2007-06

An analysis of learning algorithms in complex stochastic environments

Poor, Kristopher D.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

AN ANALYSIS OF LEARNING ALGORITHMS IN COMPLEX STOCHASTIC ENVIRONMENTS

by

Kristopher D. Poor

June 2007

Thesis Advisor:	Christian Darken
Second Reader:	Terry Norbraten

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2007	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE An Analysis of Learning Algorithms in Complex Stochastic Environments			**5. FUNDING NUMBERS**
6. AUTHOR(S) Kristopher D. Poor			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

As the military continues to expand its use of intelligent agents in a variety of operational aspects, event prediction and learning algorithms are becoming more and more important. In this paper, we conduct a detailed analysis of two such algorithms: Variable Order Markov and Look-Up Table models. Each model employs different parameters for prediction and this study attempts to determine which model is more accurate in its prediction and why. We find the models contrast in that the Variable Order Markov Model increases its average prediction probability, our primary performance measure, with increased maximum model order, while the Look-Up Table Model decreases average prediction probability with increased recency time threshold. In addition, statistical tests of results of each model indicate a consistency in each model's prediction capabilities, and most of the variation in the results could be explained by model parameters.

14. SUBJECT TERMS Event prediction, learning algorithms, agents, Markov models.			**15. NUMBER OF PAGES** 65
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

AN ANALYSIS OF LEARNING ALGORITHMS IN COMPLEX STOCHASTIC ENVIRONMENTS

Kristopher D. Poor
Ensign, United States Navy
B.S., Pennsylvania State University, 2006

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MODELING, VIRTUAL ENVIRONMENTS, AND SIMULATION (MOVES)

from the

NAVAL POSTGRADUATE SCHOOL
June 2007

Author: Kristopher D. Poor

Approved by: Christian Darken
Thesis Advisor

Terry Norbraten
Second Reader

Rudy Darken
Chair, MOVES Academic Committee

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

As the military continues to expand its use of intelligent agents in a variety of operational aspects, event prediction and learning algorithms are becoming more and more important. In this paper, we conduct a detailed analysis of two such algorithms: Variable Order Markov and Look-Up Table models. Each model employs different parameters for prediction, and this study attempts to determine which model is more accurate in its prediction and why. We find the models contrast in that the Variable Order Markov Model increases its average prediction probability, our primary performance measure, with increased maximum model order, while the Look-Up Table Model decreases average prediction probability with increased recency time threshold. In addition, statistical tests of results of each model indicate a consistency in each model's prediction capabilities, and most of the variation in the results could be explained by model parameters.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to start off by thanking Dr. Christian Darken for his guidance in this thesis process. Conducting an in-depth research study in less than one year is not an easy task and without Professor Darken keeping me on task there is no way I would have been able to complete it.

Next, I would like to thank all of my professors and classmates here at the MOVES Institute. I have learned so much from each and every one of you in more ways than one.

Finally, I would like to thank all of my friends and family. Thank you for the constant support in everything I have done. Thank you for believing in me.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Stuart Russell and Peter Norvig in their book, *Artificial Intelligence: A Modern Approach*, define an agent as “anything that can be viewed as perceiving its environment through sensors and acting upon that environment.” [1] This definition is quite broad, though, as there are numerous types of agents each with their own set of characteristics. Agents of all types have a wide range of applications, especially within the Department of Defense. The military uses agents on complex combat models and simulations, numerous operational systems, and simple internet-based intelligent training systems.

One specific type of agent is an autonomous, intelligent, utility-based agent. The characteristics of said agents are described in Nikola Kasabov’s book *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*. Because the agent is both intelligent and autonomous it takes independent actions rather than actions from fixed knowledge bases or pre-set rules and it will adapt to changes in its environment. This adaptation will take place in real-time as the agent learns the nuances of the environment in which it is operating. In addition to these characteristics, the agent must be able to analyze itself and its behavior, as its primary goal is to maximize its utility. In many cases, this “goodness” can correspond to the agent’s ability to learn its environment. [2]

In this work, we study the characteristics of autonomous, intelligent, utility-based agents which are important because it is the type of agent making predictions in this study. The agent attempts to predict events in the environment of a role-playing game. The environment consists of 19 different locations as well as different weapons and monsters. The protagonist of the game explores the environment, fights the monsters, and collects different weapons. Actions, events, and sensations that take place in the environment are then passed to the agent.

When the intelligent agent receives percepts corresponding to the actions, events, and sensations that have transpired in the environment, predictions are made. The agent attempts to predict the next percept in the sequence using one of two learning algorithms already coded into the model: Variable Order Markov and Look-Up Table. Predictions by the agent happen continuously as percepts are received and said predictions build upon each other. This is because the agent learns more about the environment as each percept is received. The actual learning process of the agent differs based on both the percept sequence it has received and which prediction algorithm is being implemented.

Each learning algorithm is effective in their prediction capabilities. Both models have parameters that can be manipulated to alter prediction capabilities. The Variable Order Markov Model can be utilized with different search depths and the Look-Up Table model can be manipulated with different relevant percept time thresholds. Altering these parameters will change the prediction probability distributions for each percept.

This research study will address relationships between percept sequences, different learning algorithms, their parameters, and prediction probabilities. The analysis of these relationships will attempt to find the best combination of agent parameters to maximize prediction probability and determine what factors have the greatest impact on how the agent makes its predictions.

B. PROBLEM STATEMENT

This thesis attempts to address how a particular agent learns in a complex stochastic environment. The primary question to be answered is how different parameters and prediction algorithms impact the agent's ability to make predictions. Along these lines, what parameters have the biggest impact on prediction of different percept types? Are there ways to improve the algorithm in order to advance the agent's learning process? These are the questions this thesis will attempt to answer.

C. TECHNICAL APPROACH

After first conducting a review of literature on intelligent software agents and various prediction techniques and algorithms, research will primarily focus on a statistical analysis of data collected while testing the different prediction algorithms. Using a server cluster, numerous percepts files of different sizes will be created to be used in the test runs. These percept files will be used in testing the agent's prediction capabilities in both numerous different test runs will follow, consisting of an analysis of variance and comparisons between varying parameters. This statistical analysis will allow for comparison of data within the Variable Order Markov and Look-Up Table models, as well as comparison to each other.

D. ORGANIZATION OF THESIS

CHAPTER I: Introduction: This chapter discusses the problem as well as a general background on intelligent agents. The problem addresses the purpose of research and a basic introduction to the research methodology.

CHAPTER II: Literature Review: This chapter summarizes literature on artificial intelligence and intelligent agents. The review will focus on three distinct components beginning with a summary of how the intelligent agents learn in complex stochastic environments. Then, a variety of prediction techniques and algorithms, such as compression or prediction by partial match, will be reviewed. Finally, a brief summary of previous research on the system being studied itself will be presented.

CHAPTER III: Application of Variable Order Markov Models to Mental Simulation: This chapter describes the variable order Markov model used by the agent for prediction. The chapter will focus primarily on how general Markov techniques can be applied as a reinforcement learning method.

CHAPTER IV: Research Methodology: This chapter provides an in-depth description of how research was conducted.

CHAPTER V: Data Analysis: This chapter presents charts and graphs showing the relationships between different parameters and prediction probabilities. The analysis will show where and how improvements in the algorithm's prediction can be made.

CHAPTER VI: Conclusions and Recommendations: This chapter provides a conclusion for the study as well as recommendations of areas for future research.

THIS PAGE INTENTIONALLY LEFT BLANK

II. LITERATURE REVIEW

This chapter consists of summaries of academic concepts used in this study as well as previous research and applications. The chapter will cover the concept of intelligent agents, the application of different prediction algorithms, the military application of event prediction, and the statistical techniques used for analysis.

A. INTELLIGENT AGENTS

The primary artificial intelligence used by the predicting agent is the concept of reinforcement learning. Reinforcement learning, in its simplest sense, consists of an agent learning its environment by recognizing percepts and by maximizing rewards or utilities. Richard Sutton and Andrew Barto, in their book *Reinforcement Learning: An Introduction*, further describe a reinforcement learning system of being comprised of four components. First, a policy, usually consisting of a function or lookup table, defines the learning agent's behavior in response to stimuli. Next, a reward function defines the goal of the agent by identifying and quantifying good and bad events in the environment. The value function, the third component, differs from the reward function in that it analyzes the long run of the model. The value function calculates the total amount of rewards that can be collected over the run of the model beginning in its current state. Finally, the fourth component of a reinforcement learning system is the model of the environment, which allows the agent to predict the next state. [3]

One primary issue arising in reinforcement learning is the exploit versus explore problem. This problem comes from the fact that the reward function and value function will not always have the same goals. The theory of exploitation states that the agent should attempt to maximize the utility of each individual action it takes. That is, the agent should always take the action with the highest utility without regard to future steps. Exploration theory contrasts instantaneous utility of exploitation. In particular, an agent could attain better long-term utility by choosing a negative action and "exploring" the environment rather than choosing a positive utility action. The agent must maintain a balance of positive utility actions and exploration, because exploration could lead to learning the environment more efficiently and to more positive actions in the future. [3]

B. PREDICTION ALGORITHMS

The paper, *On Prediction Using Variable Order Markov Models*, by Ron Regleiter, Ran El-Yaniv, and Golan Yona, examines in detail different prediction algorithms utilizing Variable Order Markov Models. The six algorithms explored are Lempel-Ziv 78, LZ-MS, an improved version of Lempel-Ziv, Prediction by Partial Match, Context Tree Weighting, Context Tree Weighting for Multi-Alphabets, and Probabilistic Suffix Trees. Each of these algorithms made predictions on sequences of proteins, English text, and musical notes, with prediction quality measured as the average log-loss similarly to lossless compression. [4]

Lempel-Ziv 78 is based on a popular compression algorithm consisting of both a learning and prediction phase. The algorithm first parses the sequence into adjacent "phrases" and composes a "dictionary" of all distinct phrases that have been parsed. Then, the learning phase constructs a tree and maintains counters for each node of the tree. Additionally, an internal node is maintained equaling the sum of the nodes on that branch. Upon completion of the learning phase, the prediction phase navigates the tree until reaching the condition of the conditional probability, and the probability is equal to the counter of the node divided by the internal counter. The idea behind LZ-MS is to implement "input shifting" and "back-shift parsing" to mine more phrases from the training sequence, allowing for more accurate predictions. [4]

The Prediction by Partial Match algorithm operates similarly to the Lempel-Ziv 78 algorithm in that it learns and predicts using a tree structure, but Prediction by Partial Match incorporates an escape and an exclusion mechanism which reduce the observed sequence at each step, thereby making prediction more accurate. The escape mechanism gives a probability of escape for symbols not appearing after the training sequence and distributes the remaining probability mass to all other symbols that do appear after the sequence. Then, exclusion uses the escape mechanism in that it will ignore the symbols which do not follow the sequence, and thus reduce the observed alphabet. Additionally, Prediction by Partial Match differs from Lempel-Ziv 78 in that it has an upper bound on the maximum order of the Variable Order Markov Model it constructs. [4]

Context Tree Weighting and Context Tree Weighting for Multi-Alphabets are basically the same algorithm, just utilized differently depending on the sequence being predicted. The general idea behind Context Tree Weighting is to construct numerous different “tree sources” from randomly generating sequences when given the initial state. Then, the Context Tree Weighting prediction becomes a mixture of all the tree sources, as it is a summation of each probability with a weighting given to it. Furthermore, the idea is to simply construct tree paths occurring in the training sequence, and then estimate probabilities for other paths. [4]

The Probabilistic Suffix Tree attempts to construct the single best Variable Order Markov Model on the given training sequence. This is done by constructing a tree consisting of unique sequences from each node to the root of the tree, then the Probabilistic Suffix Tree algorithm builds the best tree and calculates the conditional probabilities based on the frequency count. A unique aspect of the probabilities is that to be meaningful the probability must exceed some user-defined threshold. This minimum threshold is assigned to nodes with a frequency count probability of zero. [4]

The results of this study found that different algorithms had better performance for different tasks. Using average log-loss as a measure found that popular lossless compression algorithms, Prediction by Partial Match and Context Tree Weighting were superior in their prediction. However, for the “winner-takes-all” protein classification portion, the modified Lempel-Ziv 78 LZ-MS algorithm proved to be the best predictor. [4]

Another learning algorithm being studied is the concept of Schema Learning. Michael Holmes and Charles Isbell, Jr., in their paper *Schema Learning: Experience-Based Construction of Predictive Action Models*, discuss how Schema Learning can be applied to a speech modeling task. Essentially, Schema Learning can be simplified to a model that sees some action in a certain situation then correctly predict the result from this combination. The schema consists of a specified set of “sensors” which can perceive the models actions. Schema Learning is composed of two distinct phases, discovery and refinement. In the discovery phase, the model sensors develop the action/result relationships simply by observing. Then, in the refinement phrase, contexts, or sensor conditions, are added to the model which increases reliability of prediction, the measure of learning quality. This algorithm was tested against recordings of Japanese speakers saying vowel phrases and resulted in a significant improvement in predictions during the refinement phase when contexts were added to the model. [5]

C. MILITARY APPLICATION

A paper by Dietmar Kunde and Christian Darken, titled *A Mental Simulation-Based Decision-Making Architecture Applied to Ground Combat*, describes a method with which mental simulation and event prediction can be applied in a realistic military setting. The study developed a decision-making framework used for event prediction in a target acquiring process. In the model, a team of blue tanks uses learned knowledge relating to its environment, such as number of targets, visibility, and terrain data, to make decisions regarding the best time to fire at the opposition. The model was built consisting of four components: the simulation environment, situational awareness, the mental simulator, and the decision component. The simulation environment is self-explanatory, as it consists of the battlefield upon which the simulation is taking place. Built upon the simulation environment, the situational awareness component generates the current state around the blue tanks by estimating information about terrain and the enemy. The mental simulator takes the knowledge it has gained from previous situations and makes predictions on future events and outcomes in the battle. Finally, the decision component takes the prediction of the likely next event, the expected future terrain, and a loss assessment prediction for both sides and makes the decision whether to fire or not. This decision component is designed as a context tree, where the nodes are processed down until a “fire” or “hold fire” node is reached. Additionally, a threat assessment to the blue tanks is performed at each node. The study results indicate the model’s performance was consistent with humans performing the same task. This shows the potential for intelligent agent use in event prediction in military applications. [6]

D. STATISTICAL TECHNIQUES

In analyzing the data collected in this study, the primary statistical technique used is a multiple least squares linear regression. In their book, *Stats, Data, and Models*, Richard De Veaux, Paul Velleman, and David Bock define multiple least squares regression as “a linear regression with two or more predictors whose coefficients minimize the sum of the squared residuals.” This linear regression then can test an analysis of variance, t-ratios, and R square values to determine the impact of the regression parameters. [7]

An analysis of variance tests the hypothesis that the regression model provides no improvement over a simple model of the mean. This is tested using an F-ratio, and if the null hypothesis is not rejected the individual coefficients can be looked at with t-ratios. These t-ratios test the hypothesis that the true value of the coefficient is zero, and thus having no impact on the model. T-ratios show whether each individual parameter contributes to the regression. Finally, the R square value determines the percentage of error in the model that can be contributed to the model parameters and the percentage coming from random error. All of these statistics together paint the picture of how and why the model makes its predictions. [7]

THIS PAGE INTENTIONALLY LEFT BLANK

III. VARIABLE ORDER MARKOV MODELS

Variable Order Markov Models are an important and extensively used type of model with numerous application areas such as data compression, identification of DNA sequences, and music composition. The general concept of Variable Order Markov Models, as with all Markov models, is to generate a conditional probability distribution with which to predict the next state of the environment given the present environment state.

A. VARIABLE ORDER MARKOV MODELS VERSUS FIXED ORDER MARKOV MODELS

The easiest way to examine the differences between Variable Order Markov Models and Fixed Order Markov Models is with an example contrasting the two. In this example, each Markov model will create a probability distribution predicting the next character in a character sequence. Consider an infinite sequence of the subsequence *xxyz*, i.e., *xxyzxxyzxxyzxxyz…xxyz*. A Fixed Order Markov Model of order one must estimate nine conditional probabilities for the probability distribution with one character, the current state, as the condition. This model has the following probability distribution for constructing the string: {Pr(x | x), Pr(x | y), Pr(x | z), Pr(y | x), Pr(y | y), Pr(y | z), Pr(z | x), Pr(z | y), Pr(z | z)}. Similarly, a Fixed Order Markov Model of order two has 27 conditional probabilities in its distribution with two characters as the current state. The distribution will have probabilities of the form Pr(x | xy) or Pr(y | zz). A Fixed Order Markov Model of order three follows the same pattern, but produces 81 conditional probabilities in the distribution. The prediction model will estimate the probability distribution based on the fixed order and what has been observed in the environment. If a first order model has observed two subsequences, *xxyzxxyz*, its probability distribution to predict the next character will be, {Pr(x | z) = 1.0, Pr(y | z) = 0.0, Pr(z | z) = 0.0}, because the model has only seen an ‘x’ character follow a ‘z.’

So, it is easily seen that the amount of data needed to estimate the probability distributions in fixed higher order models increases exponentially, meaning there will rarely be enough data for the calculations. For instance, in the above case all 81 three letter sequences would need to have been seen at least once to even begin to accurately estimate the conditional probability distribution. Contrasting the Fixed Order Markov Model, the Variable Order Markov Model of maximum order two can construct the string using only four conditional probabilities: {Pr(x | yz) = 1.0, Pr(y | xx) = 1.0, Pr(z | y) = 1.0, Pr(x | z) = 1.0}. The maximum order in the Variable Order Markov Model is the model's maximum search depth, but as seen in the example above there is order flexibility. This is observed by the combination of first and second order models in the probability distribution. In selecting an order to use, the model chooses the maximum order that will fit, but maintains flexibility to choose lower orders. Order flexibility greatly reduces the amount of data needed for accurate estimates of the probability distribution. In addition to this, computational speed is increased due to the model being constructed as a context tree.

A context tree easily shows the model's distribution at each possible depth, as each node contains a distribution of possibilities, such as a distribution of possible letters in the above example. The depth of the node in the tree equates to the order of the Variable Order Markov Model. Zero depth, or order, corresponds to the unconditional distribution of all possibilities. A depth of one is the normal Markov model, which takes into account the most recent environment state when making predictions. Figure 1 below shows a context tree of maximum depth two of the *xxyzxxyzxxyz…xxyz* string. Each branch of the tree represents the recent environment states the model is taking into account, while each node represents the distribution of possible characters. Thus, if the model is taking into account that the last state was 'x,' the two characters it can predict might come next are 'x' and 'y,' each with a probability of 0.5. Then, at the order two nodes, the two most recent characters are taken into account. For example, the character sequence of 'zx' leads to a prediction probability of 1.0 that the next character is 'x.' Each branch of the tree terminates when it reaches a node with only one possible character, meaning it will be predicted 100 percent of the time. Figure 1, though, shows the full context tree with maximum depth of two to show the redundancy of extending branches that have already reached a terminal state.

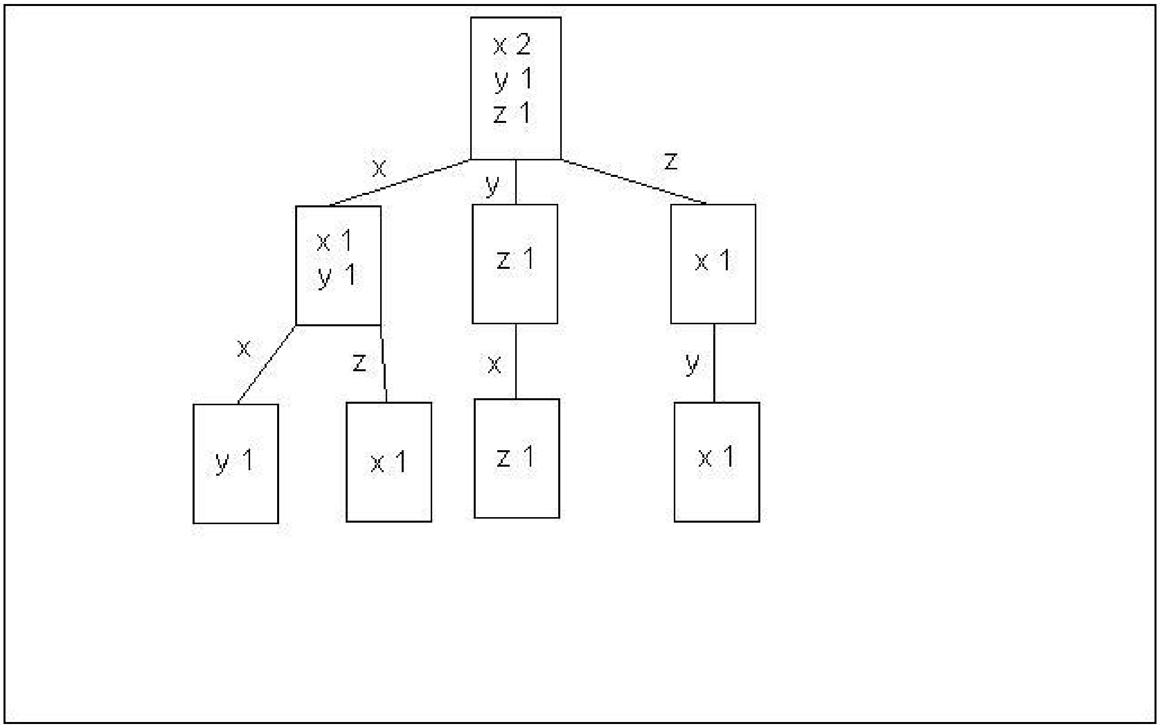


Figure 1. Context Tree with Maximum Depth of Two.

So, the two primary characteristics of Variable Order Markov Models, order flexibility and context trees, allow for a much more accommodating model than Fixed Order Markov. Therefore it is an exceptional method for prediction of our perceptual model.

B. VARIABLE ORDER MARKOV MODEL FOR PERCEPT PREDICTION

Percept prediction benefited greatly from the flexibility of the Variable Order Markov Model and the model was able to process and make predictions on files containing hundreds of thousands of percepts.

There are three key components in the Variable Order Markov Model for percept prediction: the releventL array, the predicted array, and the match depth. The relevantL array keeps track of the recent environment state in the model and contains the last ‘n’ percepts that have been processed, where ‘n’ is the maximum order of the model. This percept array is the primary factor used for prediction in the conditional probability distribution, which is represented as the predicted array. The prediction array can range from the unconditional probability distribution, where all previous percepts that have been processed are predicted with equal probability, to a single precept with probability 1.0. If the correct next percept is present in the predicted array, the model gets credit for whatever fraction said percept was given in the probability distribution. For example, if the predicted array contains 10 percepts each with equal likelihood and one of the percepts is correct, the model receives credit for 0.1 for the correct percept predicted. Finally, the match depth describes how the context tree was used to make the predictions by describing which orders, or depths, the relevant percepts match the percepts that have already been processed by the model. That is, how many and what size sequences of percepts that the model has already seen.

IV. RESEARCH METHODOLOGY

This chapter will describe the technical approach and techniques used in conducting this research, focusing on server clusters, simple scripts, percept generation, after action review, and summary files.

A. SERVER CLUSTERS

Conducting research consisted primarily of many runs of different Python-language programs. These experiments, though, required much more computing power than a basic desktop computer could provide. Thus, four Naval Postgraduate School servers, Boreus, Eurus, Zephyr, and Notus, in a computing grid, were utilized. Over the course of conducting research, the Notus server went down for repairs, so only three servers were used thereafter.

Accessing the computing grid consisting of the three servers required a secure connection, so before any processes were undertaken a secure shell needed to be set up. Setting up the secure shell would then allow the use of an ssh-agent to allow communication to all three servers at the same time from a remote source. This was a relatively simple process that only required putting a public-key in a file on each server to allow for authentication.

With a secure connection between the remote source and the server cluster, numerous processes could be done simultaneously, allowing for a greater amount of data being collected. The inter-connection of the computing grid allowed the ssh-agent to use a simple script to dispatch different jobs to each of the three servers simultaneously. Because the servers were connected and accessed through the same secure connection they could communicate to each other and work together to accomplish a task.

B. SIMPLE SCRIPTS

With the secure connection established and the servers working together, the next step was to utilize a script to dispatch jobs to each server when they were not busy with another task. The script was primarily used for conducting the "after-action review" (to be discussed below) with different sets of parameters. The script used was just a very short Python program designed to accomplish a few tasks at a time. Basically, the script only consists of the host servers and a list of the jobs to be accomplished. Each job includes all of the files needed to complete the task as well as the files to be collected as output. The script simply goes down the list of jobs and dispatches them to a server if it is available for a task.

The foremost benefit of the script is that several tasks can be accomplished at one time with only one line typed into the command line. Since each task could be quite time consuming, the script allowed more data to be collected with less time actually spent at the remote source. In addition, all of the output files were collected on the single server from which the script was run, easing the data collection process.

C. PERCEPT GENERATION

With the system now optimally set up to collect data, different percepts files must be generated, as percepts are what the after-action review uses to make predictions. Percepts are reports on the state of the game environment at different times, read as four different types: actions, events, and plus or minus sensations. Action percepts correspond to actions the player or other characters take, such as moving in a particular direction or picking up some object. Events typically correspond to a particular action, for example going in the direction of the action or the event of dropping an object. Plus and minus sensations are related to what is in the player's presence at some time. For example, when the player "Spock" enters "Paperville," a plus sensation is received, giving Spock's location as Paperville. Then, when Spock moves in some direction, a minus sensation is received as he leaves the location. These percepts reflect the state of the environment at all times, since every event, action, or sensation is accompanied by a time stamp.

In order to generate these percept files, two things must happen. First, the game environment must be set up for the player to maneuver through and take actions. But, in order to analyze the agent's prediction capabilities, numerous percept files of different sizes are needed. So, once the game environment is set up, a second agent is needed to travel through the environment taking random actions and recording the state of the game. This agent bypassed the otherwise tedious task of having to take and record thousands of actions, on which the learning algorithms could then make predictions.

Both of these actions were fairly simple, though, as we already had programs in place to perform these functions. Implementing the two programs was done in the background of the servers with minimal processing so the servers could also perform other tasks while the percept files were being generated. The first program, mudserver, set up the game environment for the agent to wander through. Then, the agent program, with parameters for the agent, to both take actions and write the percepts it receives to a separate file. These two programs ran in tandem for various lengths of times to generate percept files of different sizes. The most commonly used run was one of 24 hours, generating percept files of approximately 115,000 percepts, but various other durations were used to generate files from 3,000 to 485,000 percepts.

D. AFTER-ACTION REVIEW

The bulk of work went into the after-action review program, where the offline learning agent makes predictions based on the percept file it is given. In addition to a percept file, the after-action review requires two parameters, a learning model and the learning model's constraint. The two learning algorithms analyzed in this study were the Variable Order Markov Model, with a constraint of model order, and the Look-Up Table Model, with a constraint of time. With each learning algorithm, the basic goal of the review is to process each percept one at a time and learn to predict what the next percept will be as the agent progresses through the file. Each learning model, though, learns and makes its predictions in a different way.

The variable order Markov model, described in detail in Chapter III, bases its prediction on two main factors: relevant percepts and search depth. Relevant percepts completely depend on the model's order parameter. The relevant percept array keeps track of the last 'n' percepts that have been processed, regardless of percept type, where 'n' is the order used in the model. For example, if the model uses a Markov search depth of five, the previous five percepts that have been processed are kept in the relevant percept array. In the variable order Markov model, the relevant percept array is given the highest weight within the agent making its predictions.

Search depth is another factor directly related to the model's order parameter. The search depth is the maximum depth the Markov model will search through prediction possibilities, meaning with an order of five the algorithm will attempt find a prediction match at orders one, two, three, four, and five. Taking into account the sensation array, the Markov model searches the relevant percepts up to its maximum order and generates a prediction probability distribution in an attempt to correctly predict the next percept in the sequence.

The Look-Up Table Model uses a somewhat similar algorithm to the Variable Order Markov Model, but it uses different parameters and places higher emphasis on different things. First, the parameter for the Look-Up Table is time, a recency threshold on both percepts and sensations. For example, using the look-up table model for prediction with a time parameter of 0.1 seconds creates a sensation array of all plus percepts that have been seen in the last 0.1 seconds (without a corresponding minus percept) as well as a relevant percept array of all percepts seen in the last 0.1 seconds. The agent then attempts to exactly match these arrays to patterns that have already been seen in the percepts that have been processed, or percepts that are already in the look-up table. Making predictions with the look-up table algorithm consists of again creating a probability distribution of possible percepts based on these two arrays.

Both the variable order Markov model and the look-up table model keep a running tally of the average prediction probability as predictions are made about each subsequent percept. Though the models make their predictions based on different factors, each model calculates its average prediction probability the same way. If the next percept is correctly predicted in the probability distribution, the average prediction probability gets credit for whatever fraction of the distribution that percept had. For example, the look-up table predicts the following probability distribution: [(['feistyE', 'spock86'], 0.33333333333333331), (['tiredE', 'spock86'], 0.66666666666666663)]. The actual next percept is ['E', 'feisty', 'spock86'], so the average prediction probability gets 0.33333333333333331 factored in to the average.

Conducting research with the after-action review consisted of numerous trials using many different percept files and many different parameters. To analyze the impact of the number of percepts on prediction probability, percept files ranging from 1,000 to 485,000 percepts were used. The primary analysis, though, consisted of 15 different percept files of approximately 120,000 percepts each. These files were then tested using the Variable Order Markov Model with orders two through fifteen as well as with the Look-Up Table Model using parameters 0.01, 0.1, 0.5, 1.0, 2.5, 3.5, 5.0, 10.0, and 20.0 seconds respectively. Each run of the after-action review produced an output file which could then be further analyzed using a different program.

E. SUMMARY FILES

After completing the after-action review, each output file was analyzed further with a summary program. The summary program is very basic, as it collects information from the after-action review output file and presents it so as to make it easier to analyze. The information presented in the summary files includes average prediction probability for all percepts and the following information for each individual percept seen and aggregates of each percept type: number of occurrences, average prediction probability, error (one minus average prediction probability multiplied by the number of occurrences), average prediction probability over the last 100 occurrences, and number of times a prediction probability of zero occurred in the last 100 occurrences. This information was then used for further analysis as covered in Chapter V.

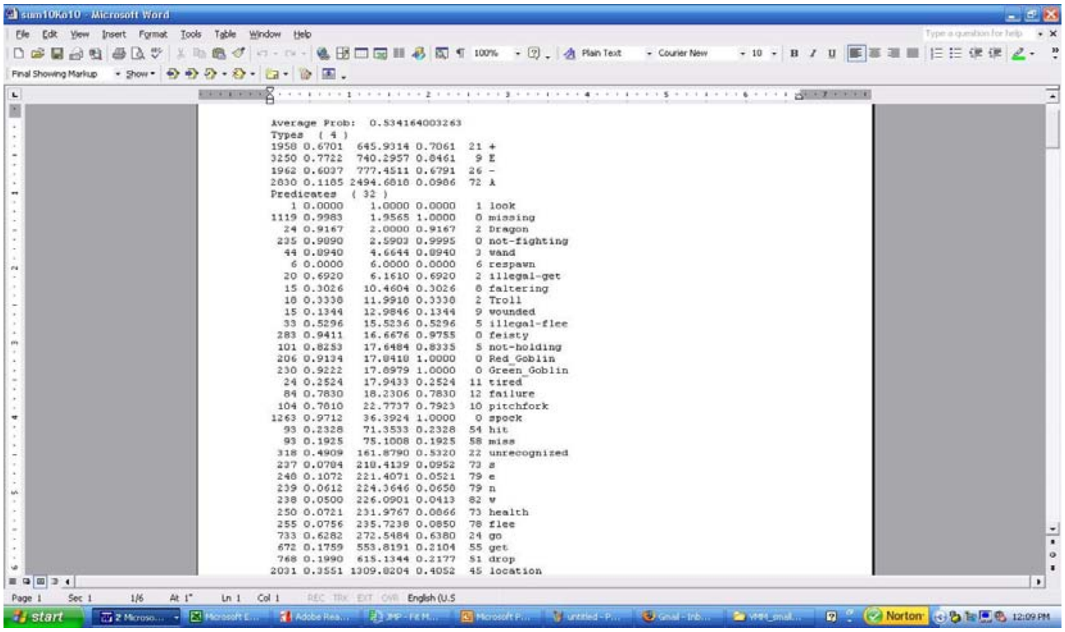
THIS PAGE INTENTIONALLY LEFT BLANK

V. DATA ANALYSIS

This chapter provides an in-depth look at the data collected via charts, graphs, and a statistical analysis. The analysis will focus on the relationship between both the order, used in the Variable Order Markov Model, or time threshold, used in the Look-Up Table Model, and the number of percepts on the average prediction probability. The statistical analysis of both the Variable Order Markov Model and the Look-Up Table Model consist primarily of a multiple factor linear regression and an analysis of variance between the same factors.

A. VARIABLE ORDER MARKOV MODEL

This analysis primarily examines 15 randomly generated percept files of similar size, ranging from approximately 105,000 to approximately 135,000 percepts. Each of these files was then run through the after-action review program with maximum model orders of two through fifteen. Furthermore, the data from the summary files, an example of which is seen below in Figure 2, were then compiled into spreadsheets and graphs.



```
Average Prob:  0.534164003263
Types  ( 4 )
1958 0.6701  645.9314 0.7061  21 +
3250 0.7722  740.2957 0.8461   9 E
1962 0.6037  777.4511 0.6791  26 -
2030 0.1105 2494.6818 0.0986  72 A
Predicates  ( 32 )
   1 0.0000    1.0000 0.0000   1 look
1119 0.9983    1.9565 1.0000   0 missing
  24 0.9167    2.0000 0.9167   2 Dragon
 235 0.9890    2.5903 0.9995   0 not-fighting
  44 0.8940    4.6644 0.8940   3 wand
   6 0.0000    6.0000 0.0000   6 respawn
  20 0.6920    6.1610 0.6920   2 illegal-get
  15 0.3026   10.4604 0.3026   8 faltering
  18 0.3338   11.9918 0.3338   2 Troll
  15 0.1344   12.9846 0.1344   9 wounded
  33 0.5296   15.5236 0.5296   5 illegal-flee
 283 0.9411   16.6676 0.9755   0 feisty
 101 0.8253   17.6484 0.8335   5 not-holding
 206 0.9134   17.8418 1.0000   0 Red_Goblin
 230 0.9222   17.8979 1.0000   0 Green_Goblin
  24 0.2524   17.9433 0.2524  11 tired
  84 0.7830   18.2306 0.7830  12 failure
 104 0.7810   22.7737 0.7923  10 pitchfork
1263 0.9712   36.3924 1.0000   0 spock
  93 0.2328   71.3533 0.2328  54 hit
  93 0.1925   75.1008 0.1925  58 miss
 318 0.4909  161.8790 0.5320  22 unrecognized
 237 0.0784  218.4139 0.0952  73 s
 248 0.1072  221.4071 0.0521  79 e
 239 0.0612  224.3646 0.0650  79 n
 238 0.0500  226.0901 0.0413  82 w
 250 0.0721  231.9767 0.0866  73 health
 255 0.0756  235.7238 0.0850  78 flee
 733 0.6282  272.5484 0.6380  24 go
 672 0.1759  553.8191 0.2104  55 get
 768 0.1990  615.1344 0.2177  51 drop
2031 0.3551 1309.8204 0.4052  45 location
```

Figure 2. Variable Order Markov Model Summary File

1. Average Prediction Probability versus Model Order

The results of the 210 trials of the Variable Order Markov Model indicate that average prediction probability increases in a logarithmic fashion with respect to increased maximum order. All 15 trials at each order produced similar results indicating that the increased information provided to the model by higher maximum order led to better prediction capability.

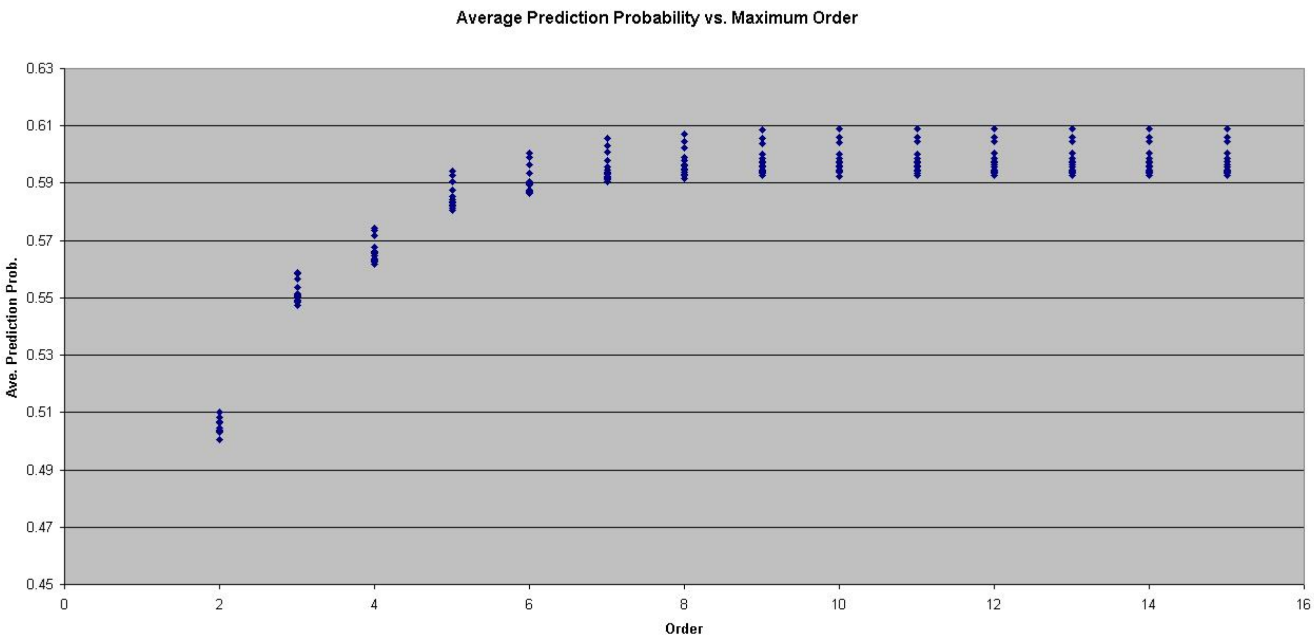


Figure 3. Average Prediction Probability versus Maximum Order

Figure 3 above shows the overall average prediction probability at each order for each of the 15 trials. The probabilities range from the minimum of approximately 0.50 to the maximum of approximately 0.61. Thus, Figure 3 illustrates the logarithmic nature of the increase in average prediction probability up to the apparent asymptote of 0.61. In addition, the small variation within the 15 trials at each order is also seen, with a maximum variation of one and a half percent at any given order.

Figure 3 also illustrates the significant growth in average prediction probability with an increase of one in maximum order up until order seven, where increases in average prediction probability become insignificant. After maximum order seven is reached, the increase in average prediction probability from any single difference of order one ranges from one quarter to one one-hundredth of one percent.

These findings are significant because they show how maximum order affects the model and how the model learns to predict. First and foremost, these results show that the Variable Order Markov Model achieves better prediction with more information. This fact is intuitive, as the model needs to know about its environment in order to make predictions about it. The maximum order determines the number of percepts in the relevant percept array, so clearly the higher order model can use more information about its environment for each prediction. Particularly, the higher order model incorporates more sensation percepts in its relevant percept array, which allows the model to make more correct '-' percept predictions. This can be seen in the 31-percent increase in average prediction probability from order two to order eight in Figure 4 below.

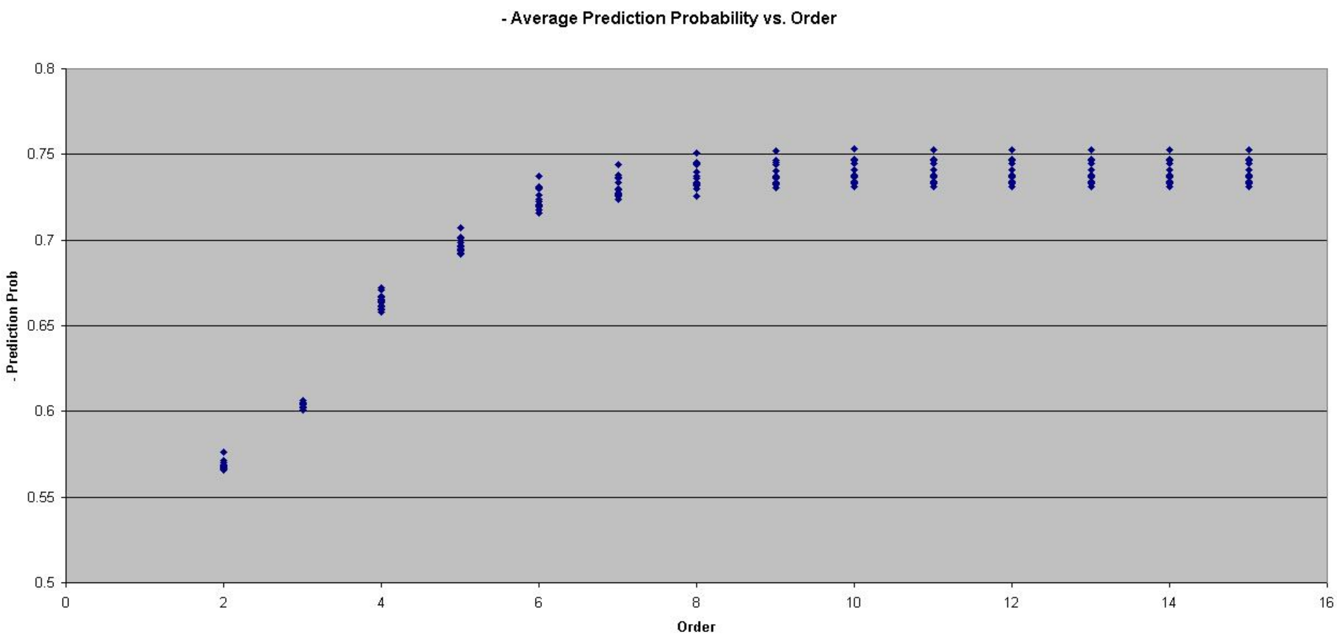


Figure 4. '-' Percept Average Prediction Probability versus Maximum Order

What this does not address, though, is why increases in maximum order past order seven result in insignificant increases in average prediction probability. This fact is related not only to the information contained in the relevant percept list, but also the match depth of the model. Seemingly, the extra information given by higher orders would always lead to better prediction probabilities. This is deceiving, though, because although higher order will always give more percepts, these percepts may not always be relevant. For instance, when the ‘go’ action is taken, Spock moves to a completely new environment. Thus, if the maximum order is 10, the relevant percept array will contain nine percepts that really are not relevant at all. The model will be making its prediction primarily based on an environment in which it is no longer present. In addition to this, higher maximum order has diminishing returns in the match depth. Rarely does the model encounter situations in which the same 10 percepts were seen in a row in the same order, let alone the same 15 percepts. No matter what the maximum order is, the model always finds matches in the context tree at low orders, such as one, two, or three. The difference between any two successive orders is constantly diminishing, as matches in the context tree at the maximum order and one less than the maximum order sum to the matches in the context tree at the maximum order of the previous model. For example, in going from an order 10 model to an order 11 model, the context tree matches at the order 10 model’s maximum order are equal to the matches at order 10 and 11 in the order 11 model. Some of depth 10 matches will match one more branch down to depth 11, but many will not. So, with each successive increase in order the match depths in the context tree diminish, meaning improvements will still be seen in the average prediction probability, but the increases will be continually smaller.

The other significant fact seen in the comparison between average prediction probability and maximum order comes from the small variation in all of the data groups. Particularly, the low variation in each aspect of the model shows the consistency in how the model makes predictions on the random percept files. This shows the reliability of the model and that it is not simply making random predictions and getting lucky.

2. Average Prediction Probability versus Number of Percepts

In addition to the maximum model order, the number of percepts on which the model is making predictions also impacts the average prediction probability. This fact should again be intuitive, as analyzing more percepts means the model sees the same environment situations more often. When the model sees and learns situations in the environment, the context tree can make more matches and matches at greater depths, leading to an increase in the model's prediction capabilities.

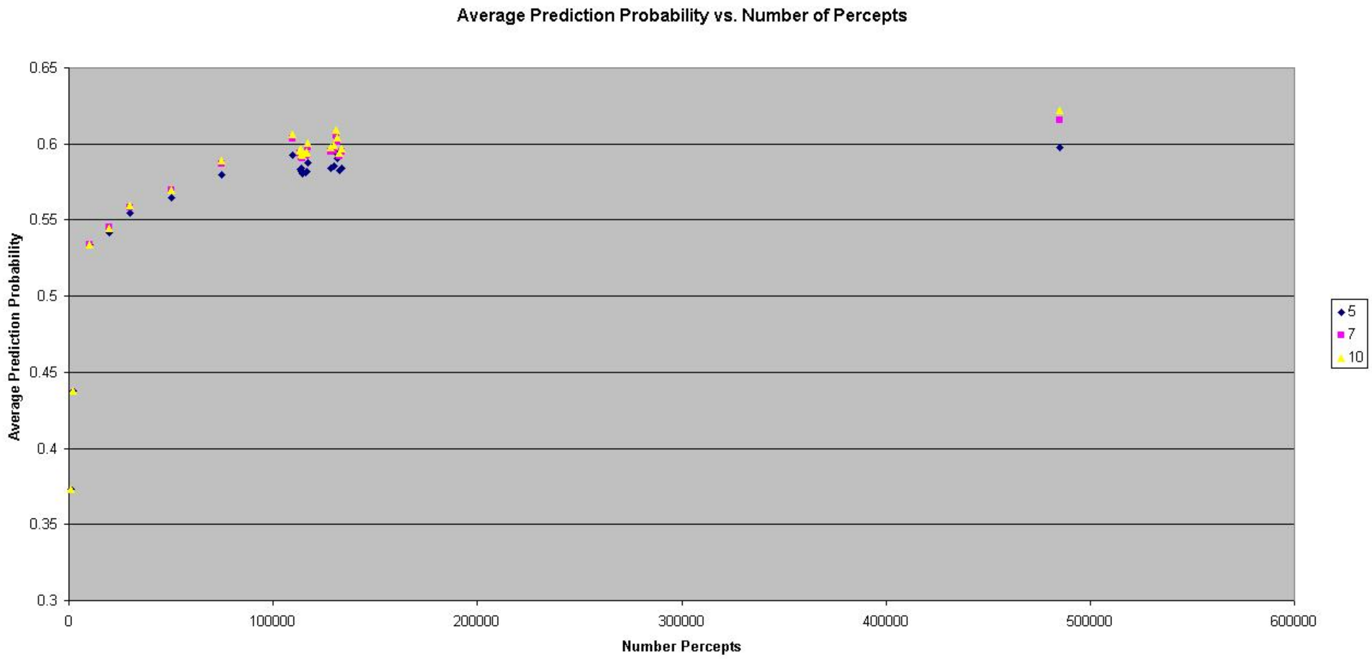


Figure 5. Average Prediction Probability versus Number of Percepts for Orders Five, Seven, and Ten

Figure 5 shows the average prediction probabilities of the same 15 percept files mentioned above as well as random files ranging from 1,000 to 75,000 percepts and a large file containing 484,000 percepts. Each of the file's average prediction probabilities are shown at orders five, seven, and ten. The increases in average prediction probability again increase roughly logarithmically with respect to the increase in the number of percepts. The graph also shows increases in the number of percepts analyzed decreases the diminishing returns aspect of increases in order. At the 1,000 percept level, there is no difference in the average prediction probabilities of orders five, seven, and ten, while at the 100,000 percept level there is only a total difference of one and a half percent between order five and ten. Contrastingly, the difference between order five and ten on the 484,000 percept trial is two and a half percent due to more percepts leading to increased and greater depth context tree matches.

Comparing average prediction probability to the number of percepts analyzed also provides further evidence to the consistency of the prediction model. Figure 6 below shows the average prediction probabilities for orders two, three, four, five, seven, and ten for the 15 percept files. The distributions of each order have identical patterns at different average prediction probabilities, illustrating the reliability of the model.

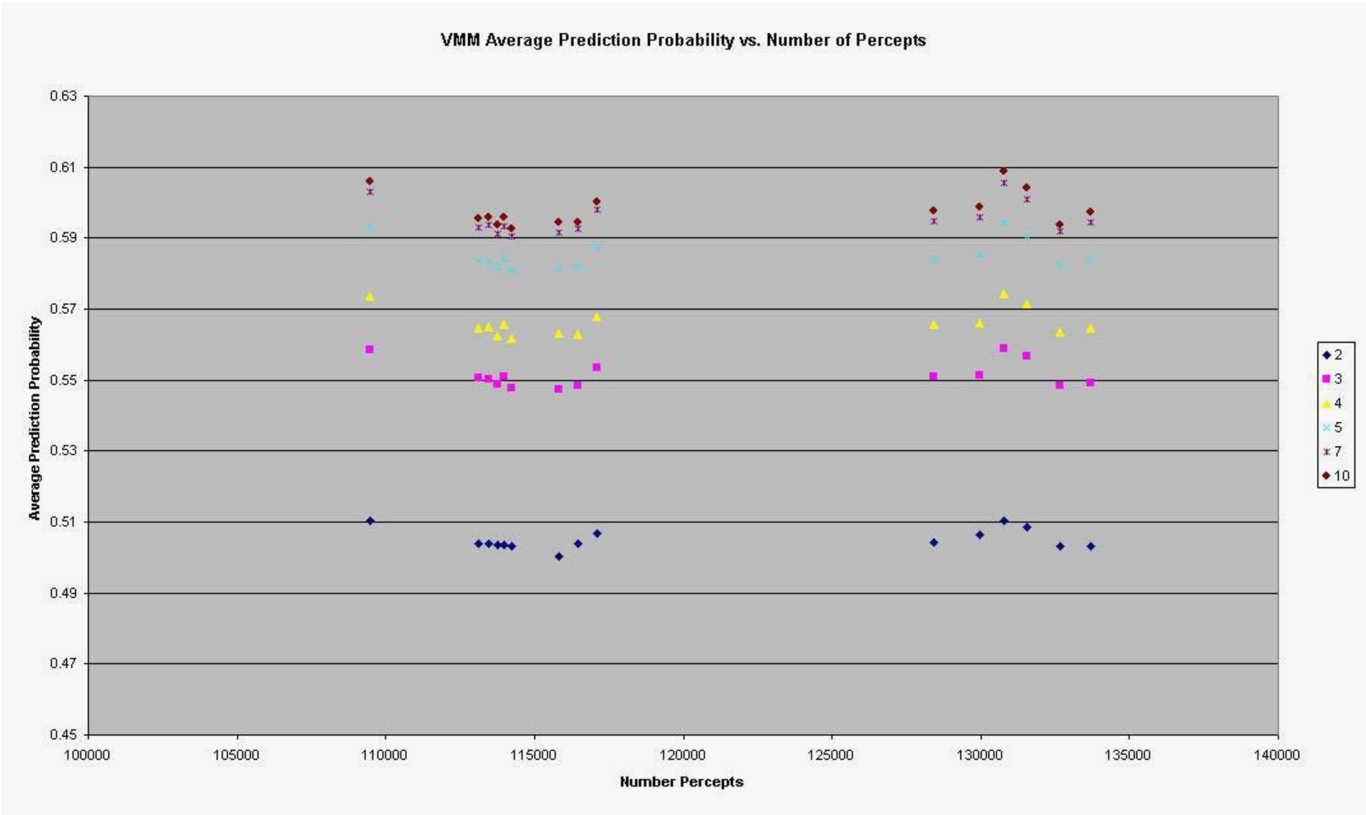


Figure 6. 15 Trial Average Prediction Probability versus Number of Percepts

3. Statistical Analysis

While the graphs above provide significant insight into predictions made by the model, a statistical analysis provides further information as to how the predictions are made and how important certain factors are. This statistical analysis is a multiple linear regression, consisting of a plot of average probability predicted by the model, leverage plots of order and number of percepts versus average prediction probability, a summary of fit of the model, an analysis of variance, parameter estimates, and an effects test of order and number of percepts.

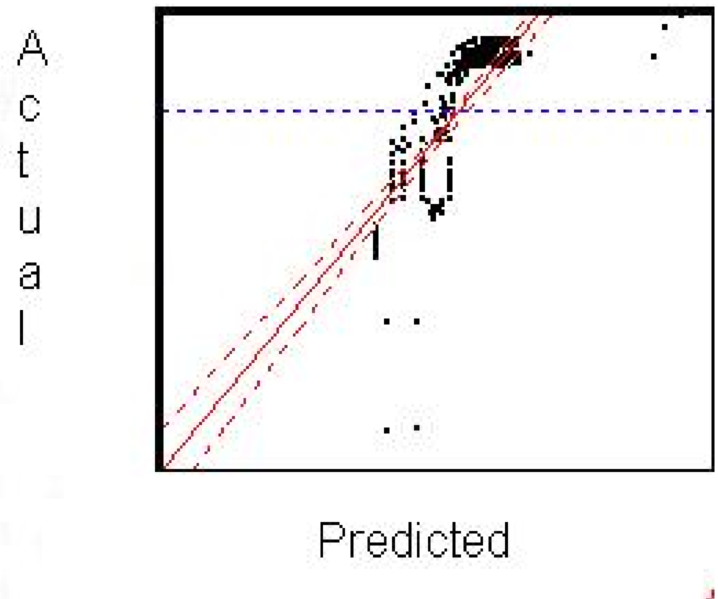


Figure 7. Actual versus Predicted Average Prediction Probability

Term	Estimate	Standard Error
Intercept	0.4936323	0.004105
Order	0.0042562	0.000378
Number of Percepts	0.0000003985	0.0000000247

Table 1. Parameter Estimates for Line of Best Fit of Model

Figure 7 above shows a plot of the actual average prediction probability versus the average predicted by the statistical model for all Variable Order Markov Model trials. Each point on the plot represents the average prediction probability for one trial, and the x-coordinate of that point represents average probability that the model predicts based on model order and the number of percepts. The solid red line is the model's line of best fit, constructed from the parameter estimates in Table 1. Basically, this line is compared to the dashed blue line, the overall mean of the average prediction probabilities, to determine which line fits the model better. The dashed red lines are confidence curves for the line of fit, and by intersecting the overall mean of the model, the confidence curves indicate the test of fit is significant with 95 percent confidence.

Whereas Figure 7 displays the model taking into account both order and the number of percepts, Figures 8 and 9 show simple plots of average prediction probability versus order and average prediction probability versus number of percepts respectively. Both these figures have the same components as Figure 7, line of fit, mean average prediction probability, and confidence curves, but the effect of both order and number of percepts are separated. Figure 8 indicates a positive correlation between average prediction probability and model order, as well as order being a significant factor with 95 percent confidence. Figure 9 indicates both of the same things for number of percepts.

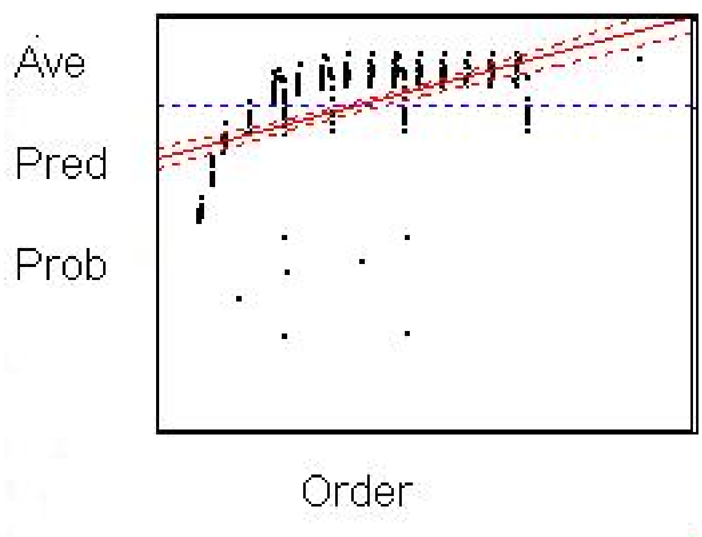


Figure 8. Average Prediction Probability versus Order with Line of Fit

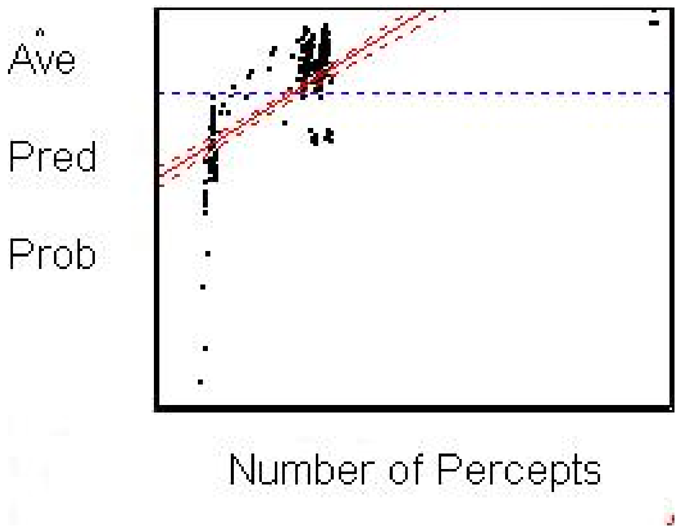


Figure 9. Average Prediction Probability versus Number of Percepts

The summary of fit of the model, shown below in Table 2, provides numerical summaries describing the fit of the model. RSquare estimates whether variation in the model comes from parameters or random error. A RSquare of zero indicates the model fits no better than the overall mean, while a RSquare of one means a perfect fit. The RSquare of this model estimates that 58 percent of the variation around the mean can be attributed to either order or the number of percepts rather than random error. This number clearly provides further proof that order and number of percepts have an impact on average prediction probability, though there is still plenty of random error present. Adjusted RSquare describes the same information, but calculation of adjusted RSquare takes into account the degrees of freedom of the model and its parameters, making it more comparable with models having different numbers of parameters. The root mean square error estimates the standard deviation of random error present in the model. This model's low root mean square error further displays the relative lack of variation in the Variable Order Markov Model and its consistency in making predictions.

RSquare	0.580654
RSquare Adjusted	0.577821
Root Mean Square Error	0.026665
Mean of Response	0.56676
Observations	299

Table 2. Summary of Fit Data for the Model

The analysis of variance of the model, shown below in Table 3, further demonstrates that the parameters order and number of percepts have an impact on average prediction probability and that the model constructed from the Variable Order Markov Model data fits better than the simple mean average prediction probability model. The key statistic in the analysis of variance is the F-ratio, which is used to test the hypothesis that all the regression parameters in the line of fit are equal to zero. In Table 3, the large F-ratio and related probability indicate that there is almost no chance of obtaining a larger F-ratio solely by chance. The fact that the probability is less than 0.0001 means there is evidence that there is at least one significant regression parameter in the model, meaning there is evidence that order and/or number of percepts has a significant impact on average prediction probability.

Source	**Degrees of Freedom**	**Sum of Squares**	**Mean Square**	**F-ratio**	**Probability > F**
Model	2	0.2914137	0.14570685	204.930522	< 0.0001
Error	296	0.2104578	0.00071101		
Total	298	0.50187151			

Table 3. Analysis of Variance for Variable Order Markov Model

Finally, the multiple regression statistical analysis provides an effects test to analyze the impact of order and the number of percepts separately. The effects test again uses an F-ratio to test the hypothesis of each parameter being zero. Once again, as seen in Table 4, both order and number of percepts have large F-ratios, providing sufficient evidence that both have a strong impact on average prediction probability. The interesting aspect of the effects test, though, is that the number of percepts has a higher F-ratio than order, indicating that the number of percepts being analyzed has a bigger impact than order. This is due to the fact that differences in average prediction probability with respect to order differ only by small amounts when compared between similar numbers of percepts. Specifically, with a very low number of percepts, there is absolutely no difference in average prediction probability when order charges. Since these averages are much lower than the majority of trials, number of percepts appears to have a slightly greater impact than it really does, though both F-ratios are clearly high enough to provide evidence of significance for both order and number of percepts even without this effect.

Source	**Degrees of Freedom**	**Sum of Squares**	**F-ratio**	**Probability > F**
Order	1	0.0900259	126.617626	< 0.0001
Number of Percepts	1	0.18493996	260.110237	< 0.0001

Table 4. Effect Test Table for Variable Order Markov Model

B. LOOK-UP TABLE MODEL

The analysis of the Look-Up Table Model was done the same way as the Variable Order Markov Model, but with recency time threshold substituted as a parameter for order. The same 15 percept files subject to the after-action review with each of the time thresholds in the following set: {0, 0.01, 0.1, 0.5, 1.0, 2.5, 3.5, 5.0, 10.0, 20.0}. Once again, summary files were compiled into spreadsheets.

1. Average Prediction Probability versus Time Threshold

Intuitively, one would assume the Look-Up Table Model works similarly to the Variable Order Markov Model and that average prediction probability improves with increased recency threshold, as the model is given more information with a larger time threshold. In fact, though, the opposite was found to be true. Prediction is done by the Look-Up Table Model using an exact matching technique, meaning the agent attempts to match exact patterns of percepts to ones that have already been seen. The algorithm looks for matches in both the sensation and relevant percept array which are made up of all sensations or percepts that have occurred in the environment within the last ‘t’ seconds, where ‘t’ is the recency time threshold. In addition, the Look-Up Table Model must match all percepts more recent than the time threshold, not just some percepts like the Variable Order Markov Model can. Clearly, since the environment consists of random actions, exact matches of much longer patterns happen less often. Therefore, the model yields much better average probability predictions with smaller time threshold, as seen below in Figure 10.

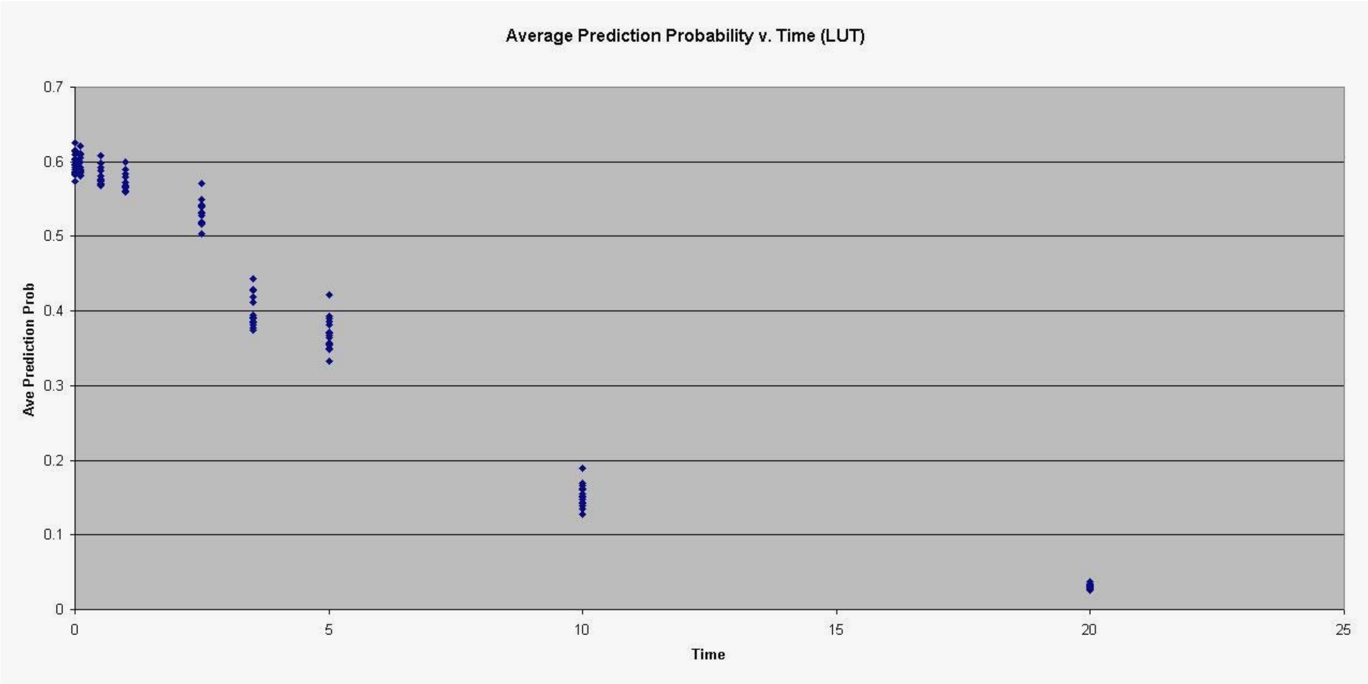


Figure 10. Average Prediction Probability versus Time Threshold

The two key characteristics observed in Figure 10 are the quick decline in average prediction probability with increased recency time threshold and the differences in variation at different thresholds. With a fast declining average prediction probability, a Look-Up Table Model with a time threshold of 20 seconds produces an average prediction probability of almost zero. So, the Look-Up Table Model's average prediction probability does not just decrease with increased time, it plunges. By comparison, the Variable Order Markov Model still has an approximate 50 percent average prediction probability at its lowest level. Additionally, the middle time threshold levels of 2.5, 3.5, 5.0, and 10.0 seconds show much more variation than the comparable low or high time thresholds. These middle stage models are much more sensitive to chance in exact matching. The low time threshold models contain only a few percepts in each array, thus exact matches are found very frequently. The very high time threshold models, on the other hand, rarely find any exact matches as there can be close to 100 percepts in certain arrays, so both low and high time thresholds display less variance. Contrastingly, mid time thresholds will find exact matches some of the time, but their medium sized percept arrays lead to more variance between trials.

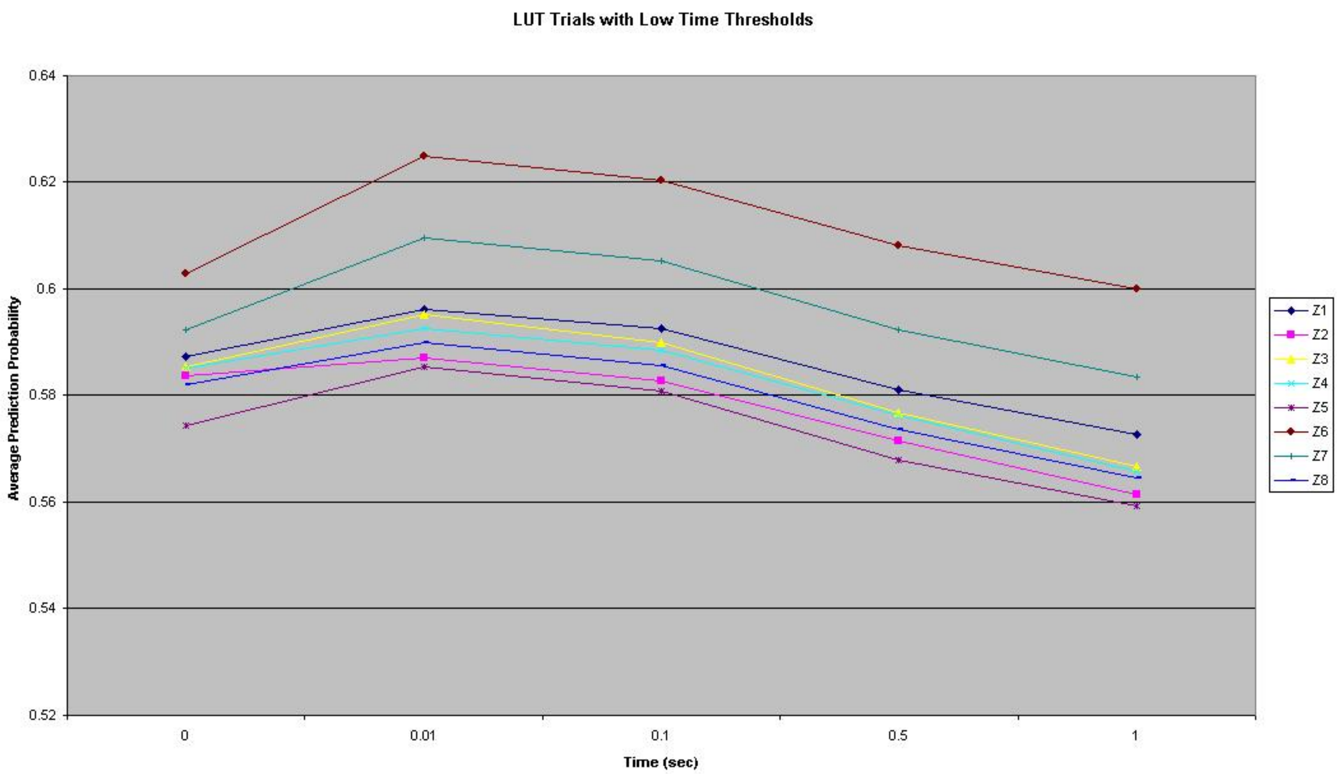


Figure 11. Averages from Low Time Threshold Trials for Eight Percept Files

Despite the decreasing nature of average prediction probability in the Look-Up Table Model with increased recency time threshold for percepts, the model does show an increase in prediction over a time threshold of zero. Eight of the 15 percept files were analyzed with a recency time threshold of zero seconds and compared with the other small time threshold trials, as seen in Figure 11 above. Figure 11 shows the increase in average prediction probability from 0.0 to 0.01, followed by the beginning of the decline.

It could be expected that with a recency time threshold of zero seconds, no predictions could be made about an environment at all, as it should have no information. The Look-Up Table Model, though, still displays a reasonably good average prediction probability. This comes from the fact that the Look-Up Table Model bases its predictions entirely on recent sensations in its exact matching prediction when there is no time threshold for relevant percepts. In fact, this is easily seen by comparing the average prediction probabilities of '+' and '-' type percepts. Below, Figure 12 shows the average prediction probability for '+' percepts and Figure 13 shows the average prediction probability for '-' percepts. In Figure 12, a steep increase is seen in the average prediction probability from time threshold 0.0 to time threshold 0.01 because information about the environment allows the model to predict when things will be sensed, which '+' percepts represent. But, Figure 13 shows that '-' percept prediction probability actually decreases from time threshold 0.0 to time threshold 0.01. Since predictions at time threshold zero are based solely on sensation, the model will predict '-' percepts more often, since '-' percepts always correspond to the '+' percepts found in the sensation array.

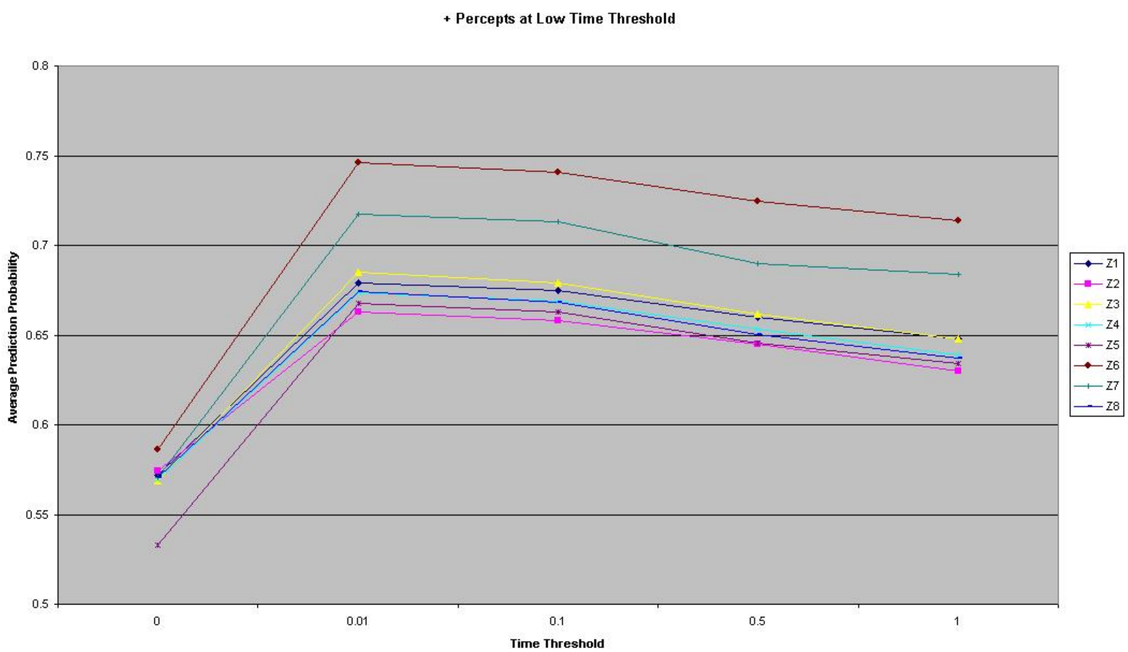


Figure 12. Average Prediction Probabilities for '+' Percepts

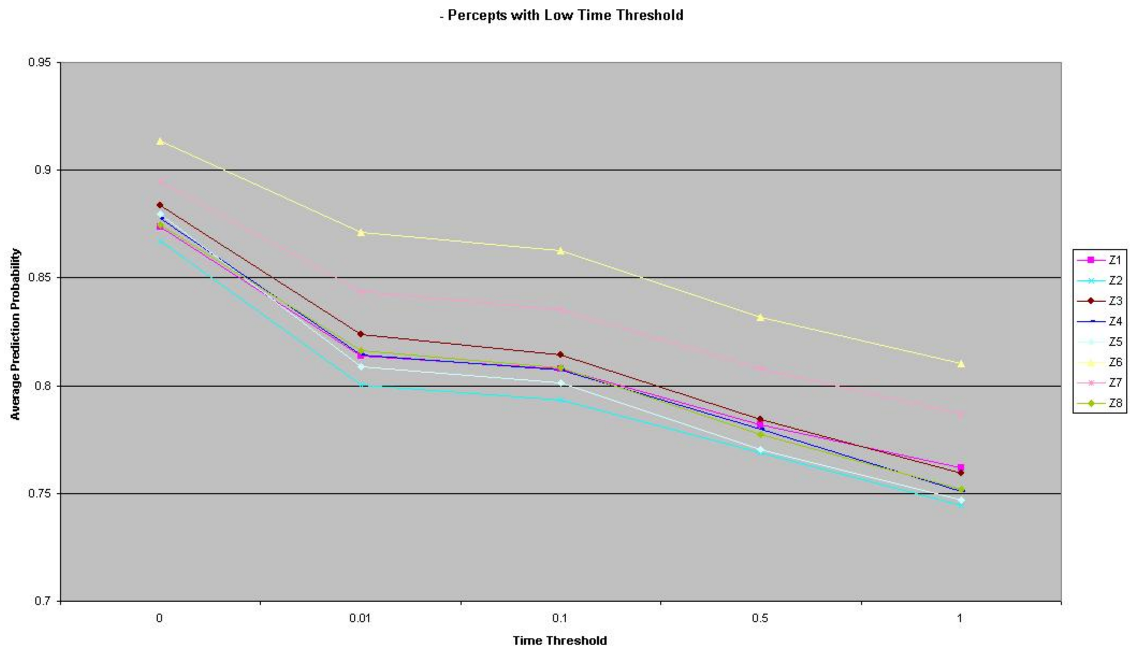


Figure 13. Average Prediction Probabilities for '-' Percepts

2. Average Prediction Probability versus Number of Percepts

Evaluating the Look-Up Table Models average prediction probability against the number of percepts once again shows the consistency of the model. Figure 14 below shows each of the averages of each of the 15 trials with seven different time thresholds. For the most part, the patterns of each trial at the different time thresholds are nearly identical, but with some of the variance describe above seen between the higher threshold trials. Even with this variance, though, the patterns are still very similar. The lower time threshold trials also exhibit almost identical patterns between each trial. So, the Look-Up Table Model displays the same consistency of the Variable Order Markov Model. In addition, the Look-Up Table Model would show the same increase in average prediction probability with increased number of percepts seen by the Variable Order Markov Model, but trials were not conducted to illustrate this fact.

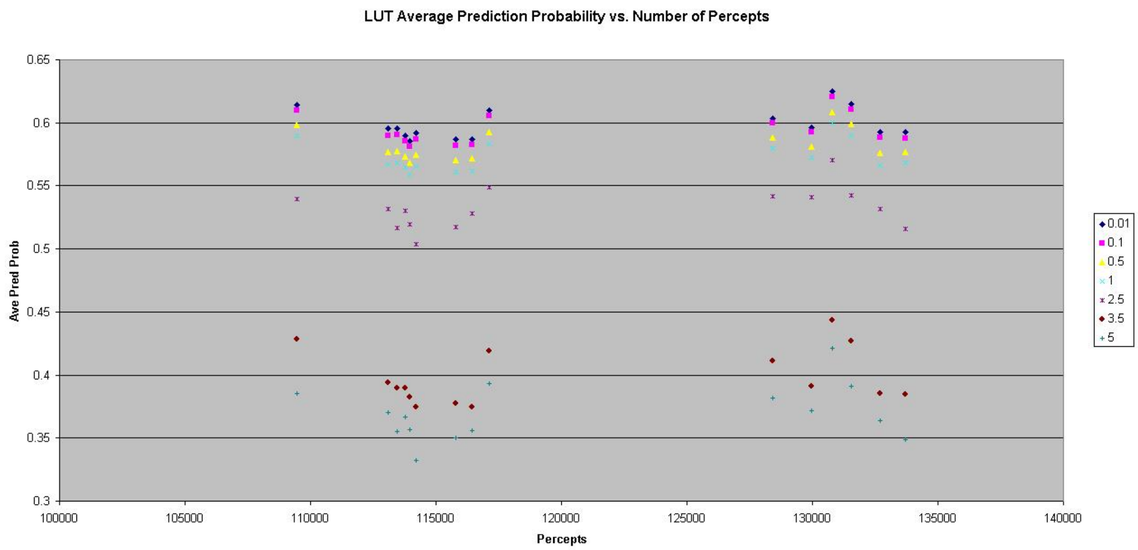


Figure 14. Average Prediction Probability versus Number of Percepts

3. Statistical Analysis

As with the Variable Order Markov Model, a multiple linear regression consisting of a plot of average probability predicted by the model, leverage plots of time threshold and number of percepts versus average prediction probability, a summary of fit of the model, an analysis of variance, parameter estimates, and an effects test of time threshold and number of percepts provide an in-depth view of the impact parameters on the Look-Up Table Model.

Figure 15, pictured below, displays the actual average prediction probability data points against the average prediction probability predicted by the model. The plot factors in both the time threshold and the number of percepts in predicting the average prediction probability. This model is constructed from parameter estimates, listed in Table 5, that make up the solid red line. This is compared to the dashed blue line which represents the overall mean of all the data in the model to determine which is a better fit of the data. The model also includes two dashed red lines indicating a 95 percent confidence interval. Since the confidence interval intersects the mean line, it indicates that there is at least one significant parameter influencing the model and the model is significant at the 5 percent level.

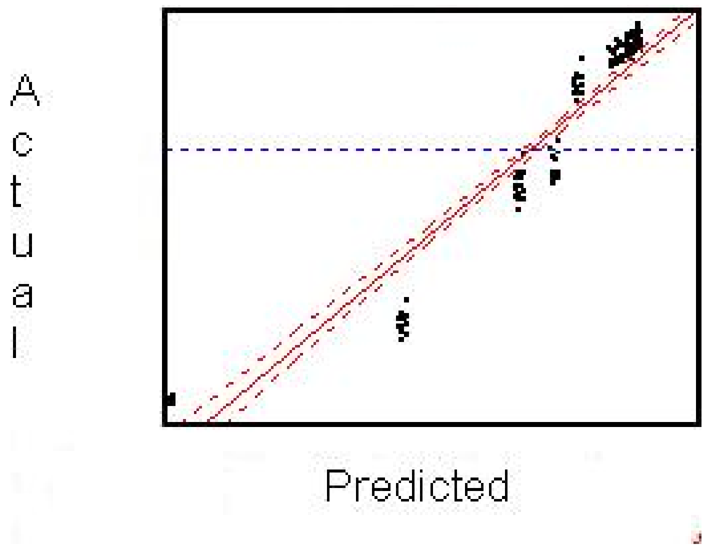


Figure 15. Actual versus Predicted Average Prediction Probability

Term	Estimate	Standard Error
Intercept	0.51996792	0.06704748
Time Threshold	-0.0307572	0.00079001
Number of Percepts	0.00000043	0.00000055

Table 5. Parameter Estimates for Line of Best Fit of Model

Both Figures 16 and 17 are leverage plots constructed in the same way as Figure 15, but each only takes into account one parameter at a time. Each displays the same line of best fit, overall model mean, and confidence intervals as the whole model. In Figure 16, the data points clearly conform to the line of best fit model much more than the overall mean model. Also, the confidence interval shows that time threshold is a significant parameter contributing to average prediction probability at the five percent level. The confidence intervals of the number of percepts leverage plot, on the other hand, indicate that the number of percepts does not have an impact on average prediction probability at the five percent level because the confidence curve does not cross the model's mean line.

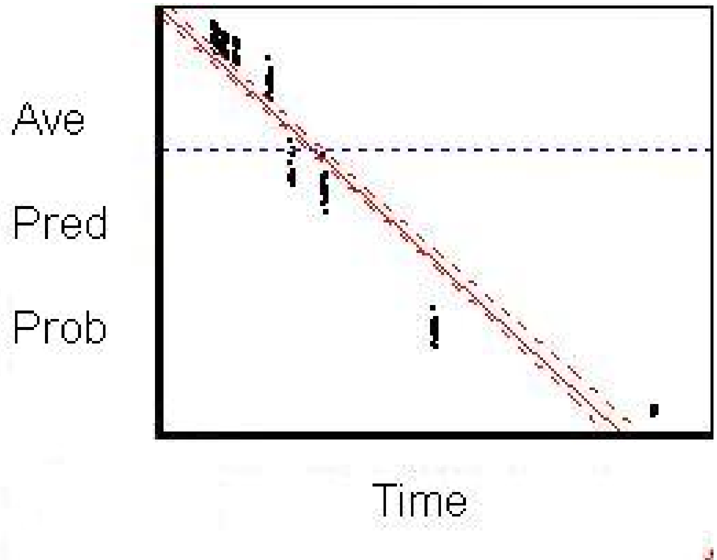


Figure 16. Average Prediction Probability versus Time

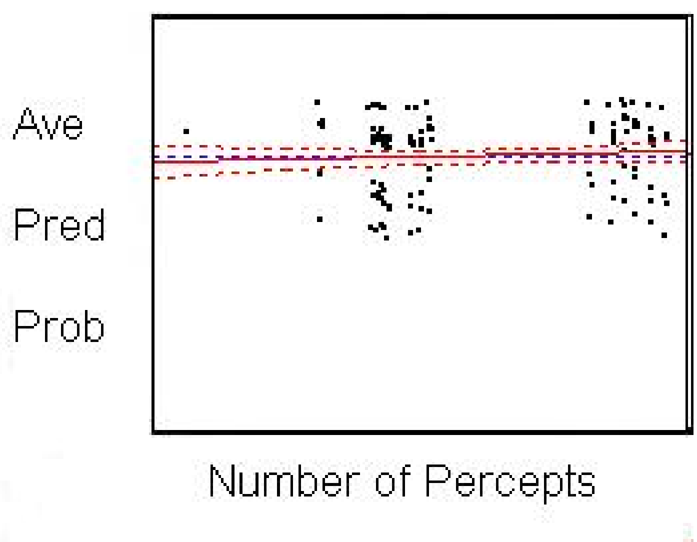


Figure 17. Average Prediction Probability versus Number of Percepts

RSquare	0.91489863
RSquare Adjusted	0.91369152
Root Mean Square Error	0.05769179
Mean of Response	0.43573312
Observations	144

Table 6. Summary of Fit Data for the Model

Source	**Degrees of Freedom**	**Sum of Squares**	**Mean Square**	**F-ratio**	**Probability > F**
Model	2	5.04525971	2.52262986	757.923819	< 0.0001
Error	296	0.46929625	0.00332834		
Total	298	5.51455597			

Table 7. Analysis of Variance for Look-Up Table Model

The multiple linear regression also produces a summary of fit, seen in Table 6, and an analysis of variance, seen in Table 7. Both summaries support the hypothesis that the model parameters, time threshold and number of percepts, have an impact on average prediction probability. In the summary of fit, the key factor is the RSquare term. Table 6 indicates that over 91 percent of the variation in the model can be explained by at least one of the parameters and not random error. Additionally, the F-ratio from the analysis of variance in Table 7 indicates almost zero probability of both parameters having no impact on average prediction probability. These numbers combine to provide overwhelming evidence that at least one parameter has an impact on the model and that variance is not being caused completely by random error.

Since there is clear evidence of a parameter impacting the model, an effects test shows the individual impact of each parameter, similar to the leverage graphs. Table 8 lists the individual F-ratios for time threshold and number of percepts and the probability of obtaining a higher F-ratio solely from random error. The results in Table 8 clearly indicate that the time threshold has significance, as there is near zero probability of obtaining a higher F-ratio. Number of percepts, though, does not appear significant due to the fact that there is almost a 50 percent chance of obtaining a higher F-ratio solely from random error. Statistically, number of percepts fails to be significant at any level up to 44 percent. Therefore, almost all of the impact on variance seen in Tables 6 and 7 above come from the time threshold and not number of percepts.

Source	**Degrees of Freedom**	**Sum of Squares**	**F-ratio**	**Probability > F**
Time Threshold	1	5.04491709	1515.7447	< 0.0001
Number of Percepts	1	0.002039	0.61261809	0.43511631

Table 8. Effects Test Table for Look-Up Table Model

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSION AND RECOMMENDATIONS

A. CONCLUSION

The goal of this thesis was to address how a particular intelligent agent learns in a complex stochastic environment, and through experimentation and statistical analysis a better understanding of two learning algorithms was achieved.

In analyzing the Variable Order Markov Model, a significant positive correlation was found between both order and number of percepts and average prediction probability. An analysis of variance and an effects test yielded very high F-ratios for both parameters, indicating each having a strong impact on prediction probability. This statistical analysis of 210 trials also showed that over 58 percent of the model's variation could be attributed to these parameters rather than random error.

Other discovered aspects of how the Variable Order Markov Model learns included a diminishing returns property with respect to increasing order. Increases in maximum model order from order two through seven routinely resulted in statistically significant increases in average prediction probability. But, increases in prediction probability were miniscule after order seven and beyond. In addition, different prediction probability patterns were seen between different percept types with increasing order. The '+' and '-' percept types experienced greater growth with increasing order than the 'E' and 'A' percept types since the higher order models incorporated more sensation percepts.

The analysis of the Look-Up Table Model provides similar insight into the prediction process. In this case, statistical analysis showed a negative correlation between increasing time threshold and average prediction probability and number of percepts was found to have no correlation to average prediction probability in this particular case. But, even with no correlation between number of percepts and average prediction probability, statistical analysis yielded a model where the parameters explained over 91 percent of the variance. Also, both analysis of variance and an effects test displayed high F-ratios which further illustrated the large impact that time threshold has on average prediction probability.

Additionally, analysis of the Look-Up Table model demonstrates a difference in average prediction probability trends between different percept types. The '-' type percepts experienced a sharper and much larger drop in average prediction probability with increasing time threshold than '+,' 'E,' and 'A' percepts because the low time threshold prediction depends heavily on sensations, due to the recency threshold limiting the total number of percepts.

A comparison of these two learning algorithms shows contrasting styles of prediction. One, the Variable Order Markov Model, produces better prediction probability by taking into account more percepts at higher orders, while the Look-Up Table Model performs better with lower parameters. In general, the Variable Order Markov Model was found to have better success in prediction with a higher mean of the average prediction probabilities taken across all orders, though the Look-Up Table Model had better '-' percept performance since it depended so heavily on sensation. Thus, despite the superior performance of the Variable Order Markov Model, the Look-Up Table Model still has some benefit and can be used in prediction. The statistical analysis gave generally better performance numbers to the Look-Up Table Model, but this was due to more trials and more differing parameters in the trials of the Variable Order Markov Model. Thus, both algorithms were shown to perform very well as they predicted actions in a stochastic environment an average of approximately 60 percent of the time with peak performance parameters.

B. RECOMMENDATIONS

The following section briefly describes areas deserving further study.

1. Improving the Learning Algorithms

One research question going unanswered due to time constraints was how to improve the algorithm to improve or accelerate the learning process. This area provides ample opportunity for further research. The Variable Order Markov Model was found to be a superior model overall, but the Look-Up Table Model was better at certain aspects, such as ‘-‘ percept prediction. Therefore, further research could be conducted to attempt to combine aspects from both algorithms and improve overall learning quality.

2. Application of Learning Algorithms

With a detailed analysis of each algorithm completed in the environment in which it was built, further research could take each algorithm outside its role-playing game environment and apply them to real-world scenarios. These learning algorithms could be applied in a military setting, similar to the work done by Dietmar Kunde and Christian Darken in their study, “*A Mental Simulation-Based Decision-Making Architecture Applied to Ground Combat*.” [6]

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Russell, S., Norvig, P. *Artificial Intelligence A Modern Approach*, Prentice Hall, 1995.

[2] Kasabov, Nikola, *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*, MIT Press, 1996.

[3] Sutton, Richard S., Barto, Andrew G. *Reinforcement Learning: An Introduction*, [online] http://www.cs.ualberta.ca/%7Esutton/book/ebook/the-book.html last accessed 19 May 2007.

[4] Begleiter, Ron et al., *On Prediction Using Variable Order Markov Models*, [online] http://www.jair.org/media/1491/live-1491-2335-jair.pdf last accessed 21 May 2007.

[5] Holmes, Michael, Isbell, Charles Lee, *Schema Learning: Experience-Based Construction of Predictive Action Models*, [online] http://books.nips.cc/papers/files/nips17/NIPS2004_0629.pdf last accessed 20 May 2007.

[6] Kunde, Dietmar, Darken, Christian, *A Mental Simulation-Based Decision-Making Architecture Applied to Ground Combat*, [online] http://www.nps.navy.mil/cs/cjdarken/06-BRIMS-013.pdf last accessed 20 May 2007.

[7] De Veaux, Richard et al., *Stats Data and Models*, Pearson Education, Inc, 2005.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California