Performance of a Single Action Partially Observable Markov Decision Process in a Recognition Task

Brett L. Moore Todd M. Quasny Larry D. Pyeatt Eric D. Sinzinger

Department of Computer Science
Texas Tech University
Lubbock, Texas 79409

{moore,quasny,pyeatt,sinzinge}@cs.ttu.edu

Keywords: Probabilistic Reasoning, POMDP, HMM, Pattern Recognition

Abstract

Partially Observable Markov Decision Processes (POMDPs) have been applied extensively to planning in environments where knowledge of an underlying process is confounded by unknown factors[3, 4, 7]. By applying the POMDP architecture to basic recognition tasks, we introduce a novel pattern recognizer that operates under partially observable conditions. This Single Action Partially Observable Markov Decision Process (SA-POMDP) is then compared to a well-known pattern recognizer, the Hidden Markov Model (HMM). Our results indicate that the SA-POMDP's performance surpasses that of the HMM in simple recognition tasks and exhibits a unique resistance to noisy inputs during the recognition process.

1 Introduction

Partially Observable Markov Decision Processes (POMDPs) have shown exciting potential for planning, despite having an intractable exact solution for sequential decision tasks[5]. The objective of this study was to apply the POMDP framework to a basic recognition task. Using a standard Hidden Markov Model (HMM) as a control for comparison, we developed the Single Action Partially Observable Markov Decision Process (SA-POMDP). The SA-POMDP was anticipated to perform equivalently with the HMM while having an advantage of mathematical simplicity.

2 Mathematical Background

Before presenting the equations for constructing an SA-POMDP, we will give a brief overview of the mathematics behind POMDPs. A POMDP is formally defined[6] as:

- a finite set of states $\mathcal{S} = \{s_0, s_1, s_2, \dots, s_{|\mathcal{S}|-1}\}$,
- a finite set of actions $\mathcal{A} = \{a_0, a_1, a_2, \dots, a_{|\mathcal{A}|-1}\}$,
- a set of actions $A(s) \subseteq \mathcal{A}$ for each state $s \in \mathcal{S}$ that can be executed in that state,
- a set of transition probabilities $\Pr(s'|s, a)\,\forall s, s' \in \mathcal{S}, a \in A(s)$,
- a set of observations $\mathcal{Z} = \{z_0, z_1, z_2, \dots, z_{|\mathcal{Z}|-1}\}$,
- the observation probabilities $\Pr(z|s', a)\,\forall z \in \mathcal{Z}, s' \in \mathcal{S}, a \in A(s)$, and
- a set of immediate rewards $r^a(s)\,\forall a \in A(s), s \in \mathcal{S}$ that are available after taking any legal action from any state.

The model given above assumes partial observability, i.e., there may not be a one-to-one mapping from observations to states, so it may be impossible to determine the current state based on observations. Instead of maintaining the current state, a probability distribution over $\mathcal{S}$ called the *belief state* is maintained. The set of all possible belief states forms the *belief space*.

Numerous versions of Bayes' conditional probability equation have been used to update the value of a belief state. The version we have adopted [6] is as follows:

$$b_z^a(s') = \frac{\Pr(z|s', a) \sum_{s \in \mathcal{S}} \Pr(s'|s, a) b(s)}{\sum_{s' \in \mathcal{S}} \left[\Pr(z|s', a) \sum_{s \in \mathcal{S}} \Pr(s'|s, a) b(s) \right]}. \tag{1}$$

For a recognition task, there is no action choice to be made at each time step. The goal is not to choose the optimal action, but simply to track the path through belief space. For purposes of recognition, the POMDP model can be simplified to have only one action available from each

Belief_State $\leftarrow$ Starting State
$\forall z \in \mathcal{Z}$
    $\forall s' \in \mathcal{S}$
        $b_z(s') = \frac{\Pr(z|s') \sum_{s \in \mathcal{S}} \Pr(s'|s) b(s)}{\sum_{s' \in \mathcal{S}} \left[\Pr(z|s') \sum_{s \in \mathcal{S}} \Pr(s'|s) b(s) \right]}$
    End
    Belief_Path$_z$ $= \max_{s'}(b)$
End

Figure 1: SA-POMDP Algorithm

state. Because we have placed the single action restriction on our POMDP, Equation 1 reduces to the following form:

$$b_z(s') = \frac{\Pr(z|s') \displaystyle\sum_{s \in \mathcal{S}} \Pr(s'|s) b(s)}{\displaystyle\sum_{s' \in \mathcal{S}} \left[\Pr(z|s') \sum_{s \in \mathcal{S}} \Pr(s'|s) b(s) \right]}. \tag{2}$$

Applying this equation, iterated over all possible states, gives us the *belief path* that the SA-POMDP followed during its recognition task. A general algorithm for the SA-POMDP is given in Figure 1.

3 Architecture

In many respects, the Single Action Partially Observable Markov Decision Process resembles the Hidden Markov Model. Both schemes comprise a finite set of states, a matrix dictating the state transition probabilities, and a matrix designating the observational probability densities. Furthermore, the classical POMDP notions of action and reward vanish when the action set is reduced to a single action. In fact, the remaining action may simply be interpreted as “change state” at each time step, similar in principle to the HMM. However, the SA-POMDP has no need of the HMMs initial state probability matrix. The SA-POMDP is assumed to begin in a designated start state with 100% certainty.

Each SA-POMDP was composed of $N + 3$ states, where N (at most) equaled the character length of the associated word. Each unique letter of the word was assigned a distinct state; hence,

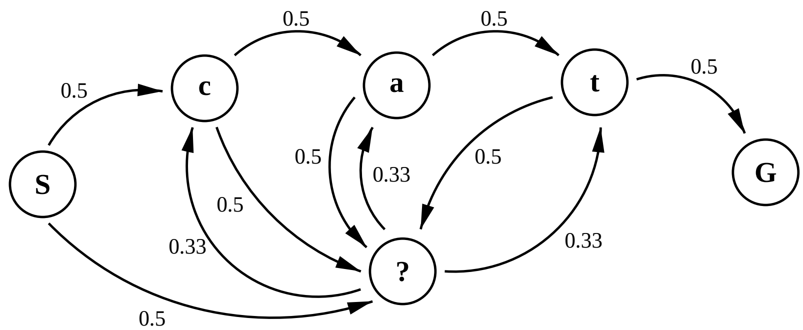

Figure 2: State transition diagram for an example SA-POMDP. Not shown are the zero-likelihood transitions.

the word model may contain recurrencies. (The ASCII value of the character was called the *state value*). The model for each word included start, goal, and nonsense states.

The nonsense state is perhaps the most distinctive feature of the SA-POMDP. Entrance into this state indicates that the belief state is of sufficient uncertainty that the model is effectively lost. Figure 2 illustrates a representative SA-POMDP model for the word *cat*. As Figure 2 implies, the state transition likelihoods are uniformly distributed among valid state changes. Furthermore, a transition to the nonsense state may never be discounted due to the system's limited observability. As an example, given the *cat* model, a transition from state *c* may goto state *a* or to the nonsense state with equal probability. Transition probabilities to states representing letters not found in the word *cat* must be zero. A transition from state *c* to state *k* should never occur in the *cat* model.

The continuous state observation densities are defined by a unimodal Gaussian centered at the *state value* with variance of 0.01. This approximation captures the element of stochasticity inherent in a partially observable system. This explicit determination of transition and observation probabilities is used in lieu of a more conventional training technique, such as gradient descent.

Our implementation consists of a two-layer architecture, as shown in Figure 3. The SA-POMDP layer is composed of an order-independent collection of SA-POMDPs, one model for each word in the vocabulary. In a manner similar to an isolated word HMM recognizer, a coordinating layer is placed above the model layer to administrate sequential dispatching of the queries to the SA-POMDPs to obtain the SA-POMDP's belief path encountered during the classification attempt. Paths correlating highly to the unknown word are rated, and the recognizer returns the most likely match.

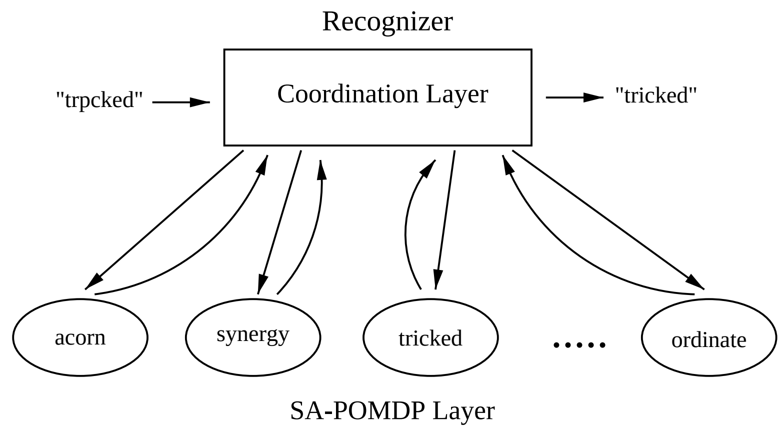

Figure 3: A block diagram of a two-layer SA-POMDP-based pattern recognizer. An unknown, possibly distorted, input is presented to the recognizer for classification, and the coordination layer polls each SA-POMDP by dispatching the unknown and accepting a corresponding belief measure. The coordinator then determines the most likely match and identifies the appropriate model as the matched class

Note that this architecture should behave similarly to an HMM-based isolated word classification system. If a word is not in the HMM vocabulary, the word with the highest degree of match is returned as the match candidate. Typically, this likelihood of match will far exceed some minimum threshold, and the recognizer is implemented to reject such spurious classification attempts as failures.

4 Methods

Although POMDPs have been used for control tasks, they have not been applied to pattern recognition. To evaluate the hypothesis that POMDP methods can be used to reliably classify patterns using a constrained knowledge of system state, we applied the SA-POMDP, to a text-based word recognition task.

Recognizing a small vocabulary of ASCII strings was expected to be a challenging but well-defined introductory task. To develop a classification strategy for the members of the vocabulary, we assigned each letter of the modeled word to a unique state. These discretely-valued states were assigned a *state value* (the ASCII value of the associated letter). Although the SA-POMDP places no such restrictions on the input, the words were composed solely of lowercase letters to facilitate the execution of this experiment. Candidate words were randomly selected from a standard

chirp	humanity	threat	tricked	exclusion
compass	latrine	special	sobering	adored
groves	loiter	sprawling	synergy	downcast
ordinate	proud	acorn	brisk	modularizes

Table 1: The vocabulary used in our experiment

dictionary file; twenty words varying in length from five to eleven letters were then selected for vocabulary inclusion. Table 1 enumerates our vocabulary.

The MATLAB™ development environment was used to construct the SA-POMDP architecture. The vocabulary was presented to the recognizer for classification under 51 differing levels of additive noise. The noise signal was sampled from a Gaussian centered at the *state value*. The amplitude of the additive noise ranged from 0.0 to 5.0 in discrete steps of 0.1.

Testing over the entire noise scale comprised a single pass, and the success rate, measured in percent, was then calculated for a dataset consisting of twenty passes. A global measure of recognizer accuracy was obtained by computing the mean and standard deviation of the obtained success rates for an entire vocabulary.

Twenty HMMs were constructed; one for each member of the vocabulary[8]. Each letter of the word was considered as one element of the associated observation sequence. Although hidden state count varied with the length of the observation sequence, observations were randomly assigned to state. Each model was iteratively trained on one thousand instances of the desired observation sequence. To facilitate recognition under noisy conditions, each element of an observation instance was perturbed with Gaussian noise centered at the *state value* of the character with a variance of 0.1. Consequently, each state's continuous observation densities were modeled with a unimodal Gaussian. The HMM architecture was implemented in a similar manner to the SA-POMDP architecture: a MATLAB™ implementation was evaluated over the noise scale, and a mean success rate was computed.

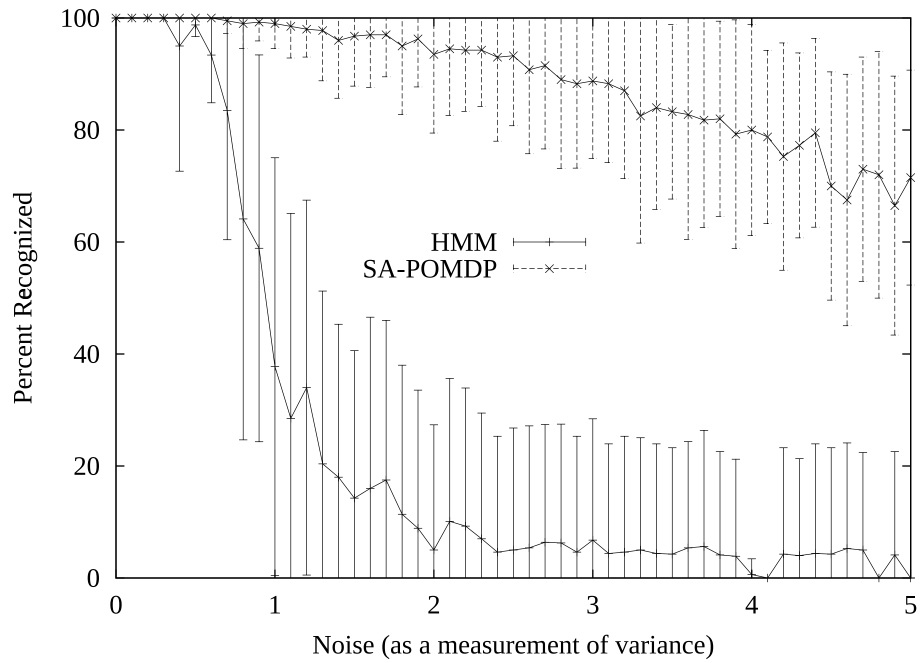

Figure 4: HMM versus SA-POMDP (mean $\pm$ σ success rate)

5 Results

The SA-POMDP recognizer exhibited a greater than expected resistance to all levels of the normally distributed noise signal. Perfect classification was maintained up to $\sigma^2 = 0.8$. Beyond this point, the system exhibited a slow logarithmic decline in average performance, reaching a minimum recognition level of 67% at extreme noise levels.

The HMM recognizer proved unable to maintain the SA-POMDP's level of accuracy. The HMM no longer performed perfectly at $\sigma^2 > 0.4$. Noise levels greater than $\sigma^2 = 0.5$ marked a precipitous decline in accuracy such that the success rate fell below forty percent at $\sigma^2 = 1.0$. The success rate continued to decline nearly exponentially to an effective rate of approximately 5% at $\sigma^2 = 2.0$.

Although the SA-POMDP maintained a perfect recognition for twice the duration of the HMM, this metric fails to address the more impressive aspects of Figure 4. As indicated in the plot, HMM performance experienced such a catastrophic falloff that a 59% difference in mean success rates appeared at $\sigma^2 = 1.0$. Further inspection of Figure 4 emphasizes the resilience of the SA-POMDP. At the maximum tested noise level, SA-POMDP demonstrated a mean success rate of 67% versus the HMMs near zero success rate.

6 Discussion

The SA-POMDP proved surprisingly adept at identifying words in partially observable environments. To express the observed aptitude in another manner, SA-POMDPs demonstrated the ability to overcome a noise level equivalent to a range of ten state values. As an example, an actual state value of ‘q’ perturbed by Gaussian noise with variance of 5.0 is generally observed somewhere in the set ranging from the state value of ‘l’ to the state value of ‘v’.

The degree of observed disparity in noise resistance between HMMs and SA-POMDPs was unanticipated. Without exception, SA-POMDPs surpassed HMMs in accuracy of recognition at comparative levels of input distortion. In tasks recognizing short words, the noise immunity of SA-POMDPs was observed to be ten times that of the HMM. Immunity declined to roughly seven times the HMM levels for tested words exceeding eight characters.

Such large differences in performance immediately call to question the suitability of the HMM as a control in this study; however, the preliminary nature of this work allows little precedent for a measure of SA-POMDP’s performance. The Hidden Markov Model, with its firm mathematical background and extensive research base, has demonstrated proficiency in difficult recognition tasks, such as automated speech recognition[1]. Thus, an HMM seemed a logical choice for comparison. Despite the relatively poor performance of this HMM, it is expected that other HMM implementations could prove more challenging to the SA-POMDP.

7 Limitations

One principle limitation of this study is the lack of SA-POMDP training. Our initial models utilized transition matrices consisting of hand assigned transition probabilities. In addition, the observation density was assumed to be a single-mixture Gaussian with an arbitrarily small variance. It is anticipated that an iterative parameter re-estimation technique, such as Baum-Welch as applied in the HMM context, will improve SA-POMDP performance. Furthermore, the effect of modeling observational densities with multimodal Gaussian mixtures has not been studied.

Since current SA-POMDP training methods include only a determination of stochastically-biased state transitions, model estimation requires little computational time. However, our recognizer acted in a naive brute-force fashion and required a great deal of time to identify a word. This problem is not unique to SA-POMDP; Hidden Markov Models similarly suffer from the curse of dimensionality, yet the use of the Baum-Welch forward and backward variables[8] make the problem tractable. As we improve and expand our SA-POMDP architecture, we expect better recognition performance and more speed.

Although the HMM demonstrated satisfactory performance as a comparator, accuracy of the control may be improved by employing more sophisticated models. Such improvements may include multi-modal Gaussian observation densities, more robust state-observation assignments (for example, k-means segmental clustering) and more thorough training sessions.

8 Future Work and Conclusions

The SA-POMDP has been demonstrated to provide an effective pattern recognition paradigm. It has the ability to map noisy continuous inputs to discrete outputs. This makes it ideally suited to operate on low SNR data. However, there are still open questions about underlying statistics, training and application.

One factor is the effect of input vector characteristics. It is unclear how the dimensionality, variance levels, and class separation of the input vector will impact recognition. Additionally, there is the issue of SA-POMDP training unaddressed. Presumably, a modified version of the Baum-Welch EM algorithm[2] may be used to tune the SA-POMDP parameters for improved results. Furthermore, architecture enhancements, such as support for multi-modal Gaussian observation densities, is under consideration.

In effort to explore the potential field of application, we intend studies on a number of pattern recognition problems. Specifically, SA-POMDP is expected to provide robust results for processes constrained by incomplete state perception. One example is the implementation of Single Action Partially Observable Markov Decision Process to a facial recognition task in a manner similar to Samaria[9]. Another example is the problem of speech recognition. The performance of SA-POMDP in this initial study indicates that it could be a novel approach to pattern recognition, with many possible applications.

9 Acknowledgments

This work was supported in part by National Science Foundation CRCD Grant 9980296. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation herein. The authors wish to thank NSF for their kind support.

References

[1] Claudio Bechetti and Lucio Prina Ricotti. *Speech Recognition: Theory and C++ Implementation*. John Wiley and Sons, Chichester,West Sussex, England, 1999. ISBN 0-471-97730-6.

[2] Jeff A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report TR-97-021, International Computer Science Institute, Berkeley CA, April 1998.

[3] Anthony R. Cassandra. A survey of POMDP applications. In Michael Littmann, editor, *Working Notes: AAAI Fall Symposium on Planning with Partially Observable Markov Decision Processes*, pages 17–24. AAAI, October 1998.

[4] Héctor Geffner and Blai Bonet. High-level planning and control with incomplete information using POMDPs. In *AIPS-98 Workshop on Integrating Planning, Scheduling and Execution in Dynamic and Uncertain Environments*, 1998.

[5] Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, August 1987.

[6] Larry D. Pyeatt. *Integration of Partially Observable Markov Decision Processes and Reinforcement Learning for Simulated Robot Navigation*. PhD dissertation, Colorado State University, Computer Science Department, July 1999.

[7] Larry D. Pyeatt and Adele E. Howe. Integrating POMDP and reinforcement learning for a two layer simulated robot architecture. In *Third International Conference on Autonomous Agents*, Seattle, Washington, May 1999.

[8] Lawrence Rabiner and Bilng-Hwang Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–15, January 1986.

[9] Ferdinando Samaria. Face segmentation for identification using hidden markov models. In *Proceedings of the 4th British Machine Vision Conference*, Guildford, September 1993. Springer-Verlag.