Using Cluster-based Stereotyping to Foster Human-Robot Cooperation

Alan R. Wagner, *Member, IEEE*

Abstract—**Psychologists note that humans regularly use categories to simplify and speed up the process of person perception [1]. The influence of categorical thinking on interpersonal expectations is commonly referred to as a stereotype. The ability to bootstrap the process of learning about a newly encountered, unknown person is critical for robots interacting in complex and dynamic social situations. This article contributes a novel cluster-based algorithm that allows a robot to create generalized models of its interactive partner. These generalized models, or stereotypes, act as a source of information for predicting the human's behavior and preferences. We show, in simulation and using real robots, that these stereotyped models of the partner can be used to bootstrap the robot's learning about the partner in spite of significant error. The results of this work have potential implications for social robotics, autonomous agents, and possibly psychology.**

I. Introduction

Psychologists note that humans regularly use categories to simplify and speed the process of person perception [1]. Macrae and Bodenhausen suggest that categorical thinking influences a human's evaluations, impressions, and recollections of the target [2]. The influence of categorical thinking on interpersonal expectations is commonly referred to as a stereotype. For better or for worse, stereotypes have a profound impact on interpersonal interaction [3]. Information processing models of human cognition suggest that the formation and use of stereotypes may be critical for quick assessment of new interactive partners [2]. From the perspective of a roboticist the question then becomes, can the use of stereotypes similarly aid the process of modeling a robot's interactive partner?

This question is potentially critical for robots operating in social environments, such as search and rescue. In environments such as these the robot must quickly determine what role its partner will play in the rescue. Moreover, the robot will not have time to learn how to best help its human teammate during a rescue by iteratively failing and receiving feedback. Rather, the robot will need to bootstrap the process of modeling its interactive partner with information from prior, similar partners.

The overarching goal of our work is to create algorithms that will allow a robot to interact socially with a wide variety of people in a multitude of different social situations. Our previous research has explored methods that allow a robot to iteratively learn a mental model through successive interaction with its human partner [4]. Our work, as well as the research of others [5, 6], has come to the conclusion that this process of creating mental models of humans is critical for behavior prediction [7], determining if a person or robot is being deceptive [8] and whether or not a person is trustworthy [9].

Towards this goal we have developed a framework for social action selection derived from social psychology's interdependence theory [10]. The research presented here contributes a key capability for this framework. Specifically, the algorithm for cluster-based stereotyping developed here provides the information necessary to create the outcome matrix representation of interaction. Recently we have demonstrated that outcome matrices can be used by a robot or agent to reason about deception [8] and about trust [9]. It was assumed, but not shown, that these representations of interaction could be created from the perceptual information available to a robot. The research contributed in this article develops a method that enables a robot to create these representations of interaction. Further, we believe that the approach presented here is the first use of schematic stereotyping on a robot [11]. A schematic stereotype is a class of stereotype in which the perceiver categorizes the partner and uses information about the category to appraise the partner.

The article begins with a review of related work. Next, an introduction to our framework for social action selection is presented followed by a detailed description of the algorithm for cluster-based stereotyping. A description of the different types of stereotyping error and their impact on behavior prediction follows. We conclude with a discussion of the experimental methods used, the experiments conducted, and the results of those experiments.

II. Related Work

Stereotypes and stereotyping have long been a topic of investigation for psychologists [12]. Schneider provides a good review of the existing work [1]. Numerous definitions of the term stereotype exist. Edwards defines a stereotype as a perceptual stimulus which arouses standardized preconceptions that, in turn, influence one's response to the stimulus [13]. Smith and Zarate describes three general classes of stereotype models: attribute-based, schematic-based, and exemplars [11].

With respect to computer science, the inclusion of techniques for stereotyping is not new. Human Computer Interaction (HCI) researchers have long used categories and stereotypes of users to influence aspects of user interface design [14, 15]. The multi-agent systems community has also explored the use of stereotypes. Ballim and Wilks use stereotypes to generate belief models of agents [16].

*This research has been support by the Georgia Tech Research Institute Social Robotics strategic initiative.

A. R. Wagner is with the Georgia Tech Research Institute, Atlanta, GA 30318 USA (404-407-6522; fax: 404-407-8081; e-mail: alan.wagner@gtri.gatech.edu).

Denzinger and Hamdan develop a system by which an agent is tentatively stereotyped, then, after interacting with the target for a period of time, stereotype switching may occur [17]. Their results indicate that the system performs well regardless of the number and quality of stereotypes. Burnett uses stereotypes to gauge an agent's trustworthiness [18].

Investigations of stereotyping with respect to robots are comparatively scarce. Fong et al. used predefined categories of users in conjunction with a human-robot collaboration task [19]. These categories influenced the robot's dialogue, actions, the information presented to the human, and the types of control afforded to the user. Duffy presents a framework for social embodiment in mobile autonomous systems [20]. His framework includes methods for representing and reasoning about stereotypes. He notes that stereotypes serve the purpose of bootstrapping the evaluation of another agent and that the perceptual features of the agent being stereotyped are an important representational consideration.

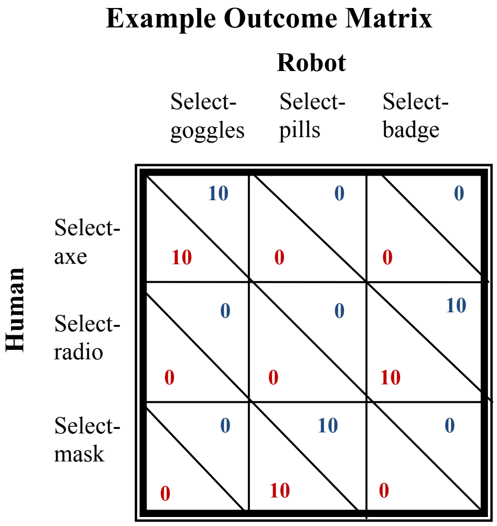
III. PARTNER MODELING

The outcome matrix (fig. 1) is a standard computational representation for interaction [21]. The term interaction describes a discrete event in which two or more individuals select interactive behaviors as part of a social situation or social environment. The term individual is used to indicate a human, a social robot, or an agent. We focus on interaction involving two individuals—dyadic interaction.

Because outcome matrices are computational representations, it is possible to describe them formally. A representation of interaction consists of 1) a finite set N of interacting individuals; 2) for each individual $i \in N$ a nonempty set A^i of actions; 3) the utility obtained by each individual for each combination of actions that could have been selected [21]. Let $a_j^i \in A^i$ be an arbitrary action j from individual i's set of actions. Let $\left(a_j^1, \dots, a_k^N\right)$ denote a combination of actions, one for each individual, and let u^i denote individual i's utility function: $u^i\left(a_j^1, \dots, a_k^N\right) \dashrightarrow \Re$ is the utility received by individual i if the individuals choose the actions $\left(a_j^1, \dots, a_k^N\right)$.

A mental model is a term used to describe a person's concept of how something in the world works [22]. We use the term partner model (denoted m^{-i}) to describe a robot's mental model of its interactive human partner. We use the term self model (denoted m^i) to describe the robot's mental model of itself. The superscript *-i* is used to express individual *i*'s partner [21].

To create outcome matrices a source of information must exist which can populate the matrix representation. The partner model and the self model serve this purpose. Further, the information needs of the outcome matrix representation can inform us as to how to construct the partner model. Thus, our partner model contains three types of information: 1) a set of partner features $\left(f_1^{-i}, \dots, f_n^{-i}\right)$; 2) an action model, A^{-i}; and 3) a utility function u^{-i}. Partner features are perceptual features used for partner recognition. These features allow the robot to recognize the person in subsequent interactions. The action model contains a list of actions available to that individual. The utility function includes information about the outcomes obtained by that individual when the robot and the human select a pair of actions. Information about the partner's beliefs, knowledge, personality, etc. could also conceivably be included in these models but were not included in the research described here.



	Select-goggles	Select-pills	Select-badge
Select-axe	10 10	0 0	0 0
Select-radio	0 0	0 0	10 10
Select-mask	0 0	10 10	0 0

Figure 1. An example outcome matrix is depicted above. This outcome matrix represents a coordination game in which the robot and the human only receive positive outcome if they select complimentary objects.

Like the partner model, the self model also contains an action model and a utility function. The action model in this case includes a list of actions available to the robot. Similarly the robot's utility function includes information about the robot's own outcomes.

The preceding discussion raises an important question: how can partner models be compared to one another? For example, how close is the partner model, m^{-i}, that the robot learned during several interactions with a human to the actual model, ${}^{*}m^{-i}$, that the person was using to select actions? We address this problem by viewing action models and utility functions as sets. The action model is a set of actions and a utility function is a set of triplets, $\langle a^i, a^{-i}, r \in \Re \rangle$, containing the action of each individual and a resulting utility value. If the contents of a partner model are viewed as the elements of a set, then the use of set theoretic measures of distance to compare different partner models is possible. The Jaccard index,

$$J(m^A, m^B) = \frac{|m^A \cup m^B| - |m^A \cap m^B|}{|m^A \cup m^B|}, \qquad (1)$$

is one measure of set distance [23]. Utility function comparisons were considered to be equal if the actions were the same and the outcome values were within 1 of each other. Because of its simplicity, this measure of distance was used in our implementation of the **create stereotypes algorithm** presented below.

But how does a robot learn a partner model? Perhaps the most simplistic method for learning a model of a partner is to just interact with the person, observe their features and action selections, and store this information in the partner model. We have shown in previous work that a robot could eventually learn a model of its interactive partner, assuming that the partner's model was static [4].

IV. STEREOTYPED PARTNER MODELS

With respect to this framework, a stereotype is a type of generalized partner model used to represent a collection or category of individual partner models. Thus, the creation of stereotypes requires the computation of these generalized partner models. Moreover, to be useful, techniques must be developed that allow for matching of a new interactive partner's perceptual features to an existing stereotype. Stereotype creation is therefore a two phase process. First, partner models are clustered with the centroids of the clusters becoming the partner model stereotype. Next, using the cluster centroids as data, a mapping from partner features to the stereotypes is learned. Our implementation utilizes agglomerative clustering for the first phase and C4.5 decision trees for the second phase. The following section details the stereotype creation process.

A. Creating Stereotypes

The **create stereotypes algorithm** (fig. 2 top) takes as input a new partner model. This input is optional. The algorithm can also be run on the robot's existing history of partner models (termed the model space). Individual partner models are learned by successively interacting with an individual and updating a model with the results from the interaction.

Initially the robot has no partner models at all in its model space. The robot seeds its model space with a model of itself, its self model. Conceptually, this act of seeding with the self model allows the robot to equate its partner's actions and preferences to its own actions and preferences. The software that we created to use the algorithm is capable of saving any partner models that the robot learned as serialized data. Hence, because the robot could always load its previously saved partner models, seeding of the model space was only necessary at the beginning of an experiment.

Once a model of a new partner has been learned, the first step of the algorithm adds the new model to the model space. Next, in lines 2 and 3, each model in the space is assigned to a unique cluster. Lines 4 and 5 perform agglomerative clustering, iterating through each cluster and, if the clusters meet a predetermined distance threshold, merging them. Equation (1) from section III is used to determine the distance between two clusters. The cluster centroids that remain after step four are stereotypes, denoted $s_1, \dots, s_n$. This list of stereotype models is saved by the robot.

In the next phase, the C4.5 algorithm is used to create a mapping, denoted ψ, from the partner's perceptual features to stereotypes. Line 7 from fig. 2 (top) creates data for the learning algorithm by pairing each model's perceptual features to its associated stereotype. In the final steps, this data is used to train a classifier mapping partner features to the stereotyped model.

The **create stereotypes algorithm** makes two important assumptions. First, it assumes the existence of a distance function capable of measuring the difference between two partner models. Equation (1) was presented as a method for measuring partner model distance (see section III). If, however, additional information (such as the partner's beliefs, motivations, goals, etc.) is added to the partner model, then a more elaborate distance function may be necessary. Second, the algorithm assumes that partner models can be merged to create new partner models. In order to merge a partner model one must merge the components of the partner model. For this work, that meant merging the action models and utility functions. Action models were merged by adding an individual action to the combined model only if the action was included in at least half of the data that composed the merged model. For example, if the merged model was created from ten individual partner models and an action existed in four of the models then it was not included in the merged model. If, however, the action existed in five of the models then it was included in the merged model. Similarly, merged utility values were derived from the average utility value of the composition utility functions.

```
Create Stereotypes Algorithm
Input:   Partner model $m^{-i}$.
Output:  Classifier $\psi$ mapping $m^{-i}$(features) to a
stereotype.

Cluster phase
1. Add $m^{-i}$ to partner model space
2. for all models in model space
3.     make a cluster
4. while centroid-distance $(c_1, c_2) < k$
5.     merge-clusters$(c_1, c_2)$

Function learning phase
6. for all models n in model space
7.     set data_j←make-pair($m_j$(features), centroid_j)
8  $\psi$ ←train-classifier( data )
9. return $\psi$

Match to Stereotype Algorithm
Input:Partner features $f_1^{-i}, \dots, f_n^{-i}$.
Output: Partner model $m^{-i}$.

1. convert $f_1^{-i}, \dots, f_n^{-i}$ to instance of classifier data
2. result ← $\psi$(classify( instance ))
3. $m^{-i}$← StereotypeList( result )
4. return $m^{-i}$
```

Figure 2. Algorithms for creating stereotypes and for matching newly perceived individuals to existing stereotypes. The **create stereotypes algorithm** operates by clustering partner models and then constructing a classifier mapping a partner's perceptual features to a stereotype. The **match to stereotype algorithm** simply uses the classifier to match the partner's perceptual features to the closest stereotype

B. Matching To Stereotypes

When perceiving a new interactive partner, the robot matches the new person's perceptual features to an existing stereotype. This process begins by converting the partner's features into an instance of data for the classifier and then using the classifier to select the correct model (fig. 2 bottom). Line 1 from fig. 2 (bottom) uses the partner's features to create an instance for classification. The result is matched to a stereotype (lines 2 and 3).

In previous work we found that use of the cluster-based algorithm for stereotyping required 5.75 fewer interactions to obtain an 80 percent rate of correct partner action prediction when compared to relearning a new model for each individual [10]. This previous research did not, however, consider the impact of stereotyping errors.

V. STEREOTYPING ERRORS

This section investigates the different types of error that can occur when a robot uses stereotypes to predict the behavior of a specific individual. It is important to examine the types and impact of stereotyping errors. Because the robot uses these models to predict and react to its human partner, errors might cause a robot to interact in a manner that is extremely detrimental to the human. Consider, as a running example, a robot tasked with learning a stereotype of a firefighter. In this case, the partner features might include perceptual information related to the person's uniform, appearance, height, weight, etc. The learned partner model would contain information related to the person's preferred actions, such as putting out fires, rescuing disaster victims, etc. Errors could potentially cause a robot to not identify a person as a firefighter, possibly leading the robot to not provide necessary information or to select the incorrect action.

Perception is one source of error. The robot may misperceive the features that describe the partner's appearance or, alternatively, it may misperceive the information contained within the partner model, such as the action being performed. For example, if the robot incorrectly perceived the firefighter's brown uniform as blue, this would be a partner feature error. Partner feature errors could potentially cause the individual to be incorrectly categorized. On the other hand, if the robot incorrectly perceives the action of putting out a fire as the action of making an arrest, this is a perceptual modeling error. In this case, the robot's action model for that particular firefighter would indicate that the firefighter makes arrests. Modeling errors could potentially cause the robot's stereotype to be inaccurate with respect to the true stereotype.

Stereotyping error can result from statistical anomalies. Ideally, the stereotype model is created from individuals that are representative of the category. It is possible, however, that the individuals from which the stereotype is created are actually outliers with respect to the overall category. Here the stereotype created is not representative of the category. Consider, for example, a robot that creates a stereotype based on models it has learned from interactions with firefighters. Rather than put out fires and rescue victims, these particular firefighters act as police officers, making arrests and writing tickets. Because the robot's stereotype has been created from outliers of the overall category, predictions based on the stereotype will be incorrect when the robot interacts with a non-outlier member of the category.

Alternatively, the robot may create a stereotype model from non-outlier or a mixture of outlier and non-outlier individuals. In this case, the stereotype will be representative of the overall category. Yet, when using the stereotype to predict the behavior of a newly encountered individual the robot may then encounter an outlier. These errors occur because although the robot's stereotype is representative of the category, the newly encountered individual is an outlier. Hence the stereotype does not accurately reflect the new individual's behavior. For example, this error would occur if the robot were to create a stereotyped model based on actions and utilities of a typical firefighter and were to then encounter an individual that is dressed like a firefighter but acts like a police officer.

One final type of error that can occur is an error in generalization. The cluster-based algorithm for stereotype creation adds actions to the stereotype model if the action occurs in more than half of the individual models that compose the stereotype. Thus, the stereotype generalizes an action to all of the members of a category even if the action is only valid for just over half of the individuals. In the worst case, just under half of the individuals in the category may have an action falsely prescribed to them by the stereotype. For example, if the action, "likes to cook", occurs in 51 percent of the firefighters used to create the robot's stereotype of a firefighter, then the action will be included in the centroid and, therefore, generalized to anyone that perceptually resembles a firefighter. Table I lists each type of potential stereotyping error and its characteristics.

TABLE I. TYPES OF STEREOTYING ERRORS

Type	Name	Error Description	Experiments
1	Partner feature error	Misrecognition of partner	No
2	Modeling error	Misrecognition of information in partner model	Yes
3	Unrepresentative stereotype	Individuals used to create stereotype are outliers	No
4	Outlier error	Individual encountered is an outlier	Yes
5	Generalization error	Unfounded generalization	No

In general, all of these errors can potentially result in faulty action prediction. We argue, however, that some of these errors are less likely to be a serious problem for robotic stereotyping applications than others. Errors related to partner features (type 1 from Table I) may be mitigated by the fact that a large amount of perceptual data can be collected in relation to a partner's perceptual features over the course of a single interaction. In just 60 seconds, for example, the robot can potentially collect 1800 frames of visual data related to the partner's visible features. Regardless of the perceptual modality, data collection related to partner features can be collected over the entire course the interaction. Hence, we argue, that missing and incorrect partner features are less likely to be a significant source of error for most stereotyping applications. Errors related to stereotypes created from outliers (type 3 from Table I), we further argue, will not be a significant source of error. By description these errors occur only when the robot's stereotype is created from models based on interactions with several outliers. Yet outliers, by definition, should not be encountered often. Hence, it is not likely that the robot will encounter several outliers. Because the centroid is created from averaged models, occasional outliers will not impact performance. Because of space, we do not explore generalization errors in this paper.

In the sections below we examine the two remaining types of error, modeling errors (type 2 from Table I) and outlier errors (type 4 from Table I), that we believe could be an important source of error during social robotics applications such as search and rescue.

VI. Empirical Evaluation

Social psychologists claim that interaction is governed by three variables: 1) the first interacting individual; 2) the second interacting individuals; and 3) the environment [24]. Given our goal of creating a social robot that could interact with any person and in a variety of environments, we needed an evaluation method that allowed for control of both the robot's interactive partner and the social environment while allowing us to observe the behavior produced by the robot.

For the field of social psychology, experiments often involve the use of a controlled environment in which an experimental subject interacts with a confederate of the experimenter [25]. Often both characteristics of the environment or of the confederate serve as independent variables that the researcher can manipulate. The behavior of the experimental subject, the robot in this case, is generally observed and recorded as the dependent variable. These types of empirical evaluations tend to be high on internal validity, meaning that their results are indicative of a causal relationship. Unfortunately, they also tend to be low on external validity, meaning that the results from these types of experiments do not typically generalize well beyond the experimental conditions.

For social robotics research, experiments that treat the robot as an experimental subject and the human that the robot interacts with as a confederate of the experimenter offer several advantages. It may allow researchers a controlled method for determining the causal impact of their algorithms on the robot's social behavior. Hence, the researcher may thus be able to first verify the internal validity of their techniques before engaging in externally valid experiments involving unstructured interaction.

Like social psychology experiments, experimental data is obtained from video recording of interactions with the robot. The video is used to record the robot's actions and responses. This video is augmented with data saved from the robot's memory relating to its perception of the partner features, objects available for selection, and action selections.

A. Experimental Setup

A coordination game was used for both the simulation and robot experiments. A coordination game is a game-theoretic social situation in which both individuals receive maximal reward only if they select coordinating actions [21]. Figure 1 depicts an example of an outcome matrix representing a coordination game. In this example, both individuals receive an outcome of 10 if they select action pairs (*select-goggle*, *select-axe*), (*select-badge*, *select-radio*), or (*select-pills*, *select-mask*) and 0 outcome if any other action pair is selected.

The notional scenario for the experiments is a situation in which a robot acts as a cooperative assistant to a human. In this scenario, the robot must select the best tool to assist its human partner. Figure 3 depicts the setup for the robot experiments. The robot selects from among the tools to its right (blue circle fig 3). The human selects from the three tools to the robot's left (red circle fig. 3). The robot and human receive the maximal outcome if and only if they select a matching pair of tools. Table II lists all of the tools used in these experiments and the groupings of matching tools. In order to receive maximal outcome the robot needed to predict the tool that the person was going to select and to then select the tool that matched.

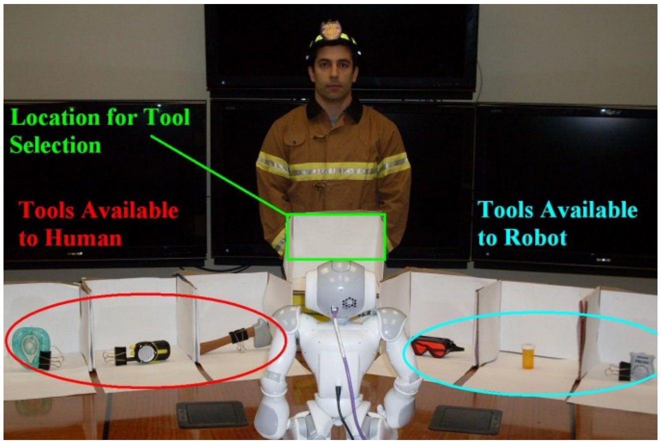


Figure 3 The experimental setup is depicted above. The objects in the blue circle are tools available to the robot in this interaction. The objects in red circle are objects that are available to the human in this interaction. The green square depicts the area where the human places tools that he or she has chosen. No tools have been chosen yet during the depicted interaction.

TABLE II. Groupings of Tool Types

Type	Tools				
1	Extinguisher	Axe	Flashlight	Helmet	Goggles
2	Antiseptic	Mask	Neckbrace	Pills	Bandage
3	Binoculars	Radio	Handcuffs	Badge	Batton

B. Human Confederates

Laboratory experiments involving controlled human behavior are standard in many psychology experiments [8]. These experiments typically require that the experimenter's confederate act in a specific manner.

The robot's interactive partner dressed and, within the limits of the experiment, acted like a firefighter. In both the simulation experiments and the robot experiments, the robot or agent was capable of perceiving a predefined list of features and feature values (Table III). In the simulation experiments the values for the features were given to the simulated robot. In the robotics experiment, some of the values for the features were determined by having the robot ask the confederate questions such as "Are you male or female?" Others were captured visually. Table III lists those that were spoken and those that were visual. To generate the visual features for the real robotics experiment, the confederate had to dress in a Halloween costume (fig. 3). Rather than seek the assistance of 30 confederates we used false beards, wigs, and differences in attire to create the appearance, based on the visual limitations of the robot, of 30 different individuals. Only one person acted as a confederate for the robot experiment. Improvements in perception may necessitate the use of different people playing the role of a confederate in the near future. The person acted like a firefighter by selecting the tools denoted as type 1 in Table II.

TABLE III. PARTNER FEATURES AND POSSIBLE VALUES

Feature Name	Values	Verbal or Visual?	Robot Exp
Badge	yes, no	visual	Yes
Uniform color	brown, green, blue	visual	Yes
Head Gear	yes, no	visual	Yes
Head Gear Color	black, green, blue	visual	Yes
Hair Color	black, blonde, red	visual	Yes
Beard	yes, no	visual	Yes
Facial Symmetry	highly, symmetric, asymmetric	visual	No
Facial Length	very wide, square, long, very long	visual	No
Skin Color	light, dark	verbal	Yes
Glasses	yes, no	verbal	Yes
Age	young, old, medium	verbal	Yes
Body Type	thin, heavy, medium	verbal	Yes
Height	tall, small, medium	verbal	Yes
Gender	male, female	verbal	Yes

VII. EXPERIMENTS

A. *Simulation Experiment*

We conducted a numerical simulation to evaluate the impact of modeling error (error type 2 from Table I) and outlier error (error type 4 from Table I). A numerical simulation of interaction focuses on the quantitative results of the algorithms and processes under examination and does not attempt to simulate aspects of the robot, the human, or the environment. As such, this technique offers advantages and disadvantages as a means for discovery. One advantage of a numerical simulation experiment is that a proposed algorithm can potentially be tested on thousands of outcome matrices representing thousands of social situations. This allows one to evaluate the statistical significance of the results. One disadvantage is that, because it is not tied to a particular robot, robot's actions, human, human's actions, or environment, the results, while extremely general, have not been shown to be true for any existent social situation, robot, or human.

This experiment simulated the tool selection game described in section VI. Confederates of the experimenter were simulated by providing the robot with a list of perceptual features from Table III representing a nominal person. The robot used this information in conjunction with the algorithms for creating and matching stereotypes (fig 2.) to obtain a stereotyped partner model of the person. This partner model was then used to predict the tool that would be selected by the simulated person. Based on this prediction the robot made its own tool selection. Finally, the simulated human selected their tool in accordance with the experimental condition and a numerical outcome value was awarded.

The dependent variable in this experiment was the mean number of correct coordinations performed by the simulated human and the robot. For this scenario, correct coordination's represent a measure of task success. The independent variable was the type of error introduced. Four different conditions were examined. In the first, baseline condition, no error was introduced. In the second condition modeling error (type 2 from Table I) was added at a rate of 50 percent. Modeling error was introduced by giving the robot a 50 percent chance of incorrectly perceiving the human's action selection. In the third condition, outlier error (type 4 from Table I) was added at a rate of 20 percent. Outliers, although perceptually similar to a firefighter, consistently selected tools not associated with firefighting (types 2 and 3 from Table II). In the final condition, both types of errors were introduced at their respective rates. Thirty trials of the experiment were run in order to obtain statistical significance. A single trial consisted of 4 interactions in the game with 15 different individuals. These experiments were conducted on a standard Dell laptop computer.

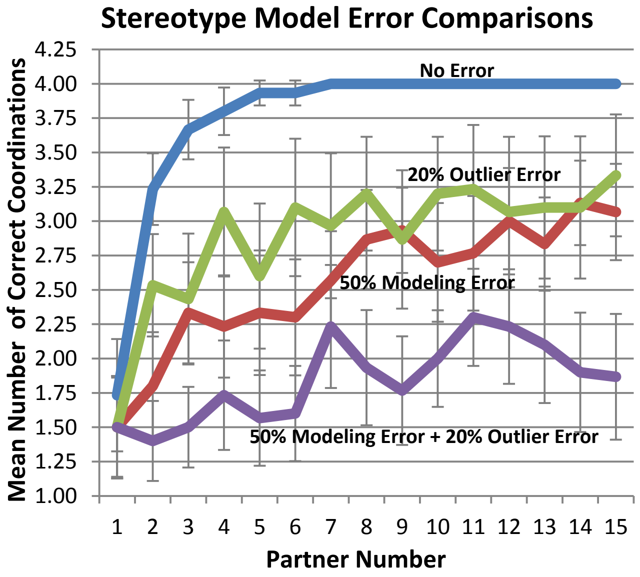


Figure 4 The results from the simulation experiment are depicted above. The blue line (top) depicts the no error condition, the green line (second) depicts the outlier condition, the red line (third) the modeling error condition, and the purple bottom line the modeling error and outlier error condition.

The results show that the introduction of error in all three of the error conditions significantly $(p < 0.01)$ impacts the robot's performance when compared to the no error condition for all partners after the second partner. We hypothesized that modeling error would result in significantly greater error than would outliers. The results, however, do not indicate a significant difference in performance between the modeling error condition and the outlier condition. This may be, in part, due to the fact that when the robot encounters an outlier it consistently fails to select the correct tool. Over the course of all 30 trials, because outliers were randomly selected, the net impact on performance is similar to the modeling error condition. Still, as will be seen in the next experiment, the impact of an outlier on performance manifests itself as a onetime error event, whereas modeling error resembles a constant level of noise. We further hypothesized that the error related to the modeling error condition would generally decrease with each new partner. We reasoned that, in spite of an error rate of 50 percent, additional interaction with new partners would direct the robot's stereotype centroid towards convergence on the true type. The results support this hypothesis. Finally, if we define reasonable performance as an average of 3 correct coordinations out of 4, then results indicate that both the modeling error only and outlier conditions obtained a reasonable level of performance over the course of the experiment.

B. *Robot Experiment*

Robot experiments were conducted on an embodied, situated NAO robot to confirm and further investigate the results from the simulation experiment. The NAO robot is a Humanoid platform with 25 degrees of freedom, integrate speech synthesis and recognition capabilities, and a camera.

The robot experiments utilized the same notational scenario already described, namely the tool selection coordination task with the robot acting as an assistant to a nominal human firefighter. For these experiments the experimenter dressed in a firefighter costume which included wigs and false beards in order to simulate different individuals. The specific features for each individual, such as whether or not the person had a beard, were determined at random before the experiment was conducted. Moreover, the tools available for selection by the robot and the person were chosen and placed at random.

At the beginning of the experiment, the robot searches for a face. Once it finds a face it collects the visual partner features from Table III. If there are any features that cannot be determined, then the robot records these features as undetermined. Next, the robot queries the firefighter for the verbal partner features. The robot then turns its head to determine which tools are available to it and to the firefighter. Next, it uses a stereotyped partner model, if one exists, to predict which tool the person will select. It uses this information to select a tool for itself and verbally states its preference. The firefighter then indicates which tool he or she has chosen by placing the tool in the central box (fig 3). Finally, after noting which tool the firefighter has selected, the robot waits for the person to either setup the tools for the next interaction or for a new individual to appear.

Over the course of the experiment the robot interacted with 15 different notional firefighters in four different coordination games each. The dependent variable in this experiment was the number of correct coordinations. The independent variable consisted of two conditions. The first condition was a no-error condition in which the firefighter always selected the tool matching the firefighter type. The second condition, like the simulation, included error in the form of outliers. Prior to the experiment outliers were selected at random with each partner having a 20 percent chance of being labeled an outlier. The result was that partners 6, 11, and 14 were selected as outliers.

Fig 5 depicts the results from the real robot experiment. In contrast to the graph in fig 4, the x-axis in fig 5. indicates the interaction number. Hence every 4 interactions represent a new partner. There were 60 interactions (15 individuals times 4 interactions per individual) total. The y-axis indicates the cumulative outcome received assuming +1 outcome for a correct coordination and -1 outcome for an incorrect coordination. A single trial again consisted of 4 interactions in the game with 15 different individuals. Only one trial was run. Thus, it was not possible to determine the statistical significance of the results. Analysis of the data collected indicated that 19.4% of the partner features were not recognized and another 8.5% were incorrectly recognized.

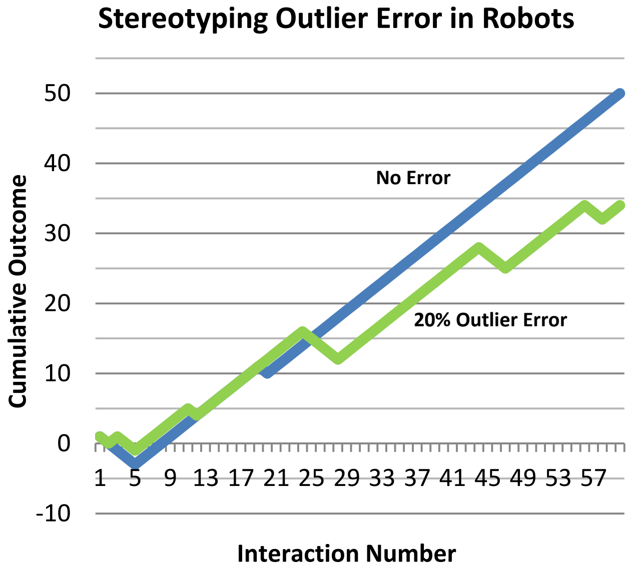


Figure 5 The results from the robot experiment are depicted above. The blue line (top) depicts the no error condition, the green line (bottom) depicts the outlier error condition.

As with the simulation experiment, the robot experiment indicates that the robot performs poorly (zero cumulative outcome during the first 6-9 interactions) in both conditions initially as it learns the stereotypes. In the no error condition, once the robot has created the stereotype it consistently picks the correct tool. In the outlier condition, the robot's performance matches the no error condition whenever it is not interacting with an outlier. The robot encounters its first outlier at interaction 24. In this case, for each of the 4 interactions, the robot incorrectly predicts the human's action selection, resulting in a cumulative outcome for this individual of -4. When the robot encounters the second outlier (at interaction number 44) it receives an outcome of -3. The final outlier (interaction 56) results in an overall outcome of -2.

VIII. Conclusion

This article contributes an algorithm for stereotype creation which utilizes cluster centroids as a stereotype representation. The **create stereotypes algorithm** clusters individual partner model information to create a stereotyped model that the robot can then use during its initial interactions with a new, unknown individual. Even though our algorithm for cluster-based stereotyping is created from well known techniques from machine learning, we believe that the overall algorithm, its use to create generalized models of people, and its relation to similar methods from psychology represent a novel and important area of research. In previous work we have shown that use of these stereotyped models results in improved model accuracy during early interaction with a new partner [10]. The research presented here examined the types of errors that can impact stereotype creation and usage. The results indicate that although the errors examined did impact the robot's performance, only the condition which included both types of errors simultaneously resulted in unacceptable performance (50 percent incorrect). Moreover, for the modeling error condition, performance continued to improve throughout the experiment.

The **create stereotypes algorithm** assumes that a learnable pattern of partner characteristics exists. The psychological literature indicates that this is the case and that humans regularly use this information to categorize and make predictions about their own interactions [1] [2].

The development of suitable distance measures may be an issue moving forward with this research. Ideally the robot's mental model of its human partner will be a rich source of information that would allow the robot to make accurate predictions about the person's future behavior. Information such as the person's beliefs, habits, or attitudes could conceivably be included in a partner model. It is not readily apparent, however, how a distance measure could be constructed to include this information.

Scalability is another challenge. The problems we examined consisted of 14 partner features and 15 potential actions. It has yet to be determined whether or not the techniques presented here will work when hundreds of partner features are available and the partner models contain hundreds of actions and utilities.

The research presented here is a small portion of a larger effort. To date we have filmed 270 minutes worth of experiments associated with different aspects of human-robot interaction and stereotyping. In addition to firefighters, we have created stereotyped models of police officers and EMTs. Some of our ongoing experiments examine the use of stereotype models as a source of inference about a new person and the inclusion of situation specific characteristics determining the appropriateness of a stereotype. Future work may also explore potential applications of this research in the areas of home healthcare and security.

Acknowledgment

The author would like to thank Zsolt Kira for his helpful comments.

References

[1] D. J. Schneider, *The Psychology of Stereotyping*. New York, New York: The Guilford Press, 2004.

[2] C. N. Macrae and G. V. Bodenhausen, "Social Cognition: Thinking Categorically about Others," *Annual Review of Psychology*, vol. 51, pp. 93-120, 2000.

[3] J. A. Bargh, M. Chen, and L. Burrows, "Automaticity of social behavior: direct effects of trait construct and stereotype activation on action," *Journal of Personality and Social Psychology*, vol. 71, pp. 230-44, 1996.

[4] A. R. Wagner, "Creating and Using Matrix Representations of Social Interaction," in *Human-Robot Interaction (HRI)*, San Diego, CA, 2009.

[5] J. Trafton, et al., "Enabling effective human–robot interaction using perspective taking in robots," *IEEE Transactions Systems, Man, Cybernetics A*, vol. 35, no. 4, p. 460–470, 2005.

[6] G. Briggs and M. Scheutz, "Facilitating Mental Modeling in Collaborative Human-Robot Interaction through Adverbial Cues," in *12th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Portland, Oregon, 2011, p. 239–247.

[7] J. K. Rilling, et al., "A Neural Basis for Social Cooperation," *Neuron*, vol. 35, pp. 395-405, Jul. 2002.

[8] A. R. Wagner and R. C. Arkin, "Robot Deception: Providing Robots with the Capacity for Deception," *International Journal of Social Robotics*, 2010.

[9] A. R. Wagner and R. Arkin, "Recognizing Situations That Demand Trust," in *IEEE International Symposium on Robot and Human Interactive Communication*, Atlanta, GA, 2011.

[10] A. R. Wagner, *The Role of Trust and Relationships in Human-Robot Social Interaction*. Ph.D. diss., School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, 2009.

[11] E. R. Smith and M. A. Zarate, "Exemplar-Based Model of Social Judgement," *Psychological Review*, pp. 3-21, 1992.

[12] L. Terveen, "An overview of human–computer collaboration," *Knowledge-Based Systems*, vol. 8 , no. (2–3), 1994.

[13] A. L. Edwards, "Studies of Stereotypes: I. The directionality and uniformity of responses to stereotypes," *Journal of Social Psychology*, vol. 12, pp. 357-366, 1940.

[14] E. Rich, "User Modeling via Stereotypes," *Cognitive Science*, vol. 3, no. 197, pp. 329-354, 1979.

[15] R. Kass and T. Finin, "General User ModelingD: A Facility to Support Intelligent Interaction," in *Intelligent User Interfaces*. ACM Press, 1991..

[16] A. Ballim and Y. Wilks, "Beliefs, stereotypes, and dynamic agent modeling," *User Modeling and User-adapted Interaction*, vol. 1, no. 1, pp. 33-65, 1991.

[17] J. Denzinger and J. Hamdan, "Improving observation-based modeling of other agents using tentative stereotyping and compactification through kd- tree structuring," *Web Intelligence and Agent Systems: An International Journal*, vol. 4, no. 3, pp. 255-270, 2006.

[18] C. Burnett, T. J. Norman, and K. Sycara, "Bootstrapping trust evaluations through stereotypes," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, Toronto, CA, 2010.

[19] T. Fong, C. Thorpe, and C. Baur, "Collaboration, dialogue, and human–robot interaction," in *Proceedings of the International Symposium on Robotics Research*, 2001.

[20] B. Duffy, "Robots Social Embodiment in Autonomous Mobile Robotics," *International Journal of Advanced Robotic Systems*, vol. 1, no. 3, 2004.

[21] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. Cambridge, MA: MIT Press, 1994.

[22] D. Norman, *Some Observations on Mental Models*, D. Genter and A. Stevens, Eds. Hillsdale, New Jersey: Erlbaum Associates, 1983.

[23] P. Jaccard, "Étude comparative de la distribution florale dans une portion des Alpes et des Jura," *Bulletin de la Société Vaudoise des Sciences Naturelles*, vol. 37, p. 547–579, 1901.

[24] C. E. Rusbult and P. A. M. Van Lange, "Interdependence, Interaction, and Relationships," *Annual Review of Psychology*, vol. 54, pp. 351-375, 2003.

[25] M. Mitchell and J. Jolley, *Research Design Explained*, 2nd ed. Orlando, FL: Harcourt Brace Jovanovich, 1992