Estimating the Reliability of MDP Policies: A Confidence Interval Approach

Joel R. Tetreault
University of Pittsburgh
LRDC
Pittsburgh PA, 15260, USA
tetreaul@pitt.edu

Dan Bohus
Carnegie Mellon University
Dept. of Computer Science
Pittsburgh, PA, 15213, USA
dbohus@cs.cmu.edu

Diane J. Litman
University of Pittsburgh
Dept. of Computer Science
LRDC
Pittsburgh PA, 15260, USA
litman@cs.pitt.edu

Abstract

Past approaches for using reinforcement learning to derive dialog control policies have assumed that there was enough collected data to derive a reliable policy. In this paper we present a methodology for numerically constructing confidence intervals for the expected cumulative reward for a learned policy. These intervals are used to (1) better assess the reliability of the expected cumulative reward, and (2) perform a refined comparison between policies derived from different Markov Decision Processes (MDP) models. We applied this methodology to a prior experiment where the goal was to select the best features to include in the MDP state-space. Our results show that while some of the policies developed in the prior work exhibited very large confidence intervals, the policy developed from the best feature set had a much smaller confidence interval and thus showed very high reliability.

1 Introduction

NLP researchers frequently have to deal with issues of data sparsity. Whether the task is machine translation or named-entity recognition, the amount of data one has to train or test with can greatly impact the reliability and robustness of one's models, results and conclusions.

One research area that is particularly sensitive to the data sparsity issue is machine learning, specifically in using Reinforcement Learning (RL) to learn the optimal action for a dialogue system to make given any user state. Typically this involves learning from previously collected data or interacting in real-time with real users or user simulators. One of the biggest advantages to this machine learning approach is that it can be used to generate optimal policies for every possible state. However, this method requires a thorough exploration of the state-space to make reliable conclusions on what the best actions are. States that are infrequently visited in the training set could be assigned sub-optimal actions, and therefore the resulting dialogue manager may not provide the best interaction for the user.

In this work, we present an approach for estimating the reliability of a policy derived from collected training data. The key idea is to take into account the uncertainty in the model parameters (MDP transition probabilities), and use that information to numerically construct a confidence interval for the expected cumulative reward for the learned policy. This confidence interval approach allows us to: (1) better assess the reliability of the expected cumulative reward for a given policy, and (2) perform a refined comparison between policies derived from different MDP models.

We apply the proposed approach to our previous work (Tetreault and Litman, 2006) in using RL to improve a spoken dialogue tutoring system. In that work, a dataset of 100 dialogues was used to develop a methodology for selecting which user state features should be included in the MDP state-space. But are 100 dialogues enough to generate reliable policies? In this paper we apply our confidence in-

Proceedings of NAACL HLT 2007, pages 276–283,
Rochester, NY, April 2007.

terval approach to the same dataset in an effort to investigate how reliable our previous conclusions are, given the amount of available training data.

In the following section, we discuss the prior work and its data sparsity issue. In section 3, we describe in detail our confidence interval methodology. In section 4, we show how this methodology works by applying it to the prior work. In sections 5 and 6, we present our conclusions and future work.

2 Previous Work

Past research into using RL to improve spoken dialogue systems has commonly used Markov Decision Processes (MDP's) (Sutton and Barto, 1998) to model a dialogue (such as (Levin and Pieraccini, 1997) and (Singh et al., 1999)).

A MDP is defined by a set of states $\{s_i\}_{i=1..n}$, a set of actions $\{a_k\}_{k=1..p}$, and a set of transition probabilities which reflect the dynamics of the environment $\{p(s_i|s_j, a_k)\}_{i,j=1..n}^{k=1..p}$: if the model is at time t in state s_j and takes action a_k, then it will transition to state s_i with probability $p(s_i|s_j, a_k)$. Additionally, an expected reward $r(s_i, s_j, a_k)$ is defined for each transition. Once these model parameters are known, a simple dynamic programming approach can be used to learn the optimal control policy π^*, i.e. the set of actions the model should take at each state, to maximize its expected cumulative reward.

The dialog control problem can be naturally cast in this formalism: the states $\{s_i\}_{i=1..n}$ in the MDP correspond to the dialog states (or an abstraction thereof), the actions $\{a_k\}_{k=1..p}$ correspond to the particular actions the dialog manager might take, and the rewards $r(s_i, s_j, a_k)$ are defined to reflect a particular dialog performance metric. Once the MDP structure has been defined, the model parameters $\{p(s_i|s_j, a_k)\}_{i,j=1..n}^{k=1..p}$ are estimated from a corpus of dialogs (either real or simulated), and, based on them, the policy which maximizes the expected cumulative reward is computed.

While most work in this area has focused on developing the best policy (such as (Walker, 2000), (Henderson et al., 2005)), there has been relatively little work done with respect to selecting the best features to include in the MDP state-space. For instance, Singh et al. (1999) showed that dialogue length was a useful state feature and Frampton and Lemon (2005) showed that the user's last dialogue act was also useful. In our previous work, we compare the worth of several features. In addition, Paek and Chickering's (2005) work showed how a state-space can be reduced by only selecting features that are relevant to maximizing the reward function.

The motivation for this line of research is that if one can properly select the most informative features, one develops better policies, and thus a better dialogue system. In the following sections we summarize our past data, approach, results, and issue with policy reliability.

2.1 MDP Structure

For this study, we used an annotated corpus of human-computer spoken dialogue tutoring sessions. The fixed-policy corpus contains data collected from 20 students interacting with the system for five problems (for a total of 100 dialogues of roughly 50 turns each). The corpus was annotated with 5 state features (Table 1). It should be noted that two of the features, Certainty and Frustration, were manually annotated while the other three were done automatically. All features are binary except for Certainty which has three values.

State	Values
Correctness	Student is correct or incorrect in the current turn
Certainty	Student is certain, neutral or uncertain in the current turn
Concept Repetition	A particular concept is either new or repeated
Frustration	Student is frustrated or not in the current turn
Percent Correct	Student answers over 66% of questions correctly in dialogue so far, or less

Table 1: State Features in Tutoring Corpus

For the action set $\{a_k\}_{k=1..p}$, we looked at what type of question the system could ask the student given the previous state. There are a total of four possible actions: ask a short answer question (one that requires a simple one word response), a complex answer question (one that requires a longer, deeper response), ask both a simple and complex question in the same turn, or do not ask a question at all (give a hint). The reward function r was the learning gain of each student based on a pair of tests before and after the entire session of 5 dialogues. The 20 students were split into two groups (high and low learners) based on their learning gain, so 10 students and their respective five dialogues were given a positive reward of +100, while the remainder were assigned a negative reward of -100. The rewards were assigned in the final dialogue state, a common approach when applying RL in spoken dialogue systems.

2.2 Approach and Results

To investigate the usefulness of different features, we took the following approach. We started with two baseline MDPs. The first model (Baseline 1) used only the Correctness feature in the state-space. The second model (Baseline 2) included both the Correctness and Certainty features. Next we constructed 3 new models by adding each of the remaining three features (Frustration, Percent Correct and Concept Repetition) to the Baseline 2 model.

We defined three metrics to compare the policies derived from these MDPs: (1) Diff's: the number of states whose policy differs from the Baseline 2 policy, (2) Percent Policy change (P.C.): the weighted amount of change between the two policies (100% indicates total change), and (3) Expected Cumulative Reward (or ECR) which is the average reward one would expect in that MDP when in the state-space.

The intuition is that if a new feature were relevant, the corresponding model would lead to a different policy and a better expected cumulative reward (when compared to the baseline models). Conversely, if the features were not useful, one would expect that the new policies would look similar (specifically, the Diff's count and % Policy Change would be low) or produce similar expected cumulative rewards to the original baseline policy.

The results of this analysis are shown in Table 2 [1] The Diff's and Policy Change metrics are undefined for the two baselines since we only use these two metrics to compare the other three features to Baseline 2. All three metrics show that the best feature to add to the Baseline 2 model is Concept Repetition since it results in the most change over the Baseline 2 policy, and also the expected reward is the highest as well. For the remainder of this paper, when we refer to Concept Repetition, Frustration, or Percent Correctness, we are referring to the model that includes that feature as well as the Baseline 2 features Correctness and Certainty.

State Feature	# Diff's	% P.C.	ECR
Baseline 1	N/A	N/A	6.15
Baseline 2	N/A	N/A	31.92
B2 + Concept Repetition	10	80.2%	42.56
B2 + Frustration	8	66.4%	32.99
B2 + Percent Correctness	4	44.3%	28.50

Table 2: Feature Comparison Results

2.3 Problem with Reliability

However, the approach discussed above assumes that given the size of the data set, the ECR and policies are reliable. If the MDP model were very fragile, that is the policy and expected cumulative reward were very sensitive to the quality of the transition probability estimates, then the metrics could reveal quite different rankings. Previously, we used a qualitative approach of tracking how the worth of each state (V-value) changed over time. The V-values indicate how much reward one would expect from starting in that state to get to a final state. We hypothesized that if the V-values stabilized as data increased, then the learned policy would be more reliable.

So is this V-value methodology adequate for assessing if there is enough data to determine a stable policy, and also for assessing if one model is better than another? Since our approach for state-space selection is based on comparing a new policy with a baseline policy, having a stable policy is extremely important since instability could lead to different conclusions. For example, in one comparison, a new policy could differ with the baseline in 8 out of 10 states. But if the MDP were unstable, adding just a little more data could result in a difference of only 4 out of 10 states. Is there an approach that can categorize whether given a certain data size,

[1] Please note that to due to refinements in code, there is a slight difference between the ECR's reported in this work and the ECR's reported in the previous work, for the three features added to Baseline 2. These changes did not alter the rankings of these models, or the conclusions of the previous work.

that the expected cumulative reward (and thus the policy) is reliable? In the next section we present a new methodology for numerically constructing confidence intervals for these value function estimates. Then, in the following section, we reevaluate our prior work with this methodology and discuss the results.

3 Confidence Interval Methodology

3.1 Policy Evaluation with Confidence Intervals

The starting point for the proposed methodology is the observation that for each state s_j and action a_k in the MDP, the set of transition probabilities $\{p(s_i|s_j, a_k)\}_{i=1..n}$ are modeled as multinomial distributions that are estimated from the transition counts in the training data:

$$\hat{p}(s_i|s_j, a_k) = \frac{c(s_i, s_j, a_k)}{\sum_{i=1}^{n} c(s_i, s_j, a_k)} \quad (1)$$

where n is the number of states in the model, and $c(s_i, s_j, a_k)$ is the number of times the system was in state s_j, took action a_k, and transitioned to state s_i in the training data.

It is important to note that these parameters are just estimates. The reliability of these estimates clearly depends on the amount of training data, more specifically on the transition counts $c(s_i, s_j, a_k)$. For instance, consider a model with 3 states and 2 actions. Say the model was in state s_1 and took action a_1 ten times. Out of these, three times the model transitioned back to state s_1, two times it transitioned to state s_2, and five times to state s_3. Then we have:

$$\hat{p}(s_i|s_1, a_1) = \langle 0.3; 0.2; 0.5 \rangle = \langle \frac{3}{10}; \frac{2}{10}; \frac{5}{10} \rangle \quad (2)$$

Additionally, let's say the same model was in state s_2 and took action a_2 1000 times. Following that action, it transitioned 300 times to state s_1, 200 times to state s_2, and 500 times to state s_3.

$$\hat{p}(s_i|s_2, a_2) = \langle 0.3; 0.2; 0.5 \rangle = \langle \frac{300}{1000}; \frac{200}{1000}; \frac{500}{1000} \rangle \quad (3)$$

While both sets of transition parameters have the same value, the second set of estimates is more reliable. The central idea of the proposed approach is to model this uncertainty in the system parameters, and use it to numerically construct confidence intervals for the value of the optimal policy.

Formally, each set of transition probabilities $\{p(s_i|s_j, a_k)\}_{i=1..n}$ is modeled as a multinomial distribution, estimated from data[2]. The uncertainty of multinomial estimates are commonly modeled by means of a Dirichlet distribution. The Dirichlet distribution is characterized by a set of parameters α_1, α_2, ..., α_n, which in this case correspond to the counts $\{c(s_i, s_j, a_k)\}_{i=1..n}$. For any given j, the likelihood of the set of multinomial transition parameters $\{p(s_i|s_j, a_k)\}_{i=1..n}$ is then given by:

$$P(\{p(s_i|s_j, a_k)\}_{i=1..n}|D) = \\ = \frac{1}{Z(D)} \prod_{i=1}^{n} p(s_i|s_j, a_k)^{\alpha_i - 1} \quad (4)$$

where $Z(D) = \frac{\prod_{i=1}^{n} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{n} \alpha_i)}$ and $\alpha_i = c(s_i, s_j, a_k)$. Note that the maximum likelihood estimates for the formula above correspond to the frequency count formula we have already described:

$$\hat{p}_{ML}(s_i|s_j, a_k) = \frac{\alpha_i}{\sum_{i=1}^{n} \alpha_i} = \frac{c(s_i, s_j, a_k)}{\sum_{i=1}^{n} c(s_i, s_j, a_k)} \quad (5)$$

To capture the uncertainty in the model parameters, we therefore simply need to store the counts of the observed transitions $c(s_i, s_j, a_k)$. Based on this model of uncertainty, we can numerically construct a confidence interval for the value of the optimal policy π^*. Instead of computing the value of the policy based on the maximum likelihood transition estimates $\hat{T}_{ML} = \{\hat{p}_M L(s_i|s_j, a_k)\}_{i,j=1..n}^{k=1..p}$, we generate a large number of transition matrices $\hat{T}_1$, $\hat{T}_1$, ... $\hat{T}_m$ by sampling from the Dirichlet distributions corresponding to the counts observed in the training data (in the experiments reported in this paper, we used $m = 1000$). We then compute the value of the optimal policy π^* in each of these models $\{V_{\pi^*}(\hat{T}_i)\}_{i=1..m}$. Finally, we numerically construct the 95% confidence interval for the value function based on the resulting value estimates: the bounds for the confidence interval are set at the lowest and highest 2.5 percentile of the resulting distribution of the values for the optimal policy $\{V_{\pi^*}(\hat{T}_i)\}_{i=1..m}$.

The algorithm is outlined below:

[2] By p we will denote the true model parameters; by $\hat{p}$ we will denote data-driven estimates for these parameters

1. compute transition counts from the training set:

$$C = \{c(s_i, s_j, a_k)\}_{i,j=1..n}^{k=1..p} \tag{6}$$

2. compute maximum likelihood estimates for transition probability matrix:

$$\hat{T}_{ML} = \{\hat{p}_{ML}(s_i|s_j, a_k)\}_{i,j=1..n}^{k=1..p} \tag{7}$$

3. use dynamic programming to compute the optimal policy π^* for model $\hat{T}_{ML}$

4. sample m transition matrices $\{\hat{T}_k\}_{k=1..m}$, using the Dirichlet distribution for each row:

$$\{\hat{p}_i(s_i|s_j, a_k)\}_{i=1..n} = \\ = Dir(\{c(s_i, s_j, a_k)\}_{i=1..n}) \tag{8}$$

5. evaluate the optimal policy π^* in each of these m models, and obtain $V_{\pi^*}(\hat{T}_i)$

6. numerically build the 95% confidence interval for V_{π^*} from these estimates.

To summarize, the central idea is to take into account the reliability of the transition probability estimates and construct a confidence interval for the expected cumulative reward for the learned policy. In the standard approach, we would compute an estimate for the expected cumulative reward, by simply using the transition probabilities derived from the training set. Note that these transition probabilities are simply estimates which are more or less accurate, depending on how much data is available. The proposed methodology does not fully trust these estimates, and asks the question: given that the real world (i.e. real transition probabilities) might actually be a bit different than we think it is, how well can we expect the learned policy to perform? Note that the confidence interval we construct, and therefore the conclusions we draw, are with respect to the policy learned from the current estimates, i.e. from the current training set. If more data becomes available, a different optimal policy might emerge, about which we cannot say much.

3.2 Related Work

Given the stochastic nature of the models, confidence intervals are often used to estimate the reliability of results in machine learning experiments, e.g. (Rivals and Personnaz, 2002), (Schapire, 2002) and (Dumais et al., 1998). In this work we use a confidence interval methodology in the context of MDPs. The idea of modeling the uncertainty of the transition probability estimates using Dirichlet models also appears in (Jaulmes et al., 2005). In that work, the authors used the uncertainty in model parameters to develop active learning strategies for partially observable MDPs, a topic not previously addressed in the literature. In our work we rely on the same model of uncertainty for the transition matrix, but use it to derive confidence intervals for the expected cumulative reward for the learned optimal policy, in an effort to assess the reliability of this policy.

4 Results

Our previous results indicated that Concept Repetition was the best feature to add to the Baseline 2 state-space model, but also that Percent Correctness and Frustration (when added to Baseline 2) offered an improvement over the Baseline MDP's. However, these conclusions were based on a very qualitative approach for determining if a policy is reliable or not. In the following subsection, we apply our approach of confidence intervals to empirically determine if given this data set of 100 dialogues, whether the estimates of the ECR are reliable, and whether the original rankings and conclusions hold up under this refined analysis. In subsection 4.2, we provide a methodology for pinpointing when one model is better than another.

4.1 Quantitative Analysis of ECR Reliability

For our first investigation, we look at the confidence intervals of each MDP's ECR over the entire data set of 20 students (later in this section we show plots for the confidence intervals as data increases). Table 3 shows the upper and lower bounds for the ECR originally reported in Table 2. The first column shows the original, estimated ECR of the MDP and the last column is the width of the bound (the difference between the upper and lower bound).

So what conclusions can we make about the reliability of the ECR, and hence of the learned policies for the different MDP's, given this amount of training data? The confidence interval for the ECR for

State Feature	ECR	Lower Bound	Upper Bound	Width
Baseline 1	6.15	0.21	23.73	23.52
Baseline 2 (B2)	31.92	-5.31	60.48	65.79
B2 + Concept Repetition	42.56	28.37	59.29	30.92
B2 + Frustration	32.99	-4.12	61.30	65.42
B2 + Percent Correctness	28.50	-5.89	57.82	63.71

Table 3: Confidence Intervals with complete dataset

the Baseline 1 model ranges from 0.21 to 23.73. Recall that the final states are capped at +100 and -100, and are thus the maximum and minimum bounds that one can see in this experiment. These bounds tell us that, if we take into account the uncertainty in the model estimates (given the small training set size), with probability 0.95 the actual true ECR for this policy will be greater than 0.21 and smaller than 23.73. The width of this confidence interval is 23.52.

For the Baseline 2 model, the bounds are much wider: from -5.31 to 60.48, for a total width of 65.79. While the ECR estimate is 31.92 (which is seemingly larger than 6.15 for the Baseline 1 model), the wide confidence interval tells us that this estimate is not very reliable. It is possible that the policy derived from this model with this amount of data could perform poorly, and even get a negative reward. From the dialogue system designer's standpoint, a model like this is best avoided.

Of the remaining three models – Concept Repetition, Frustration, and Percent Correctness, the first one exhibits a tighter confidence interval, indicating that the estimated expected cumulative reward (42.56) is fairly reliable: with 95% probability of being between 28.37 and 59.29. The ECR for the other two models (Frustration and Percent Correctness) again shows a wide confidence interval once we take into account the uncertainty in the model parameters.

These results shed more light on the shortcomings of the ECR metric used to evaluate the models in prior work. This estimate does not take into account the uncertainty of the model parameters. For example, a model can have an optimal policy with a very high ECR value, but have very wide confidence bounds reaching even into negative rewards. On the other hand, another model can have a relatively lower ECR but if its bounds are tighter (and the lower bound is not negative), one can know that that policy is less affected by poor parameter estimates stemming from data sparsity issues. Using the confidence intervals associated with the ECR gives a much more refined, quantitative estimate of the reliability of the reward, and hence of the policy derived from that data.

An extension of this result is that confidence intervals can also allow us to make refined judgments about the comparative utility of different features, the original motivation of our prior study. Basically, a model (M1) is better than another (M2) if M1's lower bound is greater than the upper bound of M2. That is, one knows that 95% of the time, the worst case situation of M1 (the lower bound) will always yield a higher reward than the best case of M2. In our data, this happens only once, with Concept Repetition being empirically better than Baseline 1, since the lower bound of Concept Repetition is 28.37 and the upper bound of Baseline 1 is 23.73. Given this situation, Concept Repetition is a useful feature which, when included in the model, leads to a better policy than simply using Correctness. We cannot draw any conclusions about the other features, since their bounds are generally quite wide. Given this amount of training data, we cannot say whether Percent Correctness and Frustration are better features than the Baseline MDP's. Although their ECR's are higher, there is too much uncertainty to definitely conclude they are better.

4.2 Pinpointing Model Cross-over

The previous analysis focused on a quantitative method of (1) determining the reliability of the MDP ECR estimate and policy, as well as (2) assessing whether one model is better than another. In this section, we present an extension to the second contribution by answering the question: given that one model is more reliable than another, is it possible to determine at which point one model's estimates become more reliable than another model's? In our

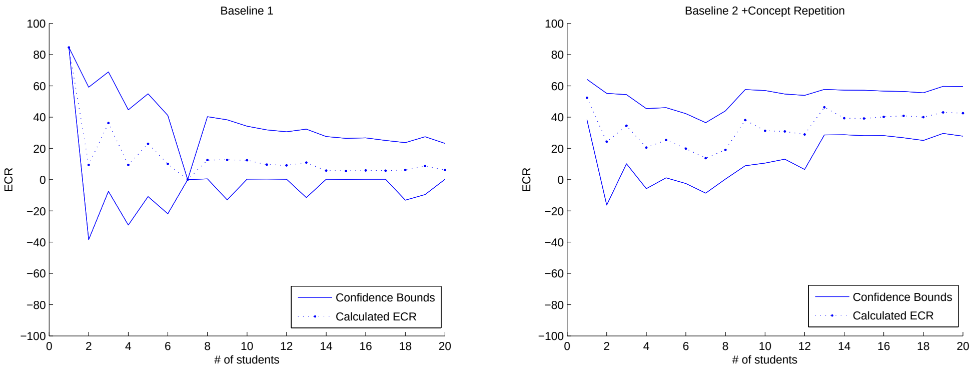

Figure 1: Confidence Interval Plots

case, we want to know at what point Concept Repetition becomes more reliable than Baseline 1. To do this, we investigate how the confidence interval changes as the amount of training data increases instead of looking at the reliability estimate at only one particular data size.

We incrementally increase the amount of training data (adding the data from one new student at a time), and calculate the corresponding optimal policy and confidence interval for the expected cumulative reward for that policy. Figure 1 shows the confidence interval plots as data is added to the MDP for the Baseline 1 and Concept Repetition MDP's. For reference, Baseline 2, Percent Correctness and Frustration plots did not exhibit the same converging behavior as these two, which is not surprising given how wide the final bounds are. For each plot, the bold lines represent the upper and lower bounds, and the dotted line represents the calculated ECR.

Analyzing the two MDP's, we find that the confidence intervals for Baseline 1 and Concept Repetition converge as more data is added, which is an expected trend. One useful result from observing the change in confidence intervals is that one can determine the point in one which one model becomes empirically better than another. Superimposing the upper and lower bounds (Figure 2) reveals that after we include the data from the first 13 students, the lower bound of Concept Repetition crosses over the upper bound of Baseline 1.

Observing this behavior is especially useful for performing model switching. In automatic model switching, a dialogue manager runs in real time and as it collects data, it can switch from using a simple dialogue model to a complex model. Confidence intervals can be used to determine when to switch from one model to the next by checking if a complex model's bounds cross over the bounds of the current model. Basically, the dialogue manager switches when it can be sure that the more complex model's ECR is not only higher, but statistically significantly so.

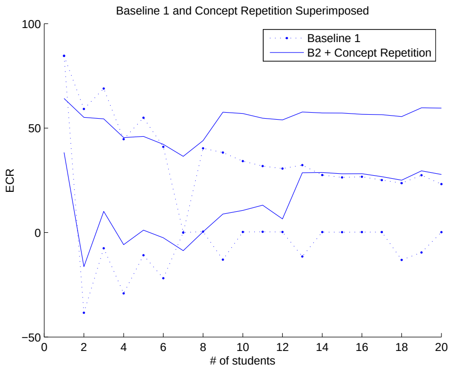

Figure 2: Baseline 1 and Concept Repetition Bounds

5 Conclusions

Past work in using MDP's to improve spoken dialogue systems have usually glossed over the issue of whether or not there was enough training data to develop reliable policies. In this work, we present a numerical method for building confidence intervals for the expected cumulative reward for a learned policy. The proposed approach allows one to (1) better assess the reliability of the expected cumulative reward for a given policy, and (2) perform a refined comparison between policies derived from different MDP models.

We applied this methodology to a prior experiment where the objective was to select the best features to include in the MDP state-space. Our results show that policies constructed from the Baseline 1 and Concept Repetition models are more reliable, given the amount of data available for training. The Concept Repetition model (which is composed of the Concept Repetition, Certainty and Correctness features) was especially useful, as it led to a policy that outperformed the Baseline 1 model, even when we take into account the uncertainty in the model estimates caused by data sparsity. In contrast, for the Baseline 2, Percent Correctness, and Frustration models, the estimates for the expected cumulative reward are much less reliable, and no conclusion can be reliably drawn about the usefulness of these features. In addition, we showed that our confidence interval approach has applications in another MDP problem: model switching.

6 Future Work

As an extension of this work, we are currently investigating in more detail what makes some MDP's reliable or unreliable for a certain data size (such as the case where Baseline 2 does not converge but a more complicated model does, such as Concept Repetition). Our initial findings indicate that, as more data becomes available the bounds tighten for most parameters in the transition matrix. However, for some of the parameters the bounds can remain wide, and that is enough to keep the confidence interval for the expected cumulative reward from converging.

Acknowledgments

We would like to thank Jeff Schneider, Drew Bagnell, Pam Jordan, as well as the ITSPOKE and Pitt NLP groups, and the Dialog on Dialogs group for their help and comments. Finally, we would like to thank the four anonymous reviewers for their comments on the initial version of this paper. Support for this research was provided by NSF grants #0325054 and #0328431.

References

S. Dumais, J. Platt, D. Heckerman, and M. Sahami. 1998. Inductive learning algorithms and representations for text categorization. In *Conference on Information and Knowledge Management*.

M. Frampton and O. Lemon. 2005. Reinforcement learning of dialogue strategies using the user's last dialogue act. In *IJCAI Wkshp. on K&R in Practical Dialogue Systems*.

J. Henderson, O. Lemon, and K. Georgila. 2005. Hybrid reinforcement/supervised learning for dialogue policies from communicator data. In *IJCAI Wkshp. on K&R in Practical Dialogue Systems*.

R. Jaulmes, J. Pineau, and D. Precup. 2005. Active learning in partially observable markov decision processes. In *European Conference on Machine Learning*.

E. Levin and R. Pieraccini. 1997. A stochastic model of computer-human interaction for learning dialogues. In *Proc. of EUROSPEECH '97*.

T. Paek and D. Chickering. 2005. The markov assumption in spoken dialogue management. In *6th SIGDial Workshop on Discourse and Dialogue*.

I. Rivals and L. Personnaz. 2002. Construction of confidence intervals for neural networks based on least squares estimation. In *Neural Networks*.

R. Schapire. 2002. The boosting approach to machine learning: An overview. In *MSRI Workshop on Nonlinear Estimation and Classification*.

S. Singh, M. Kearns, D. Litman, and M. Walker. 1999. Reinforcement learning for spoken dialogue systems. In *Proc. NIPS '99*.

R. Sutton and A. Barto. 1998. *Reinforcement Learning*. The MIT Press.

J. Tetreault and D. Litman. 2006. Comparing the utility of state features in spoken dialogue using reinforcement learning. In *NAACL*.

M. Walker. 2000. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *JAIR*, 12.