ANTICIPATION GUIDED PROACTIVE INTENTION PREDICTION FOR ASSISTIVE ROBOTS

by

Joshua Burmeister
A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Mechanical Engineering).

Golden, Colorado

Date __________________________

Signed: _________________________
Joshua Burmeister

Signed: _________________________
Dr. Xiaoli Zhang
Thesis Advisor

Golden, Colorado

Date __________________________

Signed: _________________________
Dr. Greg Jackson
Professor and Head
Department of Mechanical Engineering
ABSTRACT

When a person is performing a task, a human observer usually makes guesses about the person’s intent by considering his/her own past experiences. Humans often do this when they are assisting another in completing a task. Making guesses not only involves solid evidence (observations), but also draws on anticipated evidence (intuition) to predict possible future intent. Benefits of guessing include, quick decision making, lower reliance on observations, intuitiveness, and naturalness. These benefits have inspired a proactive guess method that allows a robot to infer human intentions. These inferences are intended to be used by a robot to make predictions about the best way to assist humans. The proactive guess involves intention predictions which are guided by future-object anticipations. To collect anticipation knowledge for supporting a robot’s intuition, a reinforcement learning algorithm is adopted to summarize general object usage relationships from human demonstrations. To simulate overall intention knowledge in practical human-centered situations to support observations, we adopt a multi-class support vector machine (SVM) model which integrates both solid and anticipated evidence. With experiments from five practical daily scenarios, the proactive guess method is able to reliably make proactive intention predictions with a high accuracy rate.
TABLE OF CONTENTS

ABSTRACT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . iii
LIST OF FIGURES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . vi
LIST OF TABLES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . viii
ACKNOWLEDGMENTS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ix
1 INTRODUCTION . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
   1.1 Motivation and Focus . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
   1.2 Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
2 KNOWLEDGE LEARNING . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
   2.1 Proactive Guess - General Methodology . . . . . . . . . . . . . . . . . . . . 7
   2.2 Training Data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
   2.3 Object-Object Learning With RL . . . . . . . . . . . . . . . . . . . . . . . . . 9
   2.4 Object-Intention Learning With SVM . . . . . . . . . . . . . . . . . . . . . . . 15
3 PROACTIVE GUESS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
   3.1 Testing Data Collection . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
   3.2 Proactive Guess - Combining RL with SVM . . . . . . . . . . . . . . . . . . 21
4 RESULTS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26
   4.1 Experiment Set-Up . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26
   4.2 Result Representation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
   4.3 Initial Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
      4.3.1 Low-Range Threshold Results . . . . . . . . . . . . . . . . . . . . . . . 32
      4.3.2 Mid-range Threshold Results . . . . . . . . . . . . . . . . . . . . . . . . 33
      4.3.3 High-range Threshold Results . . . . . . . . . . . . . . . . . . . . . . . 36
   4.4 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36
5 FUTURE WORK . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
   5.1 Object Inclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
LIST OF FIGURES

Figure 2.1 A simple example for reward system testing. The goal of the agent is to find the path from the green node to the red node (left). The optimal path is shown in yellow (right). .......................................................... 11

Figure 2.2 A heatmap representing the Q-matrix for the *DrinkingLiquid* intention. Magenta colored nodes indicate object-object pairs that are highly related. ................. 12

Figure 2.3 The same heatmap of the *DrinkingLiquid* Q-matrix shown above, but with the top two object-object relationships highlighted in each row. The value ten is given to the highest in each row, and the value five is given to the second highest relationship. .................................................. 13

Figure 2.4 An example of a traditional reward system (left) compared to the bias reward system that was experimented with (right). ................................. 14

Figure 2.5 A graph showing the cost and duration each reward system needed to converge to an optimal path. As expected the reward system with the largest bias converged the quickest and with the least cost. ................................. 15

Figure 2.6 A representation of four additional tested reward systems. ................................. 16

Figure 2.7 Results from the four tested reward systems. It was found that a bias that tends towards the end state allows the agent to converge the fastest and with the least cost. .................................................. 16

Figure 3.1 Screenshots from collected training video. .................................................. 20

Figure 3.2 The participant is reaching for soap to pour on the sponge, but the soap is out of the camera’s view. .................................................. 21

Figure 3.3 The participant uses a plastic container, however, this object is not an element in the list of available objects. .................................................. 22

Figure 3.4 A cupboard is used, however, cupboard is not an object in the set of available objects and is not closely related to any. .................................................. 22

Figure 3.5 A flowchart representation of the implemented proactive guess mechanism. .................................................. 25

Figure 4.1 A flowchart representation of the SVM-only comparison model. ................................. 27

Figure 4.2 Proactive intention prediction accuracy for initial threshold for proactive guessing. .................................................. 29

Figure 4.3 Intention prediction speeds for initial threshold value for proactive guessing. ................................. 30
Figure 4.4  Proactive intention prediction accuracy for initial threshold value for SVM-only model. ......................................................... 30

Figure 4.5  Intention prediction speeds for initial threshold value for SVM-only model. ......................................................... 31

Figure 4.6  Proactive intention prediction accuracy for low-range threshold value for proactive guessing. ......................................................... 32

Figure 4.7  Intention prediction speeds for low-range threshold value for proactive guessing. ......................................................... 33

Figure 4.8  Proactive intention prediction accuracy for mid-range threshold value for proactive guessing. ......................................................... 34

Figure 4.9  Intention prediction speeds for mid-range threshold value for proactive guessing. ......................................................... 34

Figure 4.10  Proactive intention prediction accuracy for mid-range threshold value for SVM-only model. ......................................................... 35

Figure 4.11  Intention prediction speeds for mid-range threshold value for SVM-only model. ......................................................... 35

Figure 4.12  Proactive intention prediction accuracy for high-range threshold value for SVM-only model. ......................................................... 37

Figure 4.13  Intention prediction speeds for mid-range threshold value for SVM-only model. ......................................................... 37
## LIST OF TABLES

| Table 2.1 | An example of future object anticipations given initial intention predictions. | 8 |
| Table 2.2 | An example of training data in which participants indicate which objects they use and in what order they use them for the *DrinkingLiquid* intention. | 9 |
| Table 2.3 | Codeword representation of each intention label created by ECOC. | 17 |
| Table 2.4 | Five-fold cross validation results for the SVM trained object-intention knowledge base. | 18 |
| Table 3.1 | A typical annotated video sample. | 20 |
| Table 3.2 | A walk-through of the second iteration of the proactive guess method for the object sequence [glass, sink, water]. | 24 |
| Table 4.1 | Classification data for initial threshold value for proactive guessing. | 30 |
| Table 4.2 | Classification data for initial threshold value for SVM-only model. | 31 |
| Table 4.3 | Classification data for low-range threshold value for proactive guessing. | 33 |
| Table 4.4 | Classification data for mid-range threshold value for proactive guessing. | 35 |
| Table 4.5 | Classification data for mid-range threshold value for SVM-only model. | 36 |
| Table 4.6 | Classification data for high-range threshold value for proactive guessing. | 36 |
| Table 4.7 | Classification data for high-range threshold value for SVM-only model. | 37 |
| Table 5.1 | A proposed list of 11 intentions that could be useful in future research. | 40 |
ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Xiaoli Zhang, for giving me the opportunity to work on this project. I am also very grateful for my colleague, Rui Liu, for his guidance through this project. Without the help and support of them both, the completion of this project would not have been possible. Completing this thesis is a great achievement for me, and I have learned a lot through the experience.

I would also very much like to thank Lori Sisneros, Shelly Myskiw, Matthew Garland, Dr. John Steele, Dr. Greg Jackson, Dr. Hao Zhang, Dr. Jenifer Blacklock, Dr. Derrick Rodriguez, and Dr. Ozkan Celik for all of the opportunities, council, and support that they have given me throughout my graduate career.

Finally, I want to thank my family, and especially my Gramma, for the love and support they’ve always given me.
CHAPTER 1
INTRODUCTION

This study begins with an introduction to the motivation, scope, and focus of this research. This introduction addresses related work that has inspired this research and introduces the novel ideas that will be presented.

1.1 Motivation and Focus

Humans have the ability to infer another person’s future intent. That is, based on observations of what a person is currently doing, humans are able to make proactive guesses to predict what their final intent might be. Making inferences about intent allows a human to assist another person in completing their task without the need for explicate instructions. This phenomenon is often referred to as attention sharing [1] [2] and is a foundation of human-to-human cooperation. The goal in this study is to model a human’s ability to proactively guess future intent by implementing a mechanism that integrates Reinforcement Learning (RL) and SVM classification. The presented proactive guess mechanism is intended to serve as an assistive robot’s simulated intuition. A robot endowed with this ability could make decisions based on few initial observations, make decisions quickly, and in turn, have the ability to offer assistance in a timely manner.

For a human to be able to guess another’s intention, they need to observe solid evidence as well as draw on experience to make inferences. This means that for a human to predict what kind of assistance another might need, it is necessary to intuitively relate the current observations with past experiences. Consider an example where, a human observes another person carrying a large number of grocery bags. The observer will first notice the large amount of bags they have. The observer intuitively knows that the person is probably carrying the groceries into their home and also knows that if they were to pick up some of the bags and bring them to the anticipated destination, it would probably be helpful. This ability to anticipate and make inferences is due to the fact that humans learn from experience. That is, the observer has most likely been in situations
where carrying items was involved. They have intuitively exploited the observations made during those situations to build an intuition based on experience. Recognizing that a person needs help with a task therefore comes from observation and making proactive and intuitive predictions about future intent.

The challenge of giving a robot the ability to proactively guess intent is two-fold. First, the robot must make useful observations, and second, the robot must be able to intuitively anticipate future actions. Currently, there have been great strides in research that focus on robot observation and give a robot the ability to recognize objects or activities. While the proactive guess mechanism presented in this study does rely on such a system, the novel contribution here focuses on a robot’s ability to make inferences about the future. That is, given solid evidence (observations), this study aims to simulate anticipated evidence (intuition) to guide intention predictions.

When a human completes a task, they often use a series of objects. The use of objects is a good indicator of intention. Thus, observed objects used by people is the main evidence used in this study. The proposed method both simulates experience learned from the exploitation of observations, and uses these experiences to make intention predictions. For example, when a person intends to drink water, they could use a cup, a sink, and water. This sequence of objects is related to the intent of drinking water. Similarly all objects in the sequence are related to each other, in that, when arranged in this specific order, drinking water can take place. Compiling a large data set of various object sequences related to various intentions is used in this study to train a robot’s intuition. Specifically, object-object relationships are exploited by RL, and object-intention relationships are exploited by an SVM model. The proactive guess method combines both object-object and object-intention relationships to simulate the intuition of an assistive robot.

Reinforcement learning has been chosen to train the object-object relation knowledge base because RL models the way humans learn from experience. Humans make choices, experience the outcome of those choices, and then evaluate whether or not was a good choice. Similarly the object-object knowledge base is trained by relating various objects, and then checking the actual relation between the objects based on human demonstrations. If the relation is represented in the
human demonstrations, the relation can be learned and the object-object relations can be used to help make intention predictions.

An SVM model has been the chosen to create the object-intention knowledge base because of its’ ability to perform well with complex data while still being relatively straight forward to implement. Feature vectors used by the SVM model in this study consist of object sequences that relate to intentions as well as the inter-relation between objects. From a large amount of object sequences and their labels the SVM model is able to learn patterns from the sequences, and map them to intention labels. Given an object sequence, a trained SVM model can predict an intent. Note that SVM is a method of binary classification and this study will focus on more that two intention labels. To extend SVM to more classes an Error Correcting Output Codes (ECOC) are used to extend the predictions of SVM to a multi-class model.

The novel contribution of the proactive guess method is its’ emphasis on the use of anticipated evidence to make intention predictions. This differs from most work in the robotics field which tends to focus on solid observations within the world. The main benefit of working with anticipated evidence is that it requires less observational input to make a prediction. Moreover, since the evidence is anticipated, predictions about intentions are made before the intent is complete. In general, the method uses an initial object observation to make an initial intention prediction using SVM classifiers. Next, given the prediction and the solid evidence, the object-object knowledge base is used to anticipate a future object. This produces a object sequence that includes a known object and and an anticipated object. The two objects are then treated as evidence and a final intention prediction is made from the SVM model.

To train both knowledge bases described above, data from 150 participants was collected from an Amazon Mechanical Turk [28] crowd sourced survey. In the survey, participants were asked what objects they usually use during five different activities. This data provides object sequences that relate to the 5 activity (intention) labels as well as the inter-relations of objects. To test the model, 20 human participants were asked to wear a pair of camera-glasses while they perform five different activities in their own home. The glasses are fitted with a camera and record a first-
person view of the participant’s activities. Since each video is of a first-person perspective, and since humans tend to look at objects they are using, used object sequences can easily be extracted from the videos. Each video is annotated based on the objects used in the video, and then used to evaluate the proactive guess model and compare it to a similar comparison model. The model performs to expectations, is able to make proactive guesses with a high rate of accuracy, and shows that proactive guess is beneficial to intention predictions in assistive robots.

1.2 Related Work

Robotics research has explored the idea of simulating human intuition by classifying human intentions (or activities) by observing various stimuli that relate to intentions. Contextual and environmental queues were used in [3][4][5] to classify a human's activity in addition to some use of object-intention information. Eye movement or gaze has also been widely used in intention prediction and classification [6][7][8][9][10]. The idea is that what a person is looking at can reveal their intent. This idea can be applied to gaze associated with objects or in association with a person’s attention. Object interaction and use[3][11][12] is also a popular approach and is the approach that is focused on in this study. Note that in many of the studies just mentioned, the ability of data acquisition (solid evidence) is usually a main focus, and intention classification and prediction accuracy relies on the accuracy of object, contextual, and gaze detection. Since much work has been done in the acquisition step, this study focuses on a robot’s ability to make proactive guesses about intentions assuming an accurate acquisition model.

Intention prediction models often use machine learning, or probabilistic models to be used as a knowledge bases. In [13] a Hidden Markov Model (HMM) is used to predict the intent of computer users based on user traits, and on the activities they did on a computer or mobile device. The contribution of this study was to show how the knowledge of human attributes (age, gender, education level, ...) might help to improve intention prediction. The study claims that the tree-like structure of HMM lends it’s self well to tasks on a digital device as it allows the model to classify button presses based on what a display might be showing. In our proactive guess model, we have
decided to use a Reinforcement Learning model in order to simulate human learning and intuition because of the way RL exhibits the way humans use trial and error to learn.

The work of [14] studies a robot’s ability to infer the intent of a human. Their motivation, similar to the motivation that guides this study, is to show that the simulation of human cognition is beneficial to human-robot cooperation. In this study however, the research focuses more on a robot’s perception, and less on proactive predictions about the future.

For the proactive guess approach, instead of the robot being stimulated to make predictions based on human action, proactive guess uses the objects of interaction as stimulation. This makes for a simpler, yet, still proactive approach to intention prediction. Second the knowledge bases used in [14][15] are of arraignments of human behavior-action correlations and the goal is to infer the future human action. This is yet another difference that proactive guess offers since, intention-object and object-object knowledge bases focus on objects and how they relate to other objects and intentions. The goal is to anticipate future objects and have the anticipation guide the robot towards making an overall intention predictions. Third, the confidence levels defined in [14][15] are based on likelihoods of the human to perform an action. In proactive guessing, confidence levels are calculated based on the posterior probabilities of anticipated intention predictions.

In [16], key objects involved in a human intention are observed and an intention is predicted using the SVM algorithm. Although this method uses human-object interaction to predict human intention, robot confidence is not adopted. Proactive guessing bases confidence levels on anticipations, making it more focused on the use of anticipated evidence. Moreover, observed objects in [16] are used as solid evidence to make intention predictions while objects in our method are used as solid evidence and used to make predictions about anticipated objects that are likely to be used in the future. That is, proactive guess starts with solid observation but is guided by anticipation. Proactive guessing is similar to [16] in that both methods do use SVM to predict intent.

The results presented here rely heavily on a robot’s ability to recognize objects that are being used by a human. However, since this research is focused solely on a mechanism that endows a robot with proactive guess intuition, it is assumed that the robot has the ability to recognize used
objects. Such work in computer vision has been very successful in recent years. In [16] human gaze is evaluated to classify the most salient object and then assumes that this object is the one in use. Proactive guessing is seen as an add-on to such a system since, in [16] there is more focus on the object recognition technique than there is on the intention/activity prediction. Given that the objects that a human is using can be identified, proactive guessing seeks to both improve activity classification, and classify proactively. In [27], the assumption is made that the use of objects is a strong indicator for activity classification. As such, it seems reasonable that intention predictions could also be made based on object interaction.

In [17] a similar technique is used to predict speaker intent in phone conversations. Signals or parts of phone conversations are mapped to intentions through probabilities. The confidence levels are calculated using a Bayesian updating method where predictions are made by, given a series of signals, combining conditional probabilities to refine the intention prediction. The difference in our method is that, in terms of phone conversations, in addition to predicting intention, we also want to predict the next signal or object.
Chapter 2 describes the training processes that occurred to set up each knowledge base used in the proactive guess method. Section 2.1 gives a brief overview of how proactive guessing works so that the reader can relate the each training process step with the overall experiment. Section 2.2 describes the training data used in this study and how it was collected. Section 2.3 -2.4 describe the study’s specific usage of Reinforcement Learning and the Support Vector Machine. Note that the words task, activity, and intention are often used interchangeably and refer to an activity that a human performs while using various objects. The word action is strictly reserved to define an action as it is commonly used in Reinforcement Learning.

2.1 Proactive Guess - General Methodology

Proactive guessing aims to, not only classify intentions, but also predict them before they are complete. Again, it studies the objects used by a human during intent completion as evidence to inform classification and proactive guesses. As such, intentions that require the use of multiple objects for completion are used. As a robot observes a human interacting with these objects, it anticipates future objects, and allows the anticipation to guide it’s final intention prediction.

Given that a robot can correctly identify objects in use, the following high-level example shows the general methodology employed in the proactive guess approach. Suppose that, a robot observes a human user touching the handle of a cupboard. Before the user touches another object, the robot makes predictions.

First, given the use of cupboard the robot predicts the most likely, and the second most likely intentions (e.g. 1. set the table and 2. drink water). These predictions are based on simulated experience from the object-intention knowledge base. Given each intention prediction, the robot will then anticipate the next objects that might be used. These anticipated object predictions come from the object-object knowledge base. An example of this is shown in Table 2.1.
Table 2.1: An example of future object anticipations given initial intention predictions.

<table>
<thead>
<tr>
<th></th>
<th>input</th>
<th>future-object anticipation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>(cupboard, set the table)</td>
<td>plate</td>
</tr>
<tr>
<td>2nd</td>
<td>(cupboard, drink water)</td>
<td>sink</td>
</tr>
</tbody>
</table>

Next, the robot makes new intention predictions based on both the known used objects, and the anticipated objects using the object-intention knowledge base. The posterior probability is then calculated for each intention prediction. If the difference between these two probabilities is greater than some threshold an intention prediction will be made. The difference between the two posterior probabilities is referred to as the robot’s confidence level in its’ guess. If the robot’s confidence is lower than the set threshold, an intention prediction will not yet be made and the robot will wait to observe the next object used by the human.

2.2 Training Data

Samples of the data collected from an Amazon Mechanical Turk crowd sourced survey were used to train each knowledge base. Each sample comes from a real human subject who reported the objects that they use to complete each task (intention). The five intentions used in this study are: DrinkingLiquid, WashingDinnerware, ReheatingFood, WashingHands, and TakingMedicine. This set of intentions will be denoted by $I$. For each intention a list of objects are made available for each participant to use. In the DrinkingLiquid intention, for example, the available objects are: bottle, coffee, juice, soda, water, cup, glass, mug, saucer, spoon, stirrer, straw, tap. This set of intention-specific objects is denoted by $O_{DL}$. Each of these intention-specific object sets $O_i$ is a subset of the set of all objects available for each intention. There are 40 in all, and this superset is denoted by $O$. After the participant fills out the normal objects they use to complete particular tasks, they also filled out a form where they provide alternative objects for completing the task, all of which must be elements of $O_i$. Data from a single participant for DrinkingLiquid follows the format in Table 2.2.
Table 2.2: An example of training data in which participants indicate which objects they use and in what order they use them for the *DrinkingLiquid* intention.

<table>
<thead>
<tr>
<th>( O_{DL} )</th>
<th>bottle</th>
<th>coffee</th>
<th>juice</th>
<th>soda</th>
<th>water</th>
<th>cup</th>
<th>glass</th>
<th>mug</th>
<th>saucer</th>
<th>spoon</th>
<th>stirrer</th>
<th>straw</th>
<th>tap</th>
</tr>
</thead>
<tbody>
<tr>
<td>seg1:</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>seg2:</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>seg3:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The value zero in the table above indicates that, for the particular intention category, the object named above was not used. Numbers greater than zero are used to indicate the order in which the objects are used. For example, *coffee* in the first row has the value one under it which indicates that it was used first. We also notice that both *glass* and *mug* have the value two beneath them, indicating that either object could be used as the second object in the sequence.

### 2.3 Object-Object Learning With RL

The knowledge base that supports anticipations is known as the object-object knowledge base. To create this knowledge base a Reinforcement Learning model where is used, given an intention \( I_i \) the model learns the relationships between all objects \( o \in O \). An RL method called Q-Learning is used here and, in general, begins by assigning a starting state \( s_{\text{start}} \) and a goal state \( s_{\text{goal}} \). Specifically \( s_{\text{start}} \) is a random \( o_i \in O_i \) and \( s_{\text{goal}} \) is a random \( o_n \in O_i \) where \( o_i \neq o_n \). From here the agent takes an action \( a_t \). An action is defined as moving from one state to another. More specifically, each action \( a_t \), moves to a new state/object \( o_i = s_{t+1} \). The new state is the result of one of two choices: a random action, or an action based on what the agent has learned through previous iterations of the learning process. This choice is the driver of Q-Learning and uses what is known as an \( \epsilon \)-greedy [18] value that defines how often the agent makes a random choice. In this study, \( \epsilon = 0.2 \) which indicates that a random action will be taken 20% of the time. Note that it has also been defined in training that the agent cannot perform an action that results the transition from a state to itself.

Each time the agent takes an action, it gets a rewards, \( r \), for making the action. This reward is based on each sequence of used objects from each participant. Given a reward for taking an
action a matrix called the Q-matrix is updated. This matrix ultimately stores the object-object relationships and serves as the object-object knowledge base. This is a $40 \times 40$ matrix with each $o \in O$ along both the rows and columns. Noting that an action is the transition between two objects, the relationship between the two is represented in the row of $s_t$ and the column of $s_{t+1}$.

Rewards for actions are defined in the following way:

- If $s_t \rightarrow s_{t+1}$ are sequential objects used by the participant, $r = -0.1$
- If $s_t \rightarrow s_{t+1}$ are not sequential, but were both used by the participant in any order, $r = -5$
- If either $s_t$ or $s_{t+1}$ are not objects used by the participant, $r = -10$

Once a reward is calculated, the Q-matrix is updated using the following equation:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r_{t+1} + \gamma (\arg\max_a Q(s_{t+1}, a)) - Q(s_t, a_t)]$$ (2.1)

The value $\gamma$, known as the discount factor where $\gamma \in (0, 1)$. Values closer to 1 put more weight on longterm goals and smaller values will weight more immediate goals. In this study $\gamma$ has been set to 0.9 to simulate longterm experience. In addition, lower values of $\gamma$ tend to yield faster learning processes but may find less than optimal solutions. The value $\alpha$ is known as the learning rate where $\alpha \in (0, 1)$. Larger values of this variable place more importance on learning new information and lower values place less weight on newer information. Again the simulation of experiential knowledge is important in this study and so $\alpha$ is set relatively high at 0.75 [19][20][21][22].

After any number of actions, if $o_i = o_n$ then the agent will stop and choose two new start and end states and begin a new iteration. For each intention in the set $I$, the training process performs 1,000 iterations (where a new iteration happens anytime two new starting and ending objects are chosen). Figure 2.1 shows a heat-map of the Q-matrix trained for the DrinkLiquid intention. Along the rows of the figure, the name of each object can be seen. These objects represent the first object in an object-object transition. Along each row there is a value given in each column. The column with the highest value corresponds to the second object in an object-object transition. In Figure
2.2 the same heat-map is shown but with the first and second highest row-column pair highlighted. Notice that highest object-object pair is given the value ten, and the second highest pair is given the value five. This is to make the colors in the heat map more visible.

Traditionally in Reinforcement Learning, all actions get the same reward (usually -1). However since the collected data is arranged in such a way that only a subset of \( O \) are available for each intention, it is possible exploit this to lead to quicker and less costly learned policies. To understand the effect of adding various biases for certain actions, experiments with the following simple RL example were performed. Suppose in this simple example that the agent’s goal is to find the optimal path from the green state to the red state shown in Figure 2.3.

![Figure 2.1](image.png)

Figure 2.1: A simple example for reward system testing. The goal of the agent is to find the path from the green node to the red node (left). The optimal path is shown in yellow (right).

Each State has 8 movement actions as seen in Figure 2.4. Again, each reward for all actions in RL is usually the same. In this example training starts with a reward of -1 for all actions. Then, the result of setting biases that favor taking a diagonal actions over horizontal or vertical actions is shown. For each experiment, the reward for taking a diagonal action is increased by 0.1. See Figure 2.4.

We find that increasing the bias of the diagonal action allows the agent to converge faster and converges with a lower overall cost. We see that with a bias of 1 (reward = 0) the agent converges in about 80 iterations and has a cost of around -7. In the case where all actions are given the same
Figure 2.2: A heatmap representing the Q-matrix for the *DrinkingLiquid* intention. Magenta colored nodes indicate object-object pairs that are highly related.
Figure 2.3: The same heatmap of the *DrinkingLiquid* Q-matrix shown above, but with the top two object-object relationships highlighted in each row. The value ten is given to the highest in each row, and the value five is given to the second highest relationship.
Next, knowing that a bias on the diagonal actions reduces cost and learning time, other reward systems were tested to understand the effect. Each new reward system is compared with the “No bias” or traditional system. The first reward system is similar to the diagonal bias systems tested above and each diagonal action is given a reward of -0.5. Next is a “don’t go” system in which moving North, East, or South, results in less rewards. In this case the agent should learn that taking any of these actions results in worse rewards. Next is a Northwest biased system in which taking a Northwest action yields better rewards. It is expected that this system should allow the agent to get to the end goal quicker. Finally is the NW-N-W reward system in which moving in the general direction of the end node brings better rewards. All four biased reward systems are shown in Figure 2.6. The results from these five reward systems are shown in Figure 2.7.

Because of the prior information that the goal state was located Northwest of the starting state, it’s not surprising to find that the NW bias systems allows the agent to converge the quickest and with the lowest cost. Also notice that this reward system shares the same final cost value as the NW-N-W bias system but, was able to converge much faster. We also notice that the “don’t go” reward system kept the agent from converging as fast as the others, and has the highest overall cost except for the traditional non-bias reward system.

From these results it is concluded that altering rewards for specific actions can improve both
the cost of learning, and the time needed for training. With this knowledge it was decided that this reward-bias approach could be beneficial to the training of the object-object knowledge base. As mentioned in Section 2.2. during the training process, rewards are based on how closely related an object is to the specific intention it is training for.

Figure 2.5: A graph showing the cost and duration each reward system needed to converge to an optimal path. As expected the reward system with the largest bias converged the quickest and with the least cost.

2.4 Object-Intention Learning With SVM

For the creation of the object-intention knowledge base an SVM model with a linear kernel function was adopted and was trained on the same training data used for the object-object knowledge base. SVM was the chosen model of use because it tends to work well on complex sets of data while still being straightforward to implement [23].

SVM, however, is a model for binary classification, and data needs to be classified into five intention labels, the `fitcecoc` function from MATLAB’s Statistics and Machine Learning Toolbox was used [24]. This function uses Error-Correcting Output Codes (ECOC), and is able to turn
Figure 2.6: A representation of four additional tested reward systems.

Figure 2.7: Results from the four tested reward systems. It was found that a bias that tends towards the end state allows the agent to converge the fastest and with the least cost.
a multi-classification problem into a series of binary classification problems for SVM to use [25]. ECOC does this by creating a codeword for each intention label as shown in each row of Table 2.3. In the table, each column represents a single binary learner [26]. The column $L_0$ shows that the first binary classifier only trains on training data from the DrinkLiquid and ReheatFood intentions. Similarly, the second binary learner $L_1$ uses SVM to train on DrinkLiquid and TakingMedicine and so on until the $10^{th}$ learner.

When this model is used to classify testing data, each binary classifier will classify the data as one of the two intention labels and assign it a 1 or $-1$ according to each learner. After all 10 binary classifiers are finished, the new sequence of data will be represented as its’ own codeword. This new codeword is compared to each of the intention label codewords and the “closest” one is selected as the predicted intention. The term “closest” in this sense refers the Hamming distance between two codewords.

MATLAB’s fitSVMPosterior function also allows for the calculation of posterior probabilities for each testing sample [30]. This ability is used in the confidence level calculation step of the proactive guess method described in Section 3.2. During training, a sigmoid function using 10-fold cross validation is used to map probabilities to scores from the SVM. When a new testing sample is classified, posterior probabilities for each intention label can be output. In the proactive guess method, only the top two posterior probabilities are ever used.

Table 2.3: Codeword representation of each intention label created by ECOC.

<table>
<thead>
<tr>
<th>Binary Learner</th>
<th>$L_0$</th>
<th>$L_1$</th>
<th>$L_2$</th>
<th>$L_3$</th>
<th>$L_4$</th>
<th>$L_5$</th>
<th>$L_6$</th>
<th>$L_7$</th>
<th>$L_8$</th>
<th>$L_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrinkLiquid</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ReheatingFood</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TakingMedicine</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>WashingDinnerware</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>WashingHands</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

For each training sample a $1 \times 1561$ feature vector is built and fed into the multi-class SVM model along with it’s corresponding intention label. The feature vector takes into account both the objects used and transitions between objects. In each feature vector there are 40 columns that
represent each object and 1521 columns for each possible transition (excluding object transitions to itself). We note that this feature vector is quite long, and adding more objects to training, increasing the size of the vector exponentially.

For each $I_i$, 150 training samples were used. A 5-fold cross validation test was used to get a general idea of how well the model might be able to predict intentions. The cross validation results are shown in Table 2.4.

Table 2.4: Five-fold cross validation results for the SVM trained object-intention knowledge base.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrinkingLiquid</td>
<td>0.8556</td>
<td>0.9734</td>
</tr>
<tr>
<td>WashingDinnerware</td>
<td>1.0</td>
<td>0.9398</td>
</tr>
<tr>
<td>ReheatingFood</td>
<td>1.0</td>
<td>0.98</td>
</tr>
<tr>
<td>WashingHands</td>
<td>1.0</td>
<td>0.9802</td>
</tr>
<tr>
<td>TakingMedicine</td>
<td>0.9728</td>
<td>0.9332</td>
</tr>
</tbody>
</table>

Noting that almost all accuracy and recall values are above 90% the results from the multi-class SVM model are very encouraging. It is important to note that the precision for ReheatingFood, WashingHands, and TakingMedicine all have 100% precision scores. This may indicate that the training data for our experiment may not be diverse enough. This is discussed further in Chapter 4.
CHAPTER 3
PROACTIVE GUESS

In Chapter 3 the methodology used for collecting real-world testing data for the proactive guess model is described in Section 3.1. Minor data collection errors are also described here. A detailed description of the architecture of the proactive guess method is presented in Section 3.2.

3.1 Testing Data Collection

The proactive guess method, from an implementation point of view, does heavily rely on a robot’s ability to correctly recognize objects that are being used. Thus, a real-time object recognition computer vision engine is necessary for any real world application. However, the main focus of this research is to both set up a framework for the proactive guess method and to test its’ usefulness. To stay within the constraints of an actual object-recognition system testing data is collected in such a way that might simulate the way a computer vision system would.

Since computer vision pulls data from cameras, testing data here was also collected from video. Another goal was to collect data that would represent real-world data. The best solution was to capture video from people acting out intentions in their own homes. To do this, a pair of camera-glasses was used. The camera-glasses are simply a pair of glasses with a camera embedded in the frame. The camera can record and store video on a micro SD card. The camera on the glasses located right between the eyes of the user. It was important that the camera be located in the middle of the face as it would give us the best perspective to capture participants using objects. Twenty participants were asked to take the glasses home, and record first person videos of themselves completing each of the five intentions. Due to minor human error we collected 97 videos to use for testing.

Each video was watched, and classified by the researcher. To simulate used object recognition, the researcher recorded a sequential list of all the objects that were manipulated in each video. The annotated video data ranged from one object to nine object sequences. Each of these sequences
are later converted into the same feature vector format explained in Section 2.4 and are used in the proactive guess mechanism. Figure 3.1 shows screen shots of the video that this annotation came from and Table 3.1 shows the annotation made from this video.

Table 3.1: A typical annotated video sample.

| Steve | DrinkingLiquid | glass | sink | water |

Figure 3.1: Screenshots from collected training video.

There are a few noteworthy problems that occurred during data collection. First, it was difficult for the participant to make sure that the object being manipulated was always in the field of view of the camera. All intentions were generally performed in the kitchen and participants manipulated objects on counters below the chest. The camera in the glasses seemed to be set in such a way that they pointed up possibly due to the angle the nose imposes on the glasses. Another possibility was that the camera itself was set in a position for capturing video that is level with the users face. In any case, if the participant did not tilt their head downward in an unnatural way, some objects were not captured. In addition to this, human participants completed everyday activities that were very normal and easy. Because of this, participants did not have to focus much on the task at hand and, as a result, forgot to focus their attention on the video capture process. Figure 3.2 shows a participant reaching for soap with his left hand but does not capture the action in the shot. In many of these cases the researcher had to make inferences as to which object was being manipulated. These inferences were usually based on audio cues and human intuition.

Another issue came up when participants used objects that were not in the set of available objects, \( O \), used in training. In Figure 3.3 we see that the participant is manipulating a plastic container, however, plastic container was not an object included in the set of available objects.
These types of issues were usually resolved by finding an object in $O$ that was closely related to the manipulated object and recording it as such. For this example, plastic container was changed to bowl. Similarly fork was changed to spoon. We note that this may add to errors within testing, and it would be beneficial to increase the size of $O$ to include a wider range of objects. A proposed list of objects for future research is included in Chapter 5. Along these same lines, there were some objects that were used often, but the objects were not an element of $O$ and there were no closely related object in $O$. Figure 3.4 shows the screenshot of a participant using a cupboard, which, was not an item contained in $O$. In these cases, the object was not recorded into the testing data set. These are some of the most notable objects not included in $O$: refrigerator, water dispenser, fork, stove, freezer, icecube tray, ice, pitcher, cupboard, cabinet, pot, and medicine bottle.

3.2 Proactive Guess - Combining RL with SVM

To explain how the Proactive Guess method is works to make proactive predictions in detail, a step-by-step walkthrough is presented using the sequence shown in Figure 2.3.
Figure 3.3: The participant uses a plastic container, however, this object is not an element in the list of available objects.

Figure 3.4: A cupboard is used, however, cupboard is not an object in the set of available objects and is not closely related to any.

First, for each sequence, the objects from the sequence are put into into a single vector called $O_{\text{used}}$. For this example, $O_{\text{used}} = [\text{glass, sink, water}]$. To simulate a human using objects over a period of time and to simulate the order in which they use the objects, we iteratively make intention
predictions on [glass], then on [glass, sink] and finally on [glass, sink, water].

For the first iteration, the sequence [glass] is converted to the feature vector format explained in Section 2.4. The vector is then classified by the SVM object-intention model. Along with the most likely classification, the second most likely intention classification is also output from the model. This first step is represented below.

\[\text{[glass]} \rightarrow \text{SVM} \rightarrow \text{Most Likely Intention} = \text{DrinkingLiquid} \]
\[\text{[glass]} \rightarrow \text{SVM} \rightarrow \text{Second Most Likely Intention} = \text{TakingMedicine}\]

Next, the proactive guess portion of the method is examined. Given the two results above, and the known object glass, two proactive guesses are made. The DringLiquid Q-matrix is used as a look-up table to anticipate the next object. We do the same with the TakingMedicine Q-matrix. Each new anticipated object is appended to sequence of known objects. For example, given glass and DringLiquid, we assume that the object sink is anticipated as the next object. Similarly, given straw and TakingMedicine assume that sink is anticipated as the next object. Now there are two anticipated sequences \(A_1\) and \(A_2\). Each contains an object that is known to have been used, and an anticipated future object.

\[A_1 = \text{[glass, straw]} \quad A_2 = \text{[glass, sink]}\]

Each of these sequences are converted into feature vectors and then are classified by the multi-class SVM model. The posterior probability of each classification is also produced.

\[A_1 \rightarrow \text{SVM} \rightarrow \text{DrinkingLiquid}, 0.731\]
\[A_2 \rightarrow \text{SVM} \rightarrow \text{TakingMedicine}, 0.652\]

At this point, if the difference between the two posterior probabilities is higher that a predefined threshold, the intention with the highest probability is output as the final intention. If a final intention prediction is made, no other iterations will happen on this particular object sequence sample. If this difference is less than the threshold, however, an intention prediction would not be
made and another iteration would begin on the object sequence sample. This difference between the posterior probabilities of both anticipated intention predictions is referred to as the *anticipation confidence value* or simply, *confidence value*. In the case that there is a second iteration, the next actual object in $O_{used}$ is added, so that more proactive guesses can be made. A second iteration of this process is represented in Table 3.2. In general, each iteration goes through these steps. Figure 3.5 gives a flow chart representation of the proactive guess mechanism process in general.

Table 3.2: A walk-through of the second iteration of the proactive guess method for the object sequence [glass, sink, water].

**Initial Intention Predictions from Object-Intention KB**

- $[\text{glass, sink}] \rightarrow \text{SVM} \rightarrow 1^{st} \text{ Place Intention} = \text{DrinkingLiquid}$
- $[\text{glass, sink}] \rightarrow \text{SVM} \rightarrow 2^{nd} \text{ Place Intention} = \text{TakingMedicine}$

**Next Object Anticipation from Object-Object KB**

- $(\text{sink, DrinkingLiquid}) \rightarrow \text{DrinkingLiquid Q-matrix} \rightarrow [\text{water}]$
- $(\text{sink, TakingMedicine}) \rightarrow \text{TakingMedicine Q-matrix} \rightarrow [\text{medicine}]$

- $A_1 = [\text{glass, sink, water}]$
- $A_2 = [\text{glass, sink, medicine}]$

**Anticipation Intention Predictions from Object-Intention KB**

- $A_1 \rightarrow \text{SVM} \rightarrow \text{DrinkingLiquid}, 0.918$
- $A_2 \rightarrow \text{SVM} \rightarrow \text{TakingMedicine}, 0.324$

**Confidence Value Calculation**

- Anticipation confidence value $= |0.918 - 0.324| = 0.594$
Figure 3.5: A flowchart representation of the implemented proactive guess mechanism.
Chapter 4 presents the final results and conclusions that have been learned from this study. In Section 4.1 an SVM-only model is introduced and will be used to compare with the proactive guess model. In Section 4.2 gives a detailed description of how results are organized. Actual testing results are shown in Section 4.3.

4.1 Experiment Set-Up

To examine the usefulness of the proactive guess approach compared to another model, each of the 97 testing samples were tested with a simpler intention prediction model. The motivation for this is to test whether or not proactive guessing benefits proactive intention prediction. In this comparison model, referred to as the SVM-only model, proactive guess was not used. That is, for each iteration through an object sequence, the model makes intention predictions based only on the initial intention prediction with in our method. This means that the comparison model only uses SVM once and does not make future object anticipations, or intention predictions based on anticipated objects. A flowchart showing the architecture of the SVM-only model is shown in Figure 4.1.

The SVM-only model outputs the top two intention predictions and their posterior probabilities. As such, the confidence level value in this case is calculated based from the posterior probabilities of the initial SVM intention prediction. Note that, in the proactive guess model, the confidence level is calculated from the posterior probabilities from two separate intention predictions. However, in SVM-only case, both posterior probabilities are results from the same SVM classification. It is often the case that the posterior probability of the top intention prediction is much higher than the the probability of the second place prediction. This inevitably leads to confidence value calculations that are much larger than confidence values calculated in the proactive guess model. Because of this, the SVM-only model tends to make proactive intention predictions
very early and hyper-confidently. In fact, most intention predictions are made after the first object is observed. The architecture of both models is the same except the SVM-only model bypasses the proactive guess step. This means that it does not use object anticipation to guide intention prediction.

Figure 4.1: A flowchart representation of the SVM-only comparison model.
4.2 Result Representation

The main interest in using the proactive guess approach is to see how accurate the proactive guesses are. To promote many proactive classifications to study, we intentionally set the confidence threshold low at 0.05. Again, this means that if the difference between the posterior probabilities of two anticipated intention predictions is greater than the threshold, a final intention prediction is made.

In the first figure below, Figure 4.2, the accuracy for each proactive intention classification is shown. Correctly classified object sequences are shown in blue, while incorrect classifications are shown in red. If a classification was made only after an entire object sequence was parsed, the classification is not designated as a proactive prediction and, it is not represented in the figure.

In Figure 4.3, the speed at which intention predictions were made are shown. Speed in this sense refers to the amount of the object sequence that was observed before an intention prediction was made. For example, in the object sequence [sink, water, dish, soap, sponge], if a final intention prediction is made when the third object ‘dish’ is parsed, the speed of the prediction is represented as 3/5 or 0.6. This is an important statistic to present as it give some insight into just how proactive the predictions are. This normalization can be slightly misleading due to the fact that a two-object sequence and a four-object sequence could both have a speed of 1/2. Nevertheless, this normalization of speed does give insight into understanding generally how much of an object sequence is usually parsed before a prediction is made.

To combat loss of information from normalization of speed data, data that focuses on the lengths of all object sequences is displayed. In the Table 4.1, the first row represents sequence lengths from one to ten. The second row shows a count of the number of sequences at each sequence length. The third row shows a count of the number of predictions/classifications (not necessarily proactive) made for at each sequence length. Finally, the fourth row shows the number of correct classifications made at each sequence length. In the table, the first column represents object sequences of length 1. Looking at the second row (and first column) the value 10 means
that of all testing data, there are 10 object sequences of length 1. The value in the third row, means that there were 74 classifications made after only a single object from a sequence was parsed. The fourth row shows that of these 74 classifications, 63 were correct.

4.3 Initial Results

In the proactive guess model with a confidence threshold of 0.05 (shown below), 87.6% of the classifications were correct. 74.2% of all classifications were proactive predictions and of those, 83.3% were correct. All other classifications, non-proactive were classified 100% correctly. The accuracy of proactive intention predictions for each intention are shown in Figure 4.2. Accurate intention predictions are shown in blue, and incorrect predictions are shown in red. The speeds of intention predictions are shown in Figure 4.3. Data from all classifications (proactive and non-proactive) are shown in Table 4.1.

In the SVM-only model 87.6% of the classifications were also correct. However, 89.7% of all classifications were anticipated predictions and of those 86.2% were correct. All other classifications, only 10 were classified 100% correctly. See Figures 4.4 and 4.5 for accuracy and prediction speed data in the same format as above. A table showing data for all classifications is shown in Table 4.2.

Figure 4.2: Proactive intention prediction accuracy for initial threshold for proactive guessing.
Figure 4.3: Intention prediction speeds for initial threshold value for proactive guessing.

Table 4.1: Classification data for initial threshold value for proactive guessing.

<table>
<thead>
<tr>
<th>object seq length</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>no. of seq per seq length</td>
<td>10</td>
<td>18</td>
<td>30</td>
<td>20</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>no. of classifications</td>
<td>74</td>
<td>18</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>no. of correct classifications</td>
<td>63</td>
<td>18</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 4.4: Proactive intention prediction accuracy for initial threshold value for SVM-only model.
Comparing the two, the proactive guess model made 17.2% fewer proactive intention predictions than the model that only used SVM. However, the intention predictions of the proactive guess model were only 3.36% less accurate than the SVM-only model. This data seems to point to the conclusion that the proactive guess approach does not encourage proactive intention predictions as well as the simpler model does. It is also reasonable to conclude that the proactive guess approach is more careful about making proactive predictions. Of course, this can be attributed to the higher confidence levels produced in the SVM-only model. Note that, the SVM-only model made all of its’ intention predictions after observing only the first object. This is disconcerting since, in the real world it is not reasonable to always be able to make an intention prediction after only observing one object. It could be that the SVM-only model is hyper-confident, while the proactive guess approach doesn’t make a prediction until it has a high confidence that its’ prediction will be correct. Nevertheless, it is encouraging that the proactive guess model performed in this test as expected.
That is, it was able to effectively use object anticipations to guide its’ intention predictions.

4.3.1 Low-Range Threshold Results

Next we wanted to see the result of using an even lower threshold and make comparisons with the results above. Here we use a threshold of 0.01. Such a small threshold forces proactive guesses to happen when the first object from a sequence is parsed.

For the proactive guess model, we do see an increase in the amount of proactive anticipations made. In this case 89.7% of all classifications were proactive and the accuracy of these predictions increased to 86.2%. The overall accuracy for all classifications stayed the same at 87.6%. What is interesting is that the results from this test are exactly the same as results from the SVM-only model. Notice in Table 4.3 that the proactive guess model was forced to make all of its’ intention predictions after parsing the first object and so it was reduced to function the same as the hyper-confident SVM model. The SVM-only model with the 0.01 threshold performed the same as it did with a threshold of 0.05 and the results are not shown since they are the same as in Figures 4.4 and 4.5. Data corresponding to the proactive guess model are shown in Figures 4.6 and 4.7.

Figure 4.6: Proactive intention prediction accuracy for low-range threshold value for proactive guessing.
4.3.2 Mid-range Threshold Results

Next we test both models using a mid-range threshold value of 0.35. This experiment shows that as the threshold is increased, the proactive guess model is more conservative about making proactive intention predictions. That is the proactive guess model will have to have a high level of confidence before making a prediction. For both the proactive guess and SVM-only models we get an overall accuracy of 88.6%. As expected the amount of proactive guesses decreased to 53.6%. Of these proactive guesses 78.8% were accurate. The other 45 of it’s classification were not proactive but achieved 100% accuracy.

For the SVM-only model, 89.7% were proactive and had an accuracy of 87.4% and its’ 10 non-proactive classifications were 100% correct. At this threshold level, the SVM-only model made 40.2% more proactive intention predictions than the proactive guess model. The SVM-only model also made 9.8% more correct proactive intention predictions. It is interesting to note that
with an increased threshold, the proactive guess model seems to also make less accurate proactive intention predictions. Graphical data for the proactive guess model are shown in Figures 4.8, 4.9, and Table 4.4 below. Data from the SVM-only model are shown in Figures 4.10, 4.11, and Table 4.5.

Figure 4.8: Proactive intention prediction accuracy for mid-range threshold value for proactive guessing.

Figure 4.9: Intention prediction speeds for mid-range threshold value for proactive guessing.
Table 4.4: Classification data for mid-range threshold value for proactive guessing.

<table>
<thead>
<tr>
<th>object seq length:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>no. of seq per seq length:</td>
<td>10</td>
<td>18</td>
<td>30</td>
<td>20</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>no. of classifications:</td>
<td>57</td>
<td>17</td>
<td>16</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>no. of correct classifications:</td>
<td>48</td>
<td>17</td>
<td>14</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 4.10: Proactive intention prediction accuracy for mid-range threshold value for SVM-only model.

Figure 4.11: Intention prediction speeds for mid-range threshold value for SVM-only model.
Table 4.5: Classification data for mid-range threshold value for SVM-only model.

<table>
<thead>
<tr>
<th>object seq length:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>no. of seq per seq length:</td>
<td>10</td>
<td>18</td>
<td>30</td>
<td>20</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>no. of classifications:</td>
<td>82</td>
<td>14</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>no. of correct classifications:</td>
<td>73</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

4.3.3 High-range Threshold Results

Finally, a high-range threshold at 0.65 is tested. In this experiment, the proactive guess model made no proactive predictions. That is, for each testing sequence, it waited until it had parsed all the objects in the sequence before making an intention prediction. Note that, again, its 97 non-proactive classifications, 100% were accurately classified. Interestingly, 77.3% of the SVM-only model’s classifications were proactive and had an accuracy of 93.3%. It had 22 non-proactive classifications with an accuracy of 100%. The model’s overall accuracy was 94.8%. This seems to show that when the SVM-only model is less confident, it actually makes more accurate predictions. Data corresponding to the SVM-only model are shown in Figures 4.12, 4.13, and Table 4.7. Data for the proactive guess model is shown in Table 4.6.

Table 4.6: Classification data for high-range threshold value for proactive guessing.

<table>
<thead>
<tr>
<th>object seq length:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>no. of seq per seq length:</td>
<td>10</td>
<td>18</td>
<td>30</td>
<td>20</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>no. of classifications:</td>
<td>10</td>
<td>18</td>
<td>30</td>
<td>20</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>no. of correct classifications:</td>
<td>10</td>
<td>18</td>
<td>30</td>
<td>20</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

4.5 Conclusion

The data presented above shows that the proactive guess method is effective at making proactive intention predictions. Depending on the threshold used, the method can make accurate proactive intention predictions with accuracies in the range of around 80%. It was also shown that the proactive guess method performed similar to the SVM-only model for certain threshold values. Noticing that the SVM-only model tends to make intention predictions after only parsing a single
Figure 4.12: Proactive intention prediction accuracy for high-range threshold value for SVM-only model.

![Bar chart showing correct and incorrect intent anticipations for different objects.]

Figure 4.13: Intention prediction speeds for mid-range threshold value for SVM-only model.

![Bar chart showing anticipated guess speed distribution for different thresholds.]

Table 4.7: Classification data for high-range threshold value for SVM-only model.

<table>
<thead>
<tr>
<th>object seq length:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>no. of seq per seq length:</td>
<td>10</td>
<td>18</td>
<td>30</td>
<td>20</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>no. of classifications:</td>
<td>53</td>
<td>14</td>
<td>25</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>no. of correct classifications:</td>
<td>50</td>
<td>13</td>
<td>25</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
object, it is encouraging to see that the proactive guess model takes a more natural and conservative approach to making guesses. That is, similar to human cognition, it has shown that it does use object anticipations to guide its’ proactive guessing. Note also that when humans make guesses about the future, they aren’t always correct, this too is represented by the proactive guess model. The SVM-only model showed that, while it did out perform the proactive guess model, it was very hyper-confident which may not be ideal in real-world situations where more data is at play.

Even though the SVM model was hyper-confident it still always made very accurate predictions. This may mean that the data used to train and test these models might be too unique. To further support this speculation, also consider that in each of the tests above, all non-proactive intention predictions were always 100% correct. Remember that a non-proactive guess does not use the object-object knowledge base and only uses the SVM object-intention knowledge base. In Section 2.4 it was mentioned that the cross-validation for three of the intention categories had accuracies of 100%. This could mean that the model is either overfitted, or contains relationships that are too unique. In the future it would be very interesting to expand the training and testing data to include more intention categories, and more objects. Ideas for future work are described in Chapter 5.
CHAPTER 5
FUTURE WORK

With a fully functioning proactive guessing mechanism, future research on this approach should be devoted to data collection and expansion. In this section modifications to data collection approach are explored.

5.1 Object Inclusion

The current training data comes from participants taking a survey regarding five separate tasks and had 40 objects available for them to use. However, for each separate tasks, only a subset of the 40 items were available for use. For data collection in future work, it may be beneficial to the study if participants have the option to use all items for every task. This both promotes a more real-world setup, and will eliminate any unintentional biases in the training data. However, note that our RL knowledge base did make use of knowing which objects were part of each subset, and gave different rewards based on this. Thus, if this proposed method of data collection is adopted in the future, the RL knowledge base will have to be built using a uniform reward system, or award biases based on another metric (Section 5.3).

5.2 Intention Prediction Expansion

To increase the challenge of making proactive intention predictions, expanding the number of intent labels is a natural approach. Eleven new intention categories are proposed here. Noting that the current five intentions are kitchen related, and there are a lot of objects in a kitchen, it makes sense to stay with the theme. In the following proposed list there are both new intention categories, as well as categories split from more specific categories Table 5.1. We can surmise that making intentions less general is likely to increase the cross over of objects used in different intentions - making prediction more challenging. The following list is in no particular order.
Table 5.1: A proposed list of 11 intentions that could be useful in future research.

<table>
<thead>
<tr>
<th>Action 1</th>
<th>Action 2</th>
<th>Action 3</th>
<th>Action 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrinkWater</td>
<td>DrinkCoffee</td>
<td>DrinkOtherLiquid</td>
<td>TakeMedicine</td>
</tr>
<tr>
<td>WashHands</td>
<td>WashDinnerwareByHand</td>
<td>WashDinnerwareWithMachine</td>
<td>CleanCounter</td>
</tr>
<tr>
<td>PrepareFrozenFood</td>
<td>ReheatFood</td>
<td>SetDinnerTable</td>
<td></td>
</tr>
</tbody>
</table>

5.3 Additional Objects

An expanded object list is proposed in this section. Larger objects that were often used by participants in this study have been added to this list as well as some small items. The following list groups items based on a larger object that usually contains smaller items. Note that the large items should also be included in the list. The proposed list is composed of 55 different objects.

- **Refrigerator**: water dispenser, pitcher, plastic container, soda, bottle, food, can, general container.
- **Freezer**: ice, ice cube tray, frozen food, plastic bag.
- **Cupboard**: plate, bowl, dish, cup, mug, glass, saucer, medicine.
- **Drawer**: fork, spoon, butter knife, general knife, straw, stir stick.
- **Oven**: stovetop, pot, pan, tea kettle, oven mit.
- **Sink**: water, sponge, rag, paper towel, general towel, dish soap, hand soap, dry rack.
- **Misc items**: napkin, coffee, coffee maker, tea, microwave, dishwasher, dishwasher detergent, sanitizer, aluminum foil.

This relationship between items that are contained in bigger ones could be used to add biases to the RL process. For example taking an action from the Freezer to the ice cube tray could be a greater reward than Freezer to sponge. This is a possible alternative method to introducing a bias into the reward systems of the RL process.

Considering that training data is increasingly difficult to collect as data sets get bigger, the crowd sourced survey is still recommended for future work. It is important to emphasize that for training data, the interest is in modeling knowledge bases based on human experience. Thus, a crowd sourced survey that participants can answer from experience is reasonable. Participants
would essentially be asked to, for each intention, indicate in an online form, which items are used and in what order they are used. Taking this route makes for a faster approach for collecting data and, without loss of generality, does model human experience. As for data collection for testing data, the same glasses-camera method used in this study is recommended.
In this paper, a method for proactive intention prediction for assistive robots has been presented. A detailed description of this method was presented in Chapter 2. Training data was used though a reinforcement learning model to create an object-object knowledge base. In addition to the common functionality of RL, we presented our use of a bias reward system that took advantage of pervious knowledge about objects. The same training data was used with a MATLAB ECOC supported SVM model to train the intention object knowledge base to high accuracy. The combination of these two knowledge bases built the foundation to the predictions made by the proactive guess method. In Chapter 3, a detailed walk-through of the proactive guess method was presented.

The testing data used with our method was gather from human participants who recorded first-person videos of themselves completing various tasks. Each video was annotated, and was tested with the proactive guess model and a comparison model based solely on a SVM model. Results from testing showed that the proactive guess model performed as expected and was able to make proactive intention predictions that were guided by object anticipations. The accuracy of the the proactive intention predictions is encouraging. It was also found that the comparison model, which only used SVM, often marginally out performed the proactive guess model. However, the hyper-confidence of SVM-only model may not make it the best model for real-world situations. It would be interesting in future work to test both these models again with more complex data. In Chapter 5, additional data for future work is proposed. It is the hope of the researchers that future work on this topic is now more accessible given that a working proactive guess model has been developed though this study.
REFERENCES


