeled agent's behavior.

In this paper, we evaluate improvements that deal with the problems arising from using the basic method of [DK00] in the face of having extreme numbers of observations, by increasing the knowledge a modeling agent incorporates into dealing with those extreme cases. For the case of poor success rates due to few observations, we use *reevaluative stereotyping with switching*, a concept inspired by the study of stereotypes in human psychology. *Stereotyping* refers to the usage of a complete and predefined model of another agent (or *stereotype*) to provide a modeling agent with a solid basis on which to base its predictions until more information becomes available. *Reevaluation* periodically asseses the validity of a chosen stereotype. *Switching* allows for a modeling agent to switch between different stereotypes to deal with very similar stereotypes, or to the original method of modeling with observations if no stereotype seems to fit the modeled agent and an adequate amount of observations have been aquired. To keep the number of similarity computations at an acceptable level, we suggest *compactification of observations* to a collection that is small enough to guarantee efficiency without compromising accuracy. We hoped to achieve this goal by only selecting observations that are in a certain vicinity of the situation an agent has to react to, and completely ignoring all other observations.

We tested our improvements with the OLEMAS system, a simulation of the well-known Pursuit Game in which modeling agents solve variants of the game through on-line learning of cooperative behavior. Our experiments showed that when a correct stereotype was available, stereotyping significantly improved the performance of modeling agents. When very similar stereotypes were available, only one of which was correct, reevaluation and the potential for switching between stereotypes quickly corrected erroneous choices of a stereotype. When no correct stereotype was available, reevaluation and the potential to switch to the basic method of using observed situation-action pairs and the nearest neighbor rule saved modeling agents from the danger of basing predictions on completely incorrect models. Our experiments on compactification showed that at times not only did the efficiency of the basic modeling method improve (shown by decreased run times), but that resulting solutions also improved due to the pruning of potentially misleading observations. However, it was also seen that this pruning sometimes came at a cost of accuracy, with solutions suffering due to the pruning of observations that did eventually prove to be essential.

2 Modeling other Agents' Behaviors

In order to model other agents' decision making mechanisms and resulting behavior, it is first necessary to discuss what an agent is in general, what information can be used by an agent to make decisions about others and what modeling another agent means. After this, we will present the modeling method suggested in [DK00] for modeling reactive agents, i.e. to use observed situation-action pairs (OSAPs) and the nearest-neighbor rule (NNR) for decision making. We will also point out the weaknesses of this approach.

2.1 Basic Definitions

There is no agreement in MAS on the exact definition of an agent, but most authors agree that on an abstract level an agent $\mathcal{A}g$ can be described as a triple $\mathcal{A}g$ = (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 provides the possible values for the internal data areas of $\mathcal{A}g$. For an agent in an multi-agent system, it makes sense to structure the three sets a little bit with regard to what concerns just the agent and what concerns other agents. For Dat, for example, this means that we should distinguish 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 knowledge, i.e. unsure knowledge, about other agents (Dat_{ak}).

One of the reasons for distinguishing sure knowledge and assumptions becomes obvious when we look at an agent's decision making. Internally, $\mathcal{A}g$ is a function $f_{\mathcal{A}g}$: $Sit \times Dat \rightarrow Act$. But for an external observer without any initial knowledge about the agent, the agent seems 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. Therefore, an observing agent should represent all knowledge gained through observations as assumptions. 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_{own} has negligible influence on its decision making. An agent is proactive (or knowledge-based) if its Dat_{own} has a significant influence on its decision making. Finally, the realization of $\mathcal{A}g$'s actual $f_{\mathcal{A}g}$ is determined by the agent's architecture.

The process of collecting information about another agent to predict its actions is generally refered to as *modeling* other agents. There are many different approaches as to how modeling another agent can be done, but the starting point of many of these approaches is observing the behavior $B(\mathcal{A}g,s_0)$ of the agent $\mathcal{A}g$ that is modeled starting from a situation s_0. Usually, B can be described as a sequence of situations and actions taken in these situations, i.e. $B(\mathcal{A}g,s_0) = s_0,a_0,s_1,a_1,...$ (with $s_i \in Sit$, $a_i \in Act$). Note that s_i is in a MAS not the result of agent $\mathcal{A}g$ taking action a_{i-1} in situation s_{i-1}, there are other agents and their actions also contribute to the fact that the next situation is s_i. If an agent models another agent, than obviously it cannot use an infinite sequence of observations, but is limited to the observations up to the current situation, i.e. a behavior sequence $s_0,a_0,s_1,a_1,...,a_{k-1},s_k$.

A number of issues complicate the process of modeling other agents. For example,

a modeling agent may not know the basic agent architecture of the modeled agent, a model of an agent may change over time, and the perceptions of the modeling agent can have negative influences, so that if two situations that are different from the perspective of the modeled agent are perceived by the modeling agent as similar, then the resulting model will not be very accurate.

But the biggest 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 it is extremely difficult to adequately model a proactive agent (although not impossible, for example there may be situations where its decisions are not based on the current value of its internal data areas at all). On the other hand, modeling a reactive agent is more feasible, due to the negligible influence of its Dat on decision making and particularily if observations covering the whole set Sit are available. But since modeling generally has to be based on a limited number of observed situations and subsequently taken actions, the problem of modeling even reactive agents is not easily solved.

2.2 Modeling Reactive Agents using OSAPs and NNR

As stated in the previous section, although modeling reactive agents is a feasible task, it is definitely not trivial under such realistic conditions as limited observations. In [DK00], a primitive modeling that enabled agents to model reactive agents (even in the face of limited observations) was proposed. This modeling method is to use the observed situation-action pairs (OSAPs) and the nearest-neighbor rule (NNR) together with some similarity (or distance) measure $dist$ on situations.

Naturally, $dist$ 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 $B(\mathcal{A}g,s_0) = s_0,a_0,s_1,a_1,...,s_k,a_k$ from situation s_0 to situation s_k, then the prediction of the modeling method for a situation s is a_j, if $dist(s_j,s) = \min\{dist(s_0,s),...dist(s_k,s)\}$. – this is the well-known nearest-neighbor rule. Basically, the rule states that the two situations with the smallest distance between them are the most similar, and that the action predicted is the one associated with the most similar situation. If there are several situations with the same similarity, then we use the action associated with the highest indexed situation, which ensures that newer observations have priority in the modeling process.

Modeling other agents using OSAPs and NNR has several advantages: we do not need any information about the agent's real architecture, if only one OSAP is present, it will already be able to suggest an action, it works for situations that never have been observed, it allows to incorporate new observations easily and, as was shown in [DK00], given enough observations it comes up with rather good predictions for reactive agents. Also, it has some potential to deal with learning agents (if different actions are observed for the same situation, the last observation is used) and if a proactive agent is modeled we still can correctly predict actions for situations where the influence of the data areas is neglectable.

But there are also some general problems. As is the case for all modeling approaches for other agents, the perceptions of the agent doing the modeling and its limitations can have negative influences. If two situations that are different for the modeled agent are perceived by the modeling agent as similar, then the resulting model obviously will not be very accurate. But, as already stated, this is a problem that cannot be avoided, as is the general problem caused by proactive agents.

More specific problems also became evident during experimentation on modeling using OSAPs and NNR. It was seen that a model generated out of few observations often results in inaccurate predictions. This indicates that a certain number of observations is necessary before modeling is satisfactory. On the opposite end of the extreme, a model comprised of many observations decreases the efficiency of the modeling process. This is because the similarity function is evaluated for every situation in the observed behavior, and quite a number of computations would be necessary to decide which action the model predicts for a certain situation.

Both these problems arise due to a modeling agent's lack of knowledge - in the case of too few observations, it is in the literal sense that the agent doesn't have enough information. In the case of too many observations, lack of knowledge is manifested in the use of brute force and indiscriminatory comparisons for the NNR. Therefore, both problems can be better handled if we incorporate additional knowledge into the basic method. We will present these improvements in the next section.

3 Improving Modeling with OSAPs and the NNR

In this section, we present two improvements to the simple modeling method of using OSAPs and NNR. They target the problems at opposing extremities in terms of number of observations, and both provide additional knowledge to the modeling method. The first improvement, *reevaluative stereotyping with switching*, targets having only a few OSAPs and was motivated by psychological explanations of human behavior when dealing with others, namely the concept of stereotypes. The second improvement, *compactification of observations*, employs concepts known from case-based reasoning to try to reduce the observations to just a subset containing only those that are most relevant.

3.1 Reevaluative Stereotyping with Switching

Many human beings are able to predict likely behavior of other persons rather well, even if they have known this person only for a very short time. This phenomenon may be explained by the concept of stereotyping, i.e. a human being has certain basic behavior models, the so-called *stereotypes*, and classifies newly met persons into the most likely model based on the first impression. One definition of a stereotype states that it is used to express a behavior of "persistent repetitiveness and unchanging mode of expression" (see [Jo82]), suggesting that stereotyping may not be the best way of dealing with first impressions. But [Li22] states that

> "... the real environment is altogether too big, too complex... We[individuals] are not equipped to deal with so much subtlety, so much variety, so many

> permutations and combinations. And although we have to act in that environment, we have to reconstruct it on a simpler model before we can manage with it."

With this regard, (human) stereotyping becomes a necessity. In fact, it can even be beneficial, as can be found in [Ma94], which states that stereotyping and categorization (another term for stereotyping) are "... mental tools...that... free up limited cognitive resources for the performance of other necessary or desirable mental activities".

If we look at modeling other agents using OSAPs and NNR, then stereotyping can greatly reduce the problem of not having enough observations to allow for good predictions of the observed agent's behavior. If the proper stereotype is available to the modeling agent, then only a few observations of the modeled agent can identify the right stereotype, and the enhanced knowledge provided by the stereotype aids in making predictions, ultimately allowing the modeling agent to make better informed decisions. But naturally the modeling agent might suffer from the same negative effects that can be observed for human stereotyping: a stereotype may be completely unrealistic and assumptions based on it completely inaccurate. To extend the human example then, stereotype choices can be continously reevaluated according to new experiences and encounters, with the potential of switching between stereotypes to ensure the correctness of a chosen stereotype. The problem still remains that a stereotype even remotely similar to the modeled agent's decision function $f_{\mathcal{A}g}$ may not exist, so flexibility is introduced by allowing a modeling agent to 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.

The basic idea is for a modeling agent to use stereotyping for some minimum amount of time. In fact, the agent is forced to stereotype until the minimum time has passed to ensure an accrual of diverse observations. After that point a switch to OSAPs and NNR may occur if it seems that the modeled agent does not conform to any stereotype, or if there appears to be a degradation in performance. We assume that the modeling agent has observed the modeled agent for some small period of time before modeling it. We also assume that after deciding on an initial stereotype, this decision will be periodically repeated until the task the agents have to perform or make decisions about is finished. Naturally, the period can be just one action, but usually it is more efficient to reevaluate a model after having observed several situations and the performed actions. 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 we encounter problems with the known stereotypes. For the sake of brevity we will call the complete process *stereotyping*.

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

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

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

where

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

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

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 $\mathcal{A}g_{obs}$'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 $\mathcal{A}g_{mod}$ 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 Compactification of 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 [WAD94].

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 [WAD94] 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. This improvement increases the knowledge that the NNR uses because it provides a more logical basis for decision making, by only operating on those OSAPs that are more probably closer to the situation at hand. 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

The application in which we are evaluating our improvements of Section 3 is the same as in [DK00], modeling other agents during on-line learning of cooperative behavior for hunter agents in different variants of the Pursuit Game. First we will present the on-line learning of behavior that needs modeling of other agents 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. It allows to simply give a task to an agent or a team and let them figure out how to perform it and, as in on-line learning, it allows agents to react to changes and new information and adapt to them. [DK00] presented a general method 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 that is periodically performed. "Learn" 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. While many off-line learning approaches can deal with some inaccuracy in the data and the fact that "learn" is executed periodically and therefore allows for correcting poor strategies, bad simulations will at least slow down the agents in performing the task to learn, if not making it impossible. Therefore it is very important to have good models of the other agents. While members of the own team might be willing to communicate their decision functions to a learning agent (resulting in a perfect model) this will definitely not be the case for opponents. So, on-line learning of cooperative behavior is a very good testbed for modeling approaches for other agents and if the scenarios in which the agents act are time sensitive, a lot of the agents will be more on the reactive side.

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 [BJD85]). They offer a lot of variants and so are very interesting for evaluating learning approaches for agents and agent teams, since rather different strategies are required for different variants. The general idea of a pursuit game is that a group of hunter agents has the catch one or several prey agents in some primitive grid world. Variants vary the definition of catch, the actions the agents can do, their shape and speed, the world and much more (see [DF96] for possible dimensions that can be varied).

The OLEMAS system (**O**n-**L**ine **E**volution of **M**ulti-**A**gent **S**ystems) was developed to evaluate learning approaches, more precisely evolutionary learning, using Pursuit Games as an application area. The particular learning approach we were interested in (see [DF96] and [DK00]) was using situation-action pairs (SAPs) and NNR as agent architecture and evolves agent strategies that are sets of SAPs. The concrete learning method is not so important for this work,for more details please refer to the given references.

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 $s = (x_1,y_1,o_1,...,x_n,y_n,o_n)$ and $s' = (x'_1,y'_1,o'_1,...,x'_n,y'_n,o'_n)$ as follows:

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

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 our improvements but also present the problematic results that will require additional improvements in the future. The experiments were divided into four main classes:

1. Correct stereotyping (i.e. learning hunter had a correct prey stereotype available).
2. Incorrect stereotyping (i.e. no correct prey stereotype available) followed by a switch to OSAPs and NNR.
3. Compactification of OSAPs.
4. Incorrect stereotyping followed by a switch to OSAPs and NNR, followed by compactification of OSAPs.

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. Also, recall that p_{OSAP} indicates a "switch time", i.e. the number of modeling periods in which stereotyping must occur. The results of each table entry are the average of ten runs.

5.1 General Information

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 one 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. Also for compactification, a median dividing the x axis of the observed situations was used as the single partition criteria. Note that in the following, the term *hunter performance* mainly refers to the number of steps and success rate, so that a decrease in the number of steps necessary for goal achievement and/or an increase in succes rate implies an improvement in performance.

5.2 Correct Stereotyping

The first group of experiments evaluates hunter performance when a correct stereotype for the prey is available (see Table 1).

5.2.1 Variant Descriptions

The variants for this set of experiments were V1-V6 as shown in Table 1. All these variants had *kill* as a game goal, i.e. a hunter must arrive at the same grid sector as a prey. Variants 1, 2 and 6 had the start configuration (including agent shapes) depicted in Figure 1 - a). Variant 3 substituted the agents in b) for the agents shown in a), but kept the same positioning. Likewise, variant 4 substitued the agents in c) for those in a). Variant 5 had the agent shapes depicted in a) but had random initial placement. With respect to prey strategy, in variant 1 the prey tried to evade the hunter nearest to it. In variants 2, 4 and 5 the preys tried to avoid all hunters and grid borders (with distances measured in various ways). In variants 3 and 6, preys used some sets of SAPs as strategies.

5.2.2 Result Summary

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 give results only slightly worse to when a hunter knows the prey's actual strategy which also is the outcome documented by 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.

5.3 Incorrect Stereotyping and Switching to OSAPs and NNR

The second group of experiments shows the results of using an incorrect prey stereotype with switching to OSAPs at different values of p_{OSAP} (see Table 2).

5.3.1 Variant Descriptions

In this set of experiments, variants 7, 8 and 9 had *kill* as a game goal and variants 10, 11 and 12 had *immobilize*, i.e. a prey must be made unable to perform any action. Variant 7 placed the hunter from a) and the prey from e) [1] randomly. Variant 8 substituted the agents depicted in e) for those in a) without changing positioning. Variant 9 had j) as a start configuration, variant 10 had h), variant 11 had agents in a) replaced with agents in c) and variant 12 started off with i). The preys' strategies used in varaints 7 and 8 were sets of SAPs. In variants 9 and 12, preys tried to avoid all hunters. The preys in variants 10 and 11 tried to avoid all hunters and grid borders.

5.3.2 Result Summary

The main objective of these experiments was to see what can go wrong when 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 versus sticking with the wrong stereotype. Due to the misleading influence of wrong stereotypes, we did not expect better results than those of just using OSAPs and NNR, and naturally we expected to be far off from the results of using a correct model for the prey. 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.

[1] see legend to distinguish between hunter and prey

5.4 Compactification of Observations

The third group of experiments examined the results of a hunter using a compactified set of OSAPs for prey modeling (see Table 3).

5.4.1 Variant Descriptions

All of variants 13 – 20 had *kill* as a game goal. Variants 13 – 17 had a) of Figure 1 as a start configuration. Variants 18 and 19 started off with a random placement of the agents depicted in a). Variant 20 began with the agents of e) in the positions of those in a). For variant 13, the prey strategy was to evade the nearest hunter. The preys in variants 14, 17 and 19 tried to avoid all hunters and grid borders, those in variants 15, 16 and 18 tried to avoid all hunters. The prey strategy of variant 20 was some set of SAPs.

5.4.2 Result Summary

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, expect 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 we tested requires improvement.

5.5 Combination of Concepts

The final group of experiments presents the results of reevaluative stereotyping with switching to OSAPs including compactification of those OSAPs (see Table 4).

5.5.1 Variant Descriptions

The game goal for all of variants 21 – 27 was *kill*. The start configuration of variants 21 – 25 was that of a) in Figure 1, variant 26 had f) and 27 had g). The prey in variant 21 tried to evade the hunter nearest to it. In variants 22, 24 and 27 preys attempted to avoid all hunters. Preys in variants 23, 25 and 26 avoided both hunters and grid borders.

5.5.2 Result Summary

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

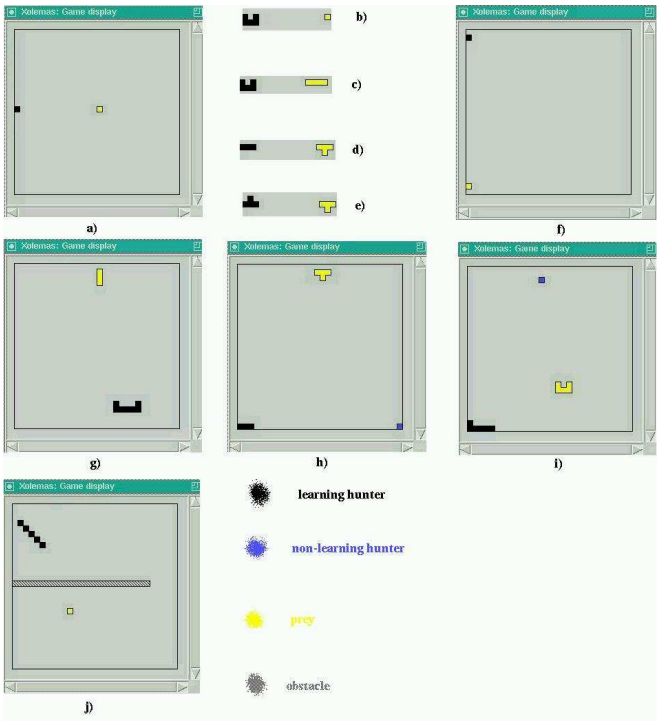

Figure 1: Start Situations

switching decreased T/St in most variants, verifying that the ideas can be combined (as expected).

5.6 Overall Experimental Summary

Our experimental evaluation showed that using a correct prey stereotype for modeling significantly improves hunter performance, and that using switching and reevaluation reduces the potential risks from selecting and using an incorrect stereotype. Moreover, using a compactified set of OSAPs decreases the learning time necessary for modeling using OSAPs and the NNR, but may come at the cost of reduced accuracy. Finally, the combination of the concepts yields results similar to those achieved when operating individually.

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 1: Correct Stereotype Available

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

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

Table 3: Influence of Compactification

V	Only Incorrect Stereotype			With Switching and Full OSAP Set			With Switching and Compactification			$p_{OSAP} = 2$
	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 4: Combination of the Ideas

6 Related Work

Modeling other agents has been researched from several directions. For example, [GD93] describes the Recursive Modeling Method (RMM), which incorporates concepts from utility, decision and game theory into agent modeling. In [KV02], agents are assumed to be homogeneous and predictions can be based on that assumption. [SRV00] assumes some theoretical optimal behavior for modeling other agents. [CM00] uses observations to produce finite automata as models for other agents, while [SG99] represents models of other agents as influence diagrams. In [PL99], coordinating agents use case–based decision theory and communicated meta–level information for learning.

We see connections of our improvements to OSAPs and NNR to [GBS98] and [RV00]. The approach in [GBS98] can be seen as producing probability values for combinations of features that constitute together 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 [RV00], agents used observed data, collected in a fixed time period, 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 our outside of the classification.

Our approach to modeling other agents also has some connections to the area of plan recognition (see [Ca01] for an overview), especially case-based plan recognition. In [KC02], for example, observations are used to fill holes in partial plans, based on similarity of situations, very similar to [DK00]. The use of partial plans has some resemblance to stereotypes, but due to the holes in the plans requiring to be filled, the important effects of having a complete model for prediction after just a few observations, which is achieved by our approach and needed for use by agents, cannot be observed. For an overview of prototype selection and generation techniques in this area, see [WM00].

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 [Sh98] for a brief overview of this area).

7 Conclusion and Future Work

We presented two improvements to modeling of other agents based on observed situation-action pairs and the nearest neighbor rule, both of which enhance the knowledge this basic method has to work with. By starting off with 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 performance of the modeling agent is usually greatly improved. When multiple stereotypes are very similar, periodic reevaluation ensures that the most correct stsreotype 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

[BJD85] M. Benda, V. Jagannathan, and R. Dodhiawalla. An Optimal Cooperation of Knowledge Sources, Technical Report BCS-G201e-28, Boeing AI Center, 1985.

[Ca01] S. Carberry. Techniques for Plan Recognition, *User Modeling and User-Adapted Interaction,* 2001, 11(1-2), pp. 31–48.

[CM00] D. Carmel and S. Markovitch. Opponent modeling in multi-agent systems, In G. Weiss and S. Sen, editors, Adaptation and Learning in Multi-Agent Systems, Lecture Notes in Artificial Intelligence, pp. 40–52. Springer, 1995.

[DF96] J. Denzinger and M. Fuchs. Experiments in Learning Prototypical Situations for Variants of the Pursuit Game, Proc. ICMAS'96, Kyoto, 1996, pp. 48–55.

[DK00] 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.

[GBS98] L. Garrido, R. Brena, and K. Sycara. Towards Modeling other Agents: A Simulation-Based Study, Proc. MABS 1998, Springer, 1998, pp. 210–225.

[GD93] P. Gmytrasiewicz and E. Durfee. Reasoning about Other Agents: Philosophy, Theory and Implementation, Proc. 12th WS on DAI, 1993, pp. 143–153.

[Jo82] R.A. Jones. Perceiving Other People: Stereotyping As a Process of Social Cognition, In A.G. Miller (ed): In the Eye of the Beholder: Contemporary Issues In Stereotyping, Praeger Publishers, 1982, pp. 41–83.

[KC02] 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.

[KV02] J. Kok and N. Vlassis. Mutual Modeling of Teammate Behavior, Technical Report IAS-UVA-02-04, University of Amsterdam, 2002.

[Li22] W. Lippmann. In *Public Opinion, New York: Harcourt*, Brace, 1922, pp. 16–17.

[Ma94] 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.

[PL99] 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.

[RV00] P. Riley and M. Veloso. On Behavior Classification in Adversarial Environments, Proc. DARS 2000, Springer-Verlag, 2000, pp. 371–380.

[Sh98] S. Shapiro. Belief Revision and Truth Maintenance Systems: An Overview and a Proposal, State University of New York at Buffalo, 1998.

[SG99] D. Suraydi and P. Gmytrasiewicz. Learning Models of Other Agents Using Influence Diagrams. Proc. Seventh International Conference on User Modeling (UM-99), pp. 223–232, 1999.

[SRV00] P. Stone, P. Riley and M. Veloso. Defining and Using Ideal Teammate and Opponent Agent Models, In AAAI Press/MIT Press, editor, Proceedings of the Seventeenth National Conference on Artificial Intelligence, 2000, pp. 1040–1045.

[WAD94] 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.

[WM00] R. Wilson and T. Martinez. Reduction Techniques for Instance-Based Learning Algorithms, Machine Learning:38, 2000, pp. 257–286.