2004-12-18

A Bayesian Technique for Task Localization in Multiple Goal Markov Decision Processes

James Carroll

Kevin Seppi
Brigham Young University, kseppi@byu.edu

A Bayesian Technique for Task Localization in Multiple Goal Markov Decision Processes

James L. Carroll
Brigham Young University
Provo, UT 84602 USA
james@jlcarroll.net

Kevin Seppi
Brigham Young University
Provo UT 84602 USA
kseppi@cs.byu.edu

Abstract

In a reinforcement learning task library system for Multiple Goal Markov Decision Process (MGMDP), localization in the task space allows the agent to determine whether a given task is already in its library in order to exploit previously learned experience. Task localization in MGMDPs can be accomplished through a Bayesian approach, however a trivial approach fails when the rewards are not distributed normally. This can be overcome through our Bayesian Task Localization Technique (BTLT).

1 Introduction

Human beings are capable of solving complex and novel control problems with little training data. One likely reason for this ability is that most problems are similar to problems previously encountered, and humans are adept at applying information from past problems to new situations [14].

We therefore propose a task library system as part of the lifelong learning [?] paradigm, or the "learning to learn" framework, in which an agent improves its learning ability as it is exposed to each successive task. Little has been said concerning the theoretical framework of learning to learn in the reinforcement learning domain [1].

Our task library system consists of three main parts: 1) task localization, 2) similarity discovery, and 3) task transfer. *Task localization* determines if a given task is already in the library. If localization determines that the new task is not already in the library then *similarity discovery* determines which tasks from the library are most similar to the new task to be learned. *Task transfer* is the process whereby a similar task from the library (known as the *source* task) is used to improve the learning of a new task (known as the *target* task).

A task library system would be useful for automating the lifelong learning process in general. Such a system would also be useful for automating the shaping process [13] [12] [15], which is a technique for acquiring complex control policies by successively performing task transfer with increasingly complex versions of the problem.

This paper focuses on task localization, the first of the three parts of the library system. Since previous work has been done on both similarity discovery and task transfer, task localization is one of the major missing pieces of a complete task library system. First we will overview the three parts of the task library system, and then focus on the technique for task localization.

1.1 Task Localization

A library of previously learned tasks is only directly useful if there is a technique for recognizing when a situation matches a task that has already been learned. Without this ability, the agent would be forced to re-learn the task. Furthermore, if the system recognizes the task only after re-learning it, then the library has not been helpful.

If the reward structure of a task is given to the agent, task localization is simple. However, since the reward function is not always expressly given to the agent in terms of its states and actions, and since rewards are often received in a stochastic manner, a complete task library system must be able to localize itself in task space by simply observing the distribution of rewards received. Such a situation arises in multiagent situations where the identity or behavior of the

other agent is unknown but may have been encountered before, in any search where the goal location is not known but may have been encountered before, and in control when the system dynamics remain the same but the desired behavior has changed in an unknown but potentially previously encountered way.

A task localization algorithm should satisfy three main properties: it must have 1) *efficiency*, meaning that it is able to localize with as few examples as possible so that localization can be performed before a given task is thoroughly re-learned; 2) *robustness*, meaning that it can function regardless of the distributions of the reward structure encountered; and 3) *adaptability*, meaning that it can adapt to a new situation in reasonable time even if the target task is not in the library. This paper proposes an efficient, robust, and adaptable Bayesian Task Localization Technique (BTLT) for model free, stochastic MGMDPs (Multiple Goal Markov Decision Problems [9]).

1.2 Similarity Discovery

Since task transfer is helpful only when the source task(s) are similar to the target task [11], it is necessary to be able to determine which of the tasks in the library are similar to the new target task. There are many different measures of task similarity that can be used, and different tasks can be similar in one respect while differing significantly in another. It would therefore be desirable if the system could determine multiple similarity measures for the tasks in the library and then automatically select a set of tasks and transfer techniques suited to the given situation. Some preliminary work on task similarity discovery and utilization was explored in [5].

1.3 Task Transfer

Task transfer is the process whereby the source task is used to improve the learning of the target task. Task transfer is used in shaping and lifelong learning. Several different mechanisms for transfer in reinforcement learning have been proposed [11] [4] [3] [8], yet more remains to be accomplished. In general, transfer from similar tasks has been found to be helpful, while transfer from tasks with a low similarity can be detrimental [11]. Most past work in RL task transfer has focused upon the single source task to single target task case. Task libraries will require the simultaneous use of multiple source tasks. [5] is a step in this direction. The agent should also be able to discover structure in the reinforcement learning world as in [16], and transfer information from pieces of different tasks once the

structure has been determined as in [3].

1.4 MDP Notation

For simplicity we assume that the reader is familiar with the basic concepts of MDPs and reinforcement learning [10]. We will use the following notation: we will represent an MDP as a 4-tuple, $(\mathbb{S}, \mathbb{A}, P(s'|s,a), f(r|s,a,s'))$ where $\mathbb{S}$ is the state space, $\mathbb{A}$ is the action space, $P(s'|s,a)$ is the transition matrix, and represents the probability of reaching state s' from state s when performing action a, and $f(r|s,a,s')$ is the probability density of rewards r received when performing action a in state s and transitioning to state s'.

A library of related MDPs is denoted L with identical state space $\mathbb{S}$ and identical action space $\mathbb{A}$. Task $l \in L$ is characterized by its unique stochastic payoff function, denoted by $f_l(r|s,a,s')$ and its state transition probabilities $P_l(s'|s,a)$. When transition probabilities are the same for identical state-action pairs across all tasks in L, formally: $\forall j, i \in L$, $\forall s, s' \in \mathbb{S}$, $\forall a \in \mathbb{A}$ $P_j(s'|s,a) = P_i(s'|s,a)$, then the collection of tasks is known as a *Multiple-goal Markov Decision Problem*, or MGMDP [9]. In an MGMDP, $f(r|s,a,s')$ is the only difference between any two tasks in the library. We assume that this property holds in the remainder of this paper.

Let n be the number of tasks in our library L. Let k represent the new task from which we are sampling and which we are trying to match to some task in our library. For simplicity we define T_i such that $P(T_i) = P(k = i)$, or the probability that our new task k is the same as some task i in our library and where i can be $1...n$, or $n+1$ if the task is new and not in the library.

Since our rewards are received stochastically, let

$$R_k(s,a) \sim \sum_{s'} f_k(r|s,a,s')P(s'|s,a).$$

Our goal is to determine the probability that the target task k matches a task i in the library, $P(T_i)$, by repeatedly sampling $R_k(s,a)$ for various s and a. Let $R_k(s,a)_1...R_k(s,a)_m$ represents $m-$ random samples drawn from $R_k(s,a)$.

2 Previous Work

2.1 MSE

The simplest technique for task localization is to sample from the target task k for some time, and then assume that the task is equivalent to the task in the

Figure 1. A situation where $R(s,a)$ is not distributed normally. The agent either hits or misses the obstacle to its right when attempting to move forward depending upon the amount of slippage in its wheels (modeled by $P(s|s,a)$. Thus the agent either received a reward of 0 or a negative reward if the obstacle is hit. The negative reward, although rare, appears to be nearly impossible under the normal assumption.

library with the lowest mean squared difference in expected reward values [5]:

$$Task = argmin_{j \in L} \sum_{s,a} (E[R_k(s,a)] - E[R_j(s,a)])^2.$$

Because this technique does not take into account any weighting of states, it cannot take advantage of the fact that samples from the reward structure in one state may be more important than the samples taken from another. Furthermore, this technique does not take into account the number of times that a specific $R_k(s,a)$ has been sampled. If $R_k(s,a)$ for some (s,a) has a high variance, then it must be sampled more than $R_k(s,a)$ for another (s,a) with a low variance to achieve the same confidence in its expected reward. It is important to take this confidence into account because, to be useful, localization must be performed before the task is fully re-learned.

Furthermore, the Mean Squared Error Technique does not return a probability, but only a most likely task. Therefore this technique requires that the task actually be in the library in order to function. In order to compute the probability that the target task is unique, and not in the library, a more statistically sound method is needed.

As shown in [5], the MSE technique can be used as

an appropriate distance measure between two learned tasks, but it is not sufficiently efficient to be useful for localization which must be performed before the task has been learned.

2.2 Trivial Bayesian Technique

First we will show that a trivial Bayesian approach to this problem is insufficient, thereby justifying the more complex solution which we will discuss later. If we were performing localization in the state space rather than in the task space, then the standard approach would be to update the probabilities at each step based on the current percepts [2]. In task space $P(r_k(s,a)|T_i)$ is the probability that the observed reward could be generated if the new task k was actually equivalent to task i in the library. By Bayes law $P(T_i|r_k(s,a))$ is

$$P(T_i|r(s,a)) = \frac{P(r(s,a)|T_i)P(T_i)}{P(r(s,a))}$$

where

$$P(r(s,a)) = \sum P(r(s,a)|T_i)P(T_i).$$

Although a similar method has proved effective for localization in the state space, it has several problems when localizing in task space. First, we must compute $P(r(s,a)|T_i)$. If we assume that the rewards are normal, then the computation of $P(r(s,a)|T_i)$ is trivial. However, there are many situations where the reward distribution in standard reinforcement learning is not normally distributed. Figure 1 illustrates a simple case in which the reward function is not normally distributed. In fact, each state-action pair can have its own unique distribution, with no pattern. Thus we must either keep full histograms for each state-action pair (which would be intractable) or we must assume that they have some parametric distribution. Unfortunately, in standard reinforcement learning situations, relatively common values for $R_k(s,a)$ can appear very unlikely under the normal assumption. It is unclear which other parametric distribution could model such situations accurately.

Experimentally, this technique failed in all but the most trivial examples. Therefore this technique must be discarded, and a technique that is more robust to these situations must be considered.

3 Algorithm: Bayesian Task Localization Technique, BTLT

Here we introduce a statistical technique that exhibits all three of our desirable qualities, efficiency, ro-

bustness, and adaptability: the Bayesian Task Localization Algorithm (BTLT).

Although the random variable $R_k(s,a)$ is not normally distributed, by the central limit theorem, a sum of $R_k(s,a)$ samples will be. Thus we want to compute $P(T_i)$, but using a sum of $r_k(s,a)_1...r_k(s,a)_n$ samples drawn from $R_k(s,a)$ rather than computing the probability of each sample as it is observed. Since the mean is computed using a sum, the mean will be approximately normal for modest sample sizes even when $R_k(s,a)$ is not. In all cases the mean is normally distributed for large sample sizes. In our experiments the mean became sufficiently close to normal long before the task could have been learned from scratch.

Because localization is usually performed while the number of samples is still small we would also like to know how confident we are in the mean for every state-action pair so that those means that are the most confident can contribute the most to our localization. We would also like to be able to insert an empirical prior on our belief concerning the mean of $R_k(s,a)$ so that it can be estimated with fewer samples.

$R_k(s,a)$ is thus a random variable with unknown values for the parameter mean $M_k(s,a)$ and standard deviation $S_k(s,a)$. We chose to model $R_k(s,a)$ with a normal gamma model [6]. Strictly speaking, this model requires that $R_k(s,a)$ be normally distributed, however the estimation of $M_k(s,a)$ provided by this technique is robust to deviations from the normal assumption in $R_k(s,a)$ because in the normal gamma model, $E[M_k(s,a)]$ is computed using a sum of individual $R_k(s,a)$'s which will be normally distributed by the central limit theorem. $Var[M_k(s,a)]$ may be slightly low due to the violations of the normal assumption, however empirically this value provides an excellent approximation to our trust in our estimation for the parameter mean.

Under the normal gamma model the prior joint distribution for $M_k(s,a)$ and $S_k(s,a)$ is as follows: the conditional distribution of $M_k(s,a)$ when $1/S_k(s,a)^2 = u$, with $u > 0$, is a normal distribution with mean θ and precision τu such that $-\infty < \theta < \infty$ and $\tau > 0$, and the marginal distribution of $1/S_k(s,a)^2$ is a gamma distribution with parameters α and β such that $\alpha > 0$ and $\beta > 0$. These four simple parameters are sufficient to represent our model of $R(s,a)$. The posterior joint distribution of $M_k(s,a)$ and $1/S_k(s,a)^2$ when $R_k(s,a)_i = r_k(s,a)_i (i = 1,...,n)$ is also a normal gamma distribution parameterized by τ, θ, α, and β, and updated as follows:

$$\tau' = \tau + n,$$

$$\theta' = \frac{\tau\theta + n\overline{r_k(s,a)}}{\tau + n},$$

$$\alpha' = \alpha + \frac{n}{2},$$

$$\beta' = \beta + \frac{1}{2}\sum_{i=1}^{n}(r_k(s,a)_i - \overline{r_k(s,a)})^2 + \frac{\tau n(\overline{r_k(s,a)} - \theta)^2}{2(\tau + n)}.$$

The marginal for $M_k(s,a)$ is a t distribution with 2α degrees of freedom and variance $\beta/\tau(\alpha - 1)$ [7].

Prior distributions for $M_k(s,a)$ were computed empirically from the other states in our task, and from the other tasks in L. Although task localization would not be performed when learning the first task that the system encounters, the normal gamma model must still be built for all tasks in the library. Because there are no other tasks in the library when the first task is learned, priors must be estimated subjectively or drawn empirically from the other state-action pairs in the same task. As more tasks are inserted into the library, more information can be drawn from the corresponding state-action pairs from the other tasks in the library in order to create better priors.

With these priors in place, we can more efficiently model $M(s,a)$ for our target and source tasks. Now our computation for $P(T_i)$ is fairly straightforward:

$$P(T_i|M_k(s,a)) = \frac{P(M_k(s,a)|T_i)P(T_i)}{P(M_k(s,a))}$$

by Bayes Theorem, and

$$P(M_k(s,a)) = \sum P(M_k(s,a)|T_i)P(T_i),$$

where $P(M_k(s,a)|T_i)$ can be found by computing the likelihood of $E[M_k(s,a)]$ in the t distribution with $mean = E[M_i(s,a)]$ and $Var = E[S_i(s,a)^2]/n_k(s,a)$ where $n_k(s,a)$ is the number of samples taken for action a in state s and task k, and with $2\alpha_i(s,a)$ degrees of freedom. This is true if we assume that $E[M_i(s,a)] \approx \mu_i(s,a)$ and that $E[S_i(s,a)] \approx \sigma_i(s,a)$ where $\mu_i(s,a)$ and $\sigma_i(s,a)$ are the true values for the mean and standard deviation of $R_i(s,a)$. This will be true so long as the sample size $n_i(s,a)$ is large. Thus we assume that we have thoroughly learned the tasks in the library, but we make no such assumption about the target task k that we are attempting to localize.

This means that we can use this technique to localize a target task k within a library before k is thoroughly learned, so long as all the source tasks in our library are thoroughly learned. Unfortunately, this technique requires that task k be in the library. If the

target task k is simply added to the library as another task, $n+1$, and the localization technique run, because $E[M_{n+1}(s,a)]$ is not approximately equal to $\mu_{n+1}(s,a)$ for low $n_k(s,a)$, the algorithm does not function correctly until the new task is thoroughly learned, at which time it is too late to be of use.

The solution to this problem is to assume that the task is in the library and then determine the task in the library that is most likely identical with the target task. We will call this task g. Then a second statistical test is used to determine if $k = g$. This is a simple hypothesis test with two hypothesis, $H_0 : \mu_k(s,a) - \mu_g(s,a) = 0$, $H_1 : \mu_k(s,a) - \mu_g(s,a) \neq 0$ for all s and a. Strictly speaking since the reward structure is continuous, the probability of H_0 is always 0. But the desired behavior is for the agent to assume that task k is equal to task g unless there is enough evidence to reject this hypothesis. Under H_0 we would expect $E[M_k(s,a)] - E[M_g(s,a)] \sim N(0, Var[M_k(s,a)] + Var[M_g(s,a)])$. If $E[M_k(s,a)] - E[M_g(s,a)]$ is within a 95% confidence interval of $N(0, Var[M_k(s,a)] + Var[M_g(s,a)])$, then we keep the null hypothesis H_0 and assume that task k is the same as task g. Otherwise we reject the null hypothesis, and assume that task k is not in the library.

This process is computationally intensive if all states in the space are considered at every step. However, the BTLT is an anytime algorithm that can look at states as they come, and can look at more states if there is extra time between interactions with the world. The agent can simply follow the policy outlined by the most probable task, while the computation to determine the most probable task updates this value as often as time allows. Furthermore, the algorithm is trivially parallelizable, which could drastically increase the number of localization computations done between each interaction with the world.

In summary, this technique avoids the violation of the normal assumption that plagued the Trivial Bayes Technique, and it allows good guesses for $P(T_i)$ with fewer interactions with the world than the Mean Squared Error Technique (MSE). It also allows the agent to weight the importance of states where its confidence in $M_k(s,a)$ is greater.

4 Methodology

We used a complex grid world in order to test our task localization algorithms. In this world an agent faces one of eight directions, and has 4 actions. The agent can either turn 45° to the right, 45° to the left, go forward, or go backward. This generates a much larger state space than in a traditional grid world.
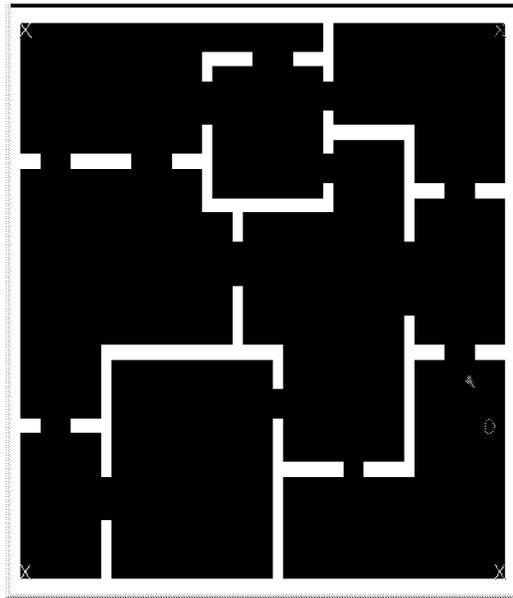
Figure 2. Complex grid-world, showing an agent ($\triangle$), a randomly placed starting point (O), and four stochastic goals (X) in each of the four corners.

Moves are probabilistic, and when an agent moves either forward or backward, the agent will either move one space counterclockwise, or clockwise from the agent's expected destination with a certain probability. The amount of randomness can be set by the experimenter, and can vary from task to task, but in our experiments was set uniformly between each task in a library to maintain the MGMDP property. The world can have multiple goals that can be either absorbing or not, and which can generate rewards probabilistically when they are reached according to a normally distributed reward function with a given mean and variance. Each goal can set these values independently. Although the rewards pay off according to a normal distribution, because the transitions into these states are random, the reward seen by the agents in any state-action pair will only be normal if the randomness in the transitions is set to 0. A reward of -3 is given whenever an agent hits a wall, and a reward of -1 is given whenever an agent moves backwards. These payoffs are uniform for all tasks, and therefore provide a large potential for generalization within a single task and between tasks in a library. Thus strong priors can be placed on certain payoffs that will be uniform across all tasks, while weaker priors can be placed on other states, which are more likely to vary between tasks.

We placed probabilistic goals in each of the four

corners. A library of tasks was generated by varying the mean and standard deviation of each goal. In this case every task had a goal in the same location, but the value of that goal varied from task to task. We also created a set of tasks with a single goal, placed randomly throughout the world. We also ran experiments with varying amounts of randomness in the transition probabilities. This generated an excellent test bed for task localization where an agent must sample from each reward repeatedly to determine the parameters for each reward's payoff.

5   Results

The Trivial Bayesian Technique functioned so long as the rewards were distributed normally. However, no consistent convergence was noted in any of our task suites when the world randomness generated non-normal rewards. In these cases, the agent would compute inappropriately small probabilities for some rewards received, which would cause the probabilities for the tasks to swing erratically. Which task appeared most probable often depended upon slight variations in the proportion that these situations were seen in each task in the library.

The simple Mean Squared Error Technique did not require the normal assumption, and eventually converged to the correct solution even in situations without normally distributed rewards. However this technique required many samples to localize correctly. This technique provided a distance metric between tasks, and could pick out the most similar task by finding the task with the minimum distance, but if the task was not in the library, it had no mechanism for determining when to reject the hypothesis that the current task was somewhere in the library [5].

In the case with the single goal locations, BTLT was able to localize after sampling from the goal state a few times whether or not the transitions caused the rewards to be distributed normally (See Figure 3). Notice that our assumption that the task is in the library dictates that the probabilities sum to one at any given time. Often the probabilities would swing quite suddenly to the correct answer when an essential piece of information was sampled during the agent's exploration of the world. The number of examples needed depended on the accuracy of the priors and the amount of steps taken before the key samples were drawn. This depended upon the size of the world, the placement of the starting position, and the placement of the goal, and therefore varied from trial to trial. Figures 4 and 5 are representative of the sorts of results encountered. Notice that localization took place long before learning
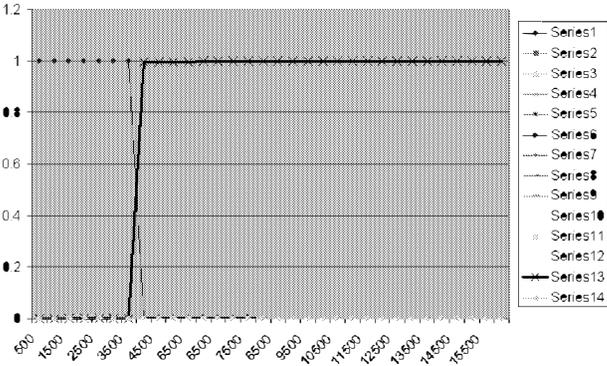

Figure 3. The probabilities of 14 tasks as BTLT localizes assuming that the task is in the library, using a random exploration technique. The y axis is the probability, and the x axis is the number of world steps taken. World randomness is 1 % and therefore the rewards were not normally distributed in this experiment.
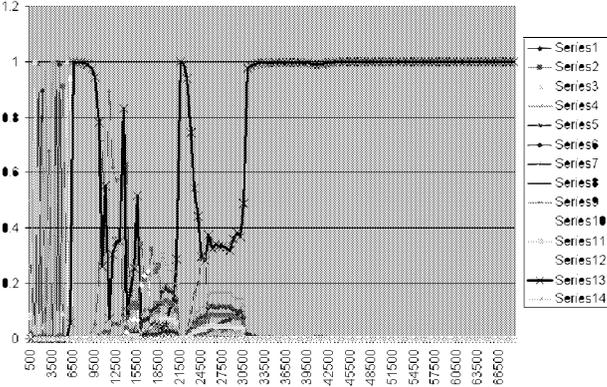
Figure 4. The probabilities of 14 tasks as BTLT localizes assuming that the task is in the library, with task switching to the most probable task at any given time.

6
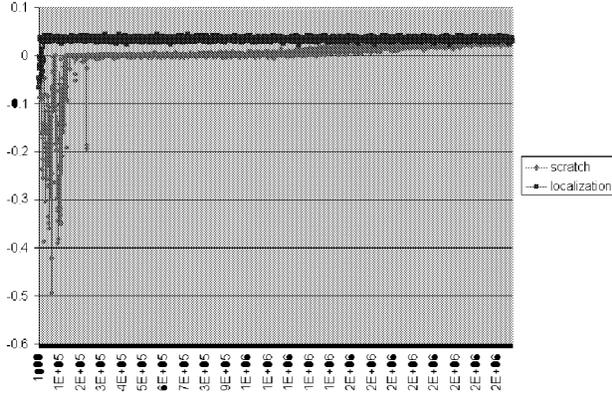
Figure 5. Average reward received when learning from scratch vs BTLT based task switching when the task is in the library. When compared to learning from scratch the time to localize appears immediate.
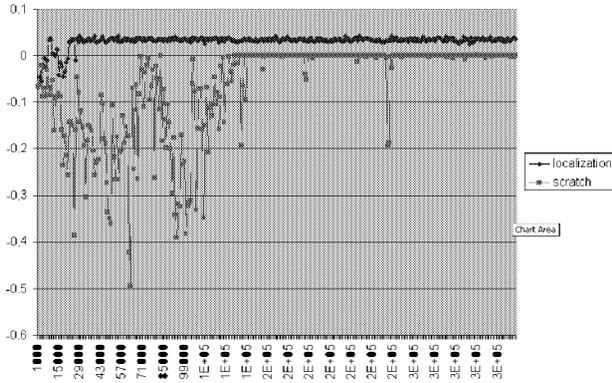
Figure 6. A closeup of Figure 5 showing that the number of negative rewards received before localization were considerably less than when learning from scratch.

from scratch could have re-learned the problem. This sort of improvement was seen in all the experiments in this domain.

In the more complex case where the rewards themselves paid off stochastically, and with very similar distributions, it is true that the agent had to sample from the goal locations more often in order to correctly localize. The amount of samples needed depended on the degree of similarity between the tasks in the library. The more similar the two tasks, the more samples were required for localization. However, the more similar the two tasks, the more their policies overlapped and the less the localization delay effected the average reward received.

Notice that it took much longer to localize when using task switching than it did when using random exploration (see Figures 3 and 4). This is because the agent took time performing inappropriate tasks while the probabilities were being recomputed and was less likely to stumble into the key sample that would have allowed the agent to localize more rapidly.

One unexpected result was that the agent initially explored its domain by performing policies that coincided with tasks in its library, and therefore received fewer negative rewards during its exploration phase. This added benefit happens because fault avoidant behavior is often uniform across tasks (see Figure 6).

When the goal was not in the library, the agent would recognize this fact with very few examples of an unexpected reward, as long as the confidence intervals were set correctly. We noticed that the results were very sensitive to this parameter. We also noticed that to avoid mistakenly rejecting the hypothesis that the task is in the library, results from state-action pairs with too few samples (in our case approximately less than or equal to 6) should be ignored.

In some pathological cases (for example if we placed two goals in the world, one where a goal was in one of the tasks in the library, and another, far out of the way in a corner) BTLT can initially converge to the wrong solution. However, if the second goal was sufficiently sampled the agent would realize that the task was novel.

6 Conclusions

We have shown how task localization, one of three major steps in the creation of a task library system, can be accomplished with a Bayesian approach in the MGMDP case. We have shown that the Trivial Bayesian Technique fails because the rewards received in most reinforcement learning problems are not distributed normally. Further, the MSE technique re-

quires more samples than BTLT to effectively localize in the task space.

BTLT overcomes these problems by placing priors on the frequencies with which tasks are observed, and then updating these beliefs based upon observations from the reward structure of the target task. The reward structure is also modeled with a Bayesian technique, using the normal gamma model. Priors for the reward structure in a given state action pair of a given task are drawn in various ways from the other tasks in the library. This allows the agent to make more appropriate guesses about the reward structure with fewer observations, and allows the agent to localize in task space with fewer observations.

Although random exploration can allow faster localization, task switching can avoid many negative rewards received while the task space is being explored. Several parameters must be tuned if the hypothesis, that the task is not in the library, must be tested.

Because this technique can converge to a suboptimal solution without sufficient exploration, it should be combined with some other exploration vs. exploitation tradeoff technique. These techniques are well understood and have been widely studied.

7   Future Work

Currently the Mean Squared Error Technique is the standard distance measure for two tasks in a library [5]. The major drawback of this method is that it requires that both tasks be correctly learned before an accurate measure of their similarity can be made. The Bayesian Task Localization Technique could be modified to not only determine the probability that one task matches another, but also to provide a distance measure between various tasks in the library that could be used for task clustering as in [5]. Just as BTLT allows localization before the target task is completely learned, such a system should be able to provide a more accurate approximation of task similarity before the target task is thoroughly learned.

Currently the source task(s) for transfer are handpicked by the user. Once distance measures have been determined that can accurately approximate the task similarity before the target task is learned, the entire transfer process could be automated. Once the agent has determined that its current situation does not match any task in its library, it should then find the most similar task in its library, and apply transfer, thus automating this part of the process.

References

[1] J. Baxter. Theoretical models of learning to learn. In L. P. Sebastian Thrun, editor, *Learning to Learn*, chapter 4, pages 71–94. Morgan Kaufmann, San Francisco, Ca., 1998.

[2] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *AAAI/IAAI, Vol. 2*, pages 896–901, 1996.

[3] J. L. Carroll and T. Peterson. Fixed vs. dynamic sub-transfer in reinforcement learning. In *ICMLA*, Las Vegas Nevada, USA, 2002. CSREA Press.

[4] J. L. Carroll, T. Peterson, and N. Owens. Memory-guided exploration in reinforcement learning. In *In IJCNN2001*, Washington, D.C., 2001.

[5] J. L. Carroll, T. Peterson, and K. Seppi. Reinforcement learning task clustering (rltc). In *ICMLA*, LA California USA, 2003.

[6] R. Dearden, N. Friedman, and S. J. Russell. Bayesian q-learning. In *AAAI/IAAI*, pages 761–768, 1998.

[7] M. H. DeGroot. *Optimal Statistical Decisions*. McGraw-Hill Book Company, New York, 1970.

[8] K. R. Dixon, R. J. Malak, and P. K. Khosla. Incorporating prior knowledge and previously learned information into reinforcement learning agents. In *Institute for Complex Engineered Systems Technical Report Series*, Carnegie Mellon University, January 2000.

[9] D. Foster and P. Dayan. Structure in the space of value functions. *Machine Learning*, 49:325–346, 2002.

[10] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *JAIR*, 4:237–285, 1996.

[11] T. Peterson, N. Owens, and J. L. Carroll. Automated shaping as applied to robot navigation. In *IEEE International Conference on Robotics and Automation*, Korea, 2001.

[12] J. Randlov. Solving complex problems with reinforcement learning. In *PH.D. Dissertation*. University of Copenhagen, 2001.

[13] J. Randlov and P. Alstrom. Learning to drive a bicycle using reinforcement learning and shaping. In *Machine Learning: Proceedings of the Eleventh International Conference*. Morgan Kaufmann, CA, 1999.

[14] A. Robins. Transfer in cognition. In L. P. Sebastian Thrun, editor, *Learning to Learn*, chapter 3, pages 45–67. Morgan Kaufmann, San Francisco, Ca., 1998.

[15] B. F. Skinner. *The Behavior of Organisms: An Experimental Analysis*. Prentice Hall, Englewood Cliffs, New Jersey, 1938.

[16] S. Thrun and J. O'Sullivan. Discovering structure in multiple learning tasks: The TC algorithm. In *International Conference on Machine Learning*, pages 489–497, 1996.