In *The Eleventh International Conference on Development and Learning (ICDL)* (ICDL 2012), San Diego, CA, Nov 2012.

Intrinsically Motivated Model Learning for a Developing Curious Agent

Todd Hester and Peter Stone
Department of Computer Science
The University of Texas at Austin
{todd,pstone}@cs.utexas.edu

***Abstract*— Reinforcement Learning (RL) agents are typically deployed to learn a specific, concrete task based on a pre-defined reward function. However, in some cases an agent may be able to gain experience in the domain prior to being given a task. In such cases, intrinsic motivation can be used to enable the agent to learn a useful model of the environment that is likely to help it learn its eventual tasks more efficiently. This paper presents the TEXPLORE with Variance-And-Novelty-Intrinsic-Rewards algorithm (TEXPLORE-VANIR), an intrinsically motivated model-based RL algorithm. The algorithm learns models of the transition dynamics of a domain using random forests. It calculates two different intrinsic motivations from this model: one to explore where the model is uncertain, and one to acquire novel experiences that the model has not yet been trained on. This paper presents experiments demonstrating that the combination of these two intrinsic rewards enables the algorithm to learn an accurate model of a domain with no external rewards and that the learned model can be used afterward to perform tasks in the domain. While learning the model, the agent explores the domain in a developing and curious way, progressively learning more complex skills. In addition, the experiments show that combining the agent's intrinsic rewards with external task rewards enables the agent to learn faster than using external rewards alone.**

I. Introduction

Reinforcement Learning (RL) agents could be useful in society because of their ability to learn and adapt to new environments and tasks. Traditionally, RL agents learn to accomplish a specific, concrete task based on a pre-defined reward function. However, in some cases an agent may be able to gain experience in the domain prior to being given this task. For example, a future domestic robot may be placed in a home and only later given various tasks to accomplish. In such cases, intrinsic motivation can be used to enable the agent to learn a useful model of the environment that can help it learn its eventual tasks more efficiently.

Past work on intrinsically motivated agents arises from two different goals [1]. The first goal comes from the active learning community, which uses intrinsic motivation to improve the sample efficiency of RL. Their goal is to help the agent to maximize its knowledge about the world and its ability to control it. The second goal comes from the developmental learning community, and is to enable cumulative, open-ended learning on robots. Our goal is to use intrinsic motivation towards both goals: 1) to improve the sample efficiency of learning, particularly in tasks with little or no external rewards; and 2) to enable the agent to perform open-ended learning without external rewards.

This paper presents an intrinsically motivated model-based RL algorithm, called TEXPLORE with Variance-And-Novelty-Intrinsic-Rewards (TEXPLORE-VANIR), that uses intrinsic motivation both for improved sample efficiency and to give the agent a curiosity drive. The agent is based on a model-based RL framework and is motivated to learn models of domains without external rewards as efficiently as possible. TEXPLORE-VANIR combines model learning through the use of random forests with two unique intrinsic rewards calculated from this model. The first reward is based on *variance* in its models' predictions to drive the agent to explore where its model is uncertain. The second reward drives the agent to *novel* states which are the most different from what its models have been trained on. The combination of these two rewards enables the agent to explore in a developing curious way, learn progressively more complex skills, and learn a useful model of the domain very efficiently.

This paper presents two main contributions:

1) Novel methods for obtaining intrinsic rewards from a random-forest-based model of the world.
2) The TEXPLORE-VANIR algorithm for intrinsically motivated model learning, which has been released open-source as a ROS package: `http://www.ros.org/wiki/rl-texplore-ros-pkg`.

Section IV presents experiments showing that TEXPLORE-VANIR: 1) learns a model more efficiently than other methods; 2) explores in a developing, curious way; and 3) can use its learned model later to perform tasks specified by a reward function. In addition, it shows that the agent can use the intrinsic rewards in conjunction with external rewards to learn a task faster than if using external rewards alone.

II. Background

This section presents background on Reinforcement Learning (RL). We adopt the standard Markov Decision Process (MDP) formalism for this work [2]. An MDP is defined by a tuple $\langle S, A, R, T \rangle$, which consists of a set of states S, a set of actions A, a reward function $R(s,a)$, and a transition function $T(s,a,s') = P(s'|s,a)$. In each state $s \in S$, the agent takes an action $a \in A$. Upon taking this action, the agent receives a reward $R(s,a)$ and reaches a new state s', determined from the probability distribution $P(s'|s,a)$. Many domains utilize a factored state representation, where the state s is represented by a vector of n state variables: $s = \langle x_1, x_2, ..., x_n \rangle$. A policy π specifies for each state which action the agent will take.

The goal of the agent is to find the policy π mapping states to actions that maximizes the expected discounted total reward over the agent's lifetime. The value $Q^{\pi}(s,a)$ of a given state-action pair (s,a) is an estimate of the expected future reward that can be obtained from (s,a) when following policy π. The optimal value function $Q^*(s,a)$ provides maximal values in all states and is determined by solving the Bellman equation:

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q^*(s',a'), \quad (1)$$

where $0 < \gamma < 1$ is the discount factor. The optimal policy π is then:

$$\pi(s) = \text{argmax}_a Q^*(s,a). \quad (2)$$

RL methods fall into two general classes: model-based and model-free methods. Model-based RL methods learn a model of the domain by approximating $R(s,a)$ and $P(s'|s,a)$ for each state and action. The agent can then calculate a policy (i.e. plan) using this model. Model-free methods update the values of actions only when taking them in the real task. One of the advantages of model-based methods is their ability to plan multi-step exploration trajectories. The agent can plan a policy to reach intrinsic rewards added into its model to drive exploration to interesting state-actions.

This work takes the approach of using a model-based RL algorithm in a domain with no external rewards. This approach can be thought of as a pure exploration problem, where the agent's goal is simply to learn as much about the world as possible. TEXPLORE-VANIR extends a model-based RL algorithm called TEXPLORE [3] to use intrinsic motivation to quickly learn an accurate model in domains with no external rewards.

III. TEXPLORE-VANIR

Our goal is to develop an intrinsically motivated curious agent using RL. This agent should use intrinsic rewards to 1) efficiently learn a useful model of the domain's transition dynamics; and 2) explore in a developing curious way. To this end, we have the following desiderata for such an algorithm:

1) The algorithm should be model-based, both to enable multi-step exploration trajectories and to allow the agent to use the learned model later to perform tasks.
2) It should incorporate generalization into its model learning so as to learn the model quickly.
3) It should not be required to visit every state-action, because doing so is intractable in large domains.

This paper presents the TEXPLORE-VANIR algorithm, which has all of these properties. TEXPLORE-VANIR follows the typical approach of a model-based RL agent. It plans a policy using its learned model (including intrinsic rewards), takes actions following that policy, acquiring new experiences which are used to improve its model, and repeats. In order to be applicable to robots, TEXPLORE-VANIR uses the Real-Time Model Based Architecture [4]. This architecture uses approximate planning with UCT [5] and parallelizes the model learning, planning, and acting such that the agent can take actions in real-time at a specified frequency.

A. Model Learning

Making the intrinsically motivated agent model-based enables it to: 1) plan multi-step exploration trajectories; 2) learn faster than model-free approaches; and 3) use the learned model to solve tasks given to it after its learning. It is desirable for the model to generalize the learned transition and reward dynamics across state-actions. This generalization enables the model to make predictions about unseen or infrequently visited state-actions, and therefore the agent does not have to visit every state-action. Thus, TEXPLORE-VANIR approaches the model learning task as a supervised learning problem, with the current state and action as the input, and the next state as the output to be predicted.

TEXPLORE-VANIR is built upon the TEXPLORE algorithm [3], which uses random forests to learn separate predictions of each of the n state features in the domain. A random forest [6] is a collection of m decision trees, each of which differs because it is trained on a random subset of experiences and has some randomness when choosing splits at the decision nodes. The agent then plans over the average of the predictions made by each tree in the forest. The decision trees work well because they generalize broadly at first, but can be refined with training to make accurate predictions for individual state-actions. Each tree in the forest represents a different hypothesis of the true model of the domain. The variance of the different trees' predictions can be used as a measure of the uncertainty in the model.

B. Intrinsic Motivation

The main contribution of this paper is a method for extending the model-based TEXPLORE algorithm for learning specific RL tasks to the intrinsically motivated TEXPLORE-VANIR algorithm. The best intrinsic rewards to use to improve the efficiency of model-learning are highly dependent on the type of model being learned. With the random forest models TEXPLORE uses, we hypothesize that the following two intrinsic motivations will perform the best: 1) preferring to explore areas of the state space where there is a large degree of uncertainty in the model, and 2) preferring regions of the state space that are far from previously explored areas (regardless of how certain the model is).

The variance of the predictions of each of the trees in the forest can be used to motivate the agent towards the state-actions where its models disagree. These state-actions are the ones where there are still multiple hypotheses of the true model of the domain. TEXPLORE-VANIR calculates a measure of the variance in the predictions of each state feature for a given state-action:

$$D(s,a) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{m} D_{KL}(P_j(x_i|s,a)||P_k(x_i|s,a)), \quad (3)$$

where for every pair of models (j and k) in the forest, it sums the KL-divergences between the predicted probability distributions for each feature i. $D(s,a)$ measures how much the predictions of the different models disagree. This measure is different than just measuring where the predictions are noisy, as $D(s,a)$ will be 0 if all the tree models predict the same stochastic outcome distribution. An intrinsic reward proportional to this variance measure, the VARIANCE-REWARD, is incorporated into the agent's model for planning:

$$R(s,a) = vD(s,a), \tag{4}$$

where v is a coefficient determining how big this reward should be.

This reward will drive the agent to the state-actions where its models have not yet converged to a single hypothesis of the world's dynamics. However, there will still be cases where all of the agent's models make incorrect predictions. For the random forest model that TEXPLORE-VANIR uses, the model is more likely to be incorrect when it has to generalize its predictions farther from the experiences it is trained on. Therefore, TEXPLORE-VANIR uses a second intrinsic reward based on the L_1 distance in feature space from a given state-action and the nearest one that the model has been trained on. This distance is calculated separately for each action. For an action a, X_a is the set of all the states where this action was taken. Then, $\delta(s,a)$ is the L_1 distance from the given state s to the nearest state where action a has been taken:

$$\delta(s,a) = \min_{s_x \in X_a} ||s - s_x||_1, \tag{5}$$

where each feature is normalized to range from 0 to 1. A reward proportional to this distance, the NOVELTY-REWARD, drives the agent to explore the state-actions that are the most novel compared to the previously seen state-actions:

$$R(s,a) = n\delta(s,a), \tag{6}$$

where n is a coefficient determining how big this reward should be. One nice property of this reward is that given enough time, it will drive the agent to explore *all* the state-actions in the domain, as any unvisited state-action is different in some feature from the visited ones. However, it will start out driving the agent to explore the state-actions that are the most different from ones it has seen.

The TEXPLORE with Variance-And-Novelty-Intrinsic-Rewards algorithm (TEXPLORE-VANIR) is completed by combining these two intrinsic rewards. They can be combined with different weightings of their coefficients (v and n), or with an external reward defining a task. A combination of the two intrinsic rewards should drive the agent to learn a model more efficiently, as well as explore in a developing and curious way: seeking out novel and interesting state-actions, while exploring increasingly complex parts of the domain.

IV. Empirical Results

Evaluating the benefits of intrinsic motivation is not as straightforward as evaluating a standard RL agent on a specific task. Rather than attempting to accrue reward on a given task, a curious agent's goal is better stated as preparing itself for any task. We therefore evaluate TEXPLORE-VANIR in four ways on a complex domain with no external rewards. First, we measure the accuracy of the agent's learned model in predicting the domain's transition dynamics. Second, we test whether the learned model can be used to perform tasks in the domain when given a reward function. Third, we examine the agent's exploration to see if it is exploring in a developing, curious way. Finally, we demonstrate that TEXPLORE-VANIR can combine its intrinsic rewards with external rewards to learn faster than if it was given only external rewards. These results demonstrate that the intrinsic rewards and model learning approach TEXPLORE-VANIR uses are sufficient for the agent to explore in a developing curious way and to efficiently learn a transition model that is useful for performing tasks in the domain.

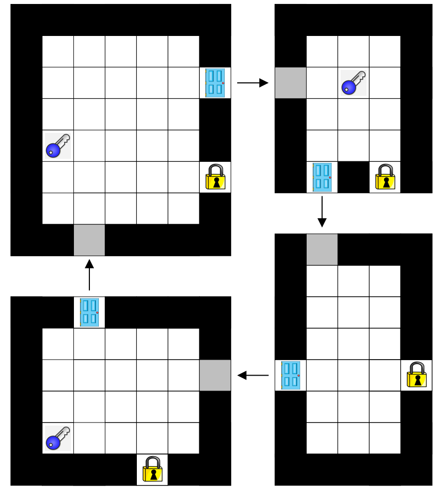


Fig. 1. The Light World domain. In each room, the agent must navigate to the key, PICKUP they key, navigate to the lock, PRESS it, and then navigate to and exit through the door to the next room.

The agent is tested on the Light World domain [7], shown in Figure 1. In this domain, the agent goes through a series of rooms. Each room has a door, a lock, and possibly a key. The agent must go to the lock and press it to open the door, at which point it can then leave the room. It cannot go back through the door in the opposite direction. If a key is present, it must pickup the key before pressing the lock. Open doors, locks, and keys each emit a different color light that the agent can see. The agent has sensors that detect each color light in each cardinal direction. The agent's state is made up of 17 different features: its X and Y location in the room, the ID of the room it is in, whether it has the KEY, whether the door is LOCKED, as well as the values of the 12 light sensors, which detect each of the three color lights in the four cardinal directions. The agent can take six possible actions: it can move in each of the four cardinal directions, PRESS the lock, or PICKUP the key. The first four actions are stochastic; they move the agent in the intended direction with probability 0.9 and to either side with probability 0.05 each. The PRESS and PICKUP actions are only effective when the agent is on top of the lock and the key, respectively, and then only with probability 0.9. The agent starts in a random state in the top left room in the domain, and can proceed through the rooms indefinitely.

This domain is well-suited for this task because the domain has a rich feature space and complex dynamics. There are simple actions that move the agent, as well as more complex actions (PICKUP and PRESS) that interact with objects in different ways. There is a progression of the complexity of the uses of these two actions. Picking up the key is easier than pressing the lock, as the lock requires the agent to have already picked up the key and not yet unlocked the door.

Based on informal testing, we set TEXPLORE-VANIR's parameters to $v = 1$ and $n = 3$. TEXPLORE-VANIR is tested against the following agents:

1) Agent which selects actions *randomly*
2) Agent which is given an intrinsic motivation for regions with more *competence progress* (based on R-IAC [8])
3) Agent which is given an intrinsic motivation for regions with more *prediction errors*
4) Agent which uses R-MAX style rewards (terminal reward of R_{max} for state-actions with fewer than m visits)
5) Agent which acts randomly with a *tabular* model
6) R-MAX algorithm [9]

These six algorithms provide four different ways to explore using TEXPLORE-VANIR's random forest model, as well two approaches using a tabular model. The tabular model is initialized to predict self-transitions for state-actions that have not been visited.

One of the more well-known intrinsic motivation algorithms is Robust Intelligent Adaptive Curiosity (R-IAC) [8]. R-IAC does not adopt the RL framework, but is similar in many respects. R-IAC splits the state space into regions and learns a model of the transition dynamics in each region. It maintains an error curve for each region and uses the slope of this curve as the intrinsic reward for the agent, driving the agent to explore the areas where its model is improving the most (rewarding *competence progress*). This approach is intended for very large multi-dimensional continuous domains where learning may take many thousands of steps. We have created a method based on this idea to compare with our approach (the *Competence Progress* method). This method splits the state space into random regions at the start, maintains error curves in each region, and provides intrinsic rewards based on competence progress within a region. These intrinsic rewards are combined with the same TEXPLORE model learning approach as the other methods. As another comparison, the *Prediction Error* method uses the same regions, but rewards areas with high prediction error.

All the algorithms are run in the Light World domain for 1000 steps without any external reward. During this phase, the agent is free to play and explore in the domain, all the while learning a model of the dynamics of this world. All of the algorithms use the RTMBA parallel architecture [4] and take 2.5 actions per second.

First, we examine the accuracy of the agent's learned model. After every 25 steps, 5000 state-actions from the domain are randomly sampled and the variational distance between the model's predicted next state probabilities are compared with the true next state probabilities. Figure 2 shows the variational distance between these distributions, averaged over the 5000 sampled state-actions. This figure shows that TEXPLORE-VANIR learns significantly more accurate models than the other methods ($p < 0.025$). The next best algorithm is R-MAX. However, using R-MAX style reward with the TEXPLORE model strategy is worse than acting randomly. This result illustrates our point that the best intrinsic reward is dependent on the particular model learning approach that is used. The method rewarding visiting regions with high prediction error performs poorly, possibly because it is not visiting the right state-actions within these regions.

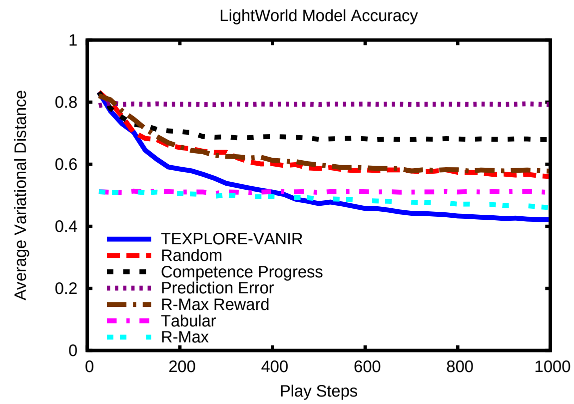
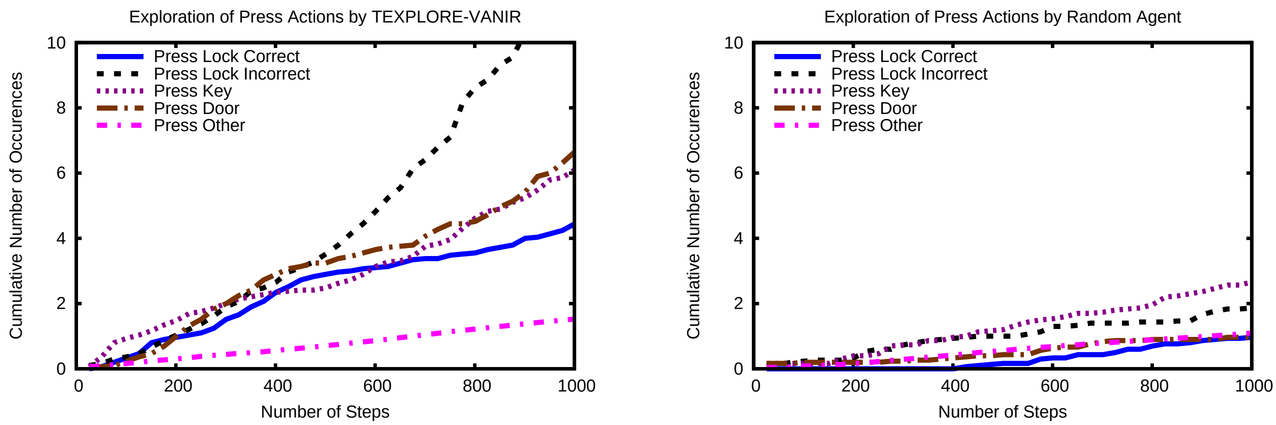


Fig. 2. Accuracy of each algorithm's model plotted versus number of steps the agent has taken, averaged over 30 trials and 5000 randomly sampled state-actions. TEXPLORE-VANIR learns the most accurate models.

While TEXPLORE-VANIR and R-MAX appear to learn fairly accurate models, it is more important for the algorithms to be accurate in the interesting and useful parts of the domain than for them to be accurate about every state-action. Therefore, we next test if the learned models are useful to perform a task. After the algorithms learned models without rewards for 1000 steps, they are provided with a reward function for a task. The task is for the agent to continue moving through the rooms (requiring it to use the keys and locks). The reward function is a reward of 10 for moving from one room to the next, and a reward of 0 for all other actions. In this second phase, the agents act greedily with respect to their previously learned transition models and the given external reward function with *no* intrinsic rewards for 3000 steps.

Figure 3 shows the cumulative external reward received by each algorithm over the 3000 steps of the task. Again, TEXPLORE-VANIR performs the best, slightly out-performing R-MAX and significantly out-performing the other methods ($p < 0.001$). Learning an accurate transition model appears to lead to good performance on the task, as both TEXPLORE-VANIR and R-MAX perform well on the task.

Next, the exploration of the TEXPLORE-VANIR agent is examined. In addition to learning an accurate and useful model, we desire the agent to exhibit a developing curiosity. Precisely, the agent should progressively learn more complex skills in the domain, rather than explore randomly or exhaustively. Figures 4(a) and 4(b) show the cumulative number of times that TEXPLORE-VANIR and the random agent select the PRESS action in various states over 1000 steps in the task with no external rewards, averaged over 30 trials. Comparing



(a) TEXPLORE-VANIR (b) Random Agent.

Fig. 4. This plot shows the cumulative number of times that TEXPLORE-VANIR and a Random Agent select the PRESS action in various states over 1000 steps in the task with no external rewards, averaged over 30 trials. Note that the random agent attempts the PRESS action much less than TEXPLORE-VANIR does. TEXPLORE-VANIR starts out trying to PRESS the key, which is the easiest object to find, and eventually does learn to press the lock, but has difficulty learning when to press the lock (it must be with the key but without the door already being open). The agent does not try calling the PRESS action on random states very often. In contrast, the random agent calls PRESS action on random states more often than it calls it correctly on the lock.

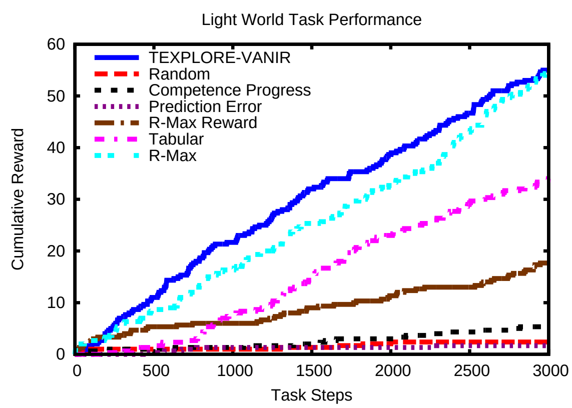


Fig. 3. Cumulative rewards received by each algorithm over the 3000 steps of the task, averaged over 30 trials. Agents act greedily with respect to their previously learned transition model and the given external reward function. TEXPLORE-VANIR receives the most reward.

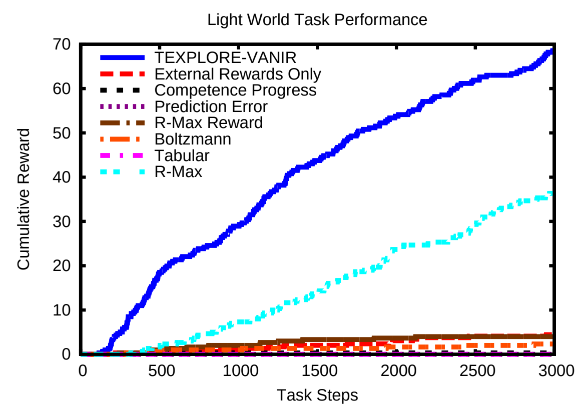


Fig. 5. Cumulative rewards received by each algorithm, using intrinsic and external rewards combined, over the 3000 steps of the task, averaged over 30 trials. TEXPLORE-VANIR receives the most reward, while the agent using only external rewards performs very poorly.

the two figures shows that TEXPLORE-VANIR calls the PRESS action many more times than the random agent. Figure 4(a) also shows that TEXPLORE-VANIR tries PRESS on objects more often than on random states in the domain. In contrast, Figure 4(b) shows that the random agent tries PRESS on arbitrary states more often than it uses it correctly.

Analyzing the exploration of TEXPLORE-VANIR further, Figure 4(a) shows that it initially tries PRESS on the key, which is the easiest object to access, then tries it on the lock, and then on the door. The figure also shows that TEXPLORE-VANIR takes longer to learn the correct dynamics of the lock, as it continues to PRESS the lock incorrectly, either without the key or with the door already unlocked. These plots show that TEXPLORE-VANIR is acting in an intelligent, curious way, trying actions on the objects in order from the easiest to hardest to access, and going back to the lock repeatedly to learn its more complex dynamics.

Finally, not only should the agent's intrinsic rewards be useful when learning in task without external rewards, they should also make an agent in a domain with external rewards learn more efficiently. For this experiment, the algorithms are run for 3000 steps with their intrinsic rewards added to the previously used external reward function that rewards moving between rooms. Instead of an agent acting randomly, we instead have one agent acting using only the external rewards, and one performing Boltzmann, or soft-max, exploration with temperature $\tau = 0.2$. Figure 5 shows the cumulative external reward received by each agent over the 3000 steps of the task. TEXPLORE-VANIR receives significantly more reward than the other algorithms ($p < 0.001$), followed by R-MAX. Now that exploration and exploitation are no longer separated into separate phases, the exploration of R-MAX is too aggressive and costs it external reward.

These results show that TEXPLORE-VANIR's intrinsic rewards out-perform other exploration approaches and intrinsic motivations combined with the TEXPLORE model. TEXPLORE-VANIR performs similarly to R-MAX when exploration and exploitation are split into separate phases, but out-performs R-MAX significantly when combining intrinsic and external rewards together. TEXPLORE-VANIR explores the domain in a curious manner progressing from state-actions with easier dynamics to those that are more difficult. Finally, in a task with external rewards, TEXPLORE-VANIR can use its intrinsic rewards to speed up learning with respect to an algorithm using only external rewards.

It is important to note that the best intrinsic rewards are dependent on the learning algorithm and the domain. For example, the competence progress rewards used by R-IAC are intended to be used in complex high-dimensional domains where learning is slow. It takes quite a few samples in one region to get an reasonable estimate of the derivative of the error. In the Light World domain, by the time the algorithm has determined error is improving in a region, the agent has already learned a model of that region and no longer needs to explore there. When using other model learning methods, the best intrinsic reward will vary as well, for example, the R-MAX reward works well for a tabular model, but not for a random forest model.

V. Related Work

Many model-based RL algorithms use "exploration bonus" intrinsic rewards to drive the agent to explore more efficiently. As one example, R-MAX [9] uses intrinsic rewards to guarantee that it will learn the optimal policy within a bounded number of steps. The algorithm learns a maximum-likelihood tabular model of the task and provides intrinsic rewards to state-actions that have been visited less than m times. These rewards drive the agent to visit each state-action enough times to learn an accurate model.

A few methods provide intrinsic rewards to an agent to drive it to where its model is improving the most. For example, R-IAC [8] rewards regions where the model error is improving the most. An alternative is to learn a separate predictor of the change in model error and use its predicted values as the intrinsic reward to drive exploration [10].

Simsek and Barto [11] present an approach for the pure exploration problem, where there is no concern with receiving external rewards. They provide a Q-LEARNING agent [12] with intrinsic rewards for where its value function is most improving. This reward speeds up the agent's learning of the true task. However, such a reward is not necessary for model-based agents, which perform value function updates by planning on their model. This algorithm requires an external reward, as the intrinsic reward is speeding up the learning of the task defined by the external reward function.

Singh et al. [13] argue that in nature, intrinsic rewards come from evolution and exist to help us perform any task. Agents using intrinsic rewards combined with external rewards should perform better than those using solely external rewards. For two different algorithms and tasks, they search over a broad set of possible task and agent specific intrinsic rewards and find rewards that make the agent learn faster than if it solely used external rewards.

These different approaches demonstrate that the correct intrinsic motivation is dependent on the type of algorithm. For example, with a Q-LEARNING agent [12], it makes sense to give intrinsic rewards for where the value backups will have the largest effect, as done in [11]. When learning with a tabular model, the agent must gain enough experiences in each state-action to learn an accurate model of it. Thus it makes sense to use intrinsic motivation to drive the agent to acquire these experiences, as done by R-MAX [9]. With a model learning approach that generalizes as TEXPLORE-VANIR's does, the best intrinsic rewards are different again.

VI. Conclusion

This paper presents the TEXPLORE-VANIR algorithm for intrinsically motivated learning, available at `http://www.ros.org/wiki/rl-texplore-ros-pkg`. This algorithm combines random forest based model learning with two novel intrinsic rewards. One reward drives the agent to where the model is uncertain in its predictions, and the second drives the agent to acquire novel experiences that its model has not been trained on. Experiments show empirically that TEXPLORE-VANIR can learn accurate and useful models in a domain with no external rewards. In addition, TEXPLORE-VANIR's intrinsic rewards drive the agent to learn in a developing and curious way, progressing from learning easier to more difficult skills. TEXPLORE-VANIR can also combine its intrinsic rewards with external task rewards to learn a task faster than using external rewards alone. One goal for future work is to extend TEXPLORE-VANIR to work in large continuous state spaces, so that it can apply to some robotic tasks.

References

[1] M. Lopes and P.-Y. Oudeyer, "Guest editorial: Active learning and intrinsically motivated exploration in robots: Advances and challenges," *IEEE Transactions on Autonomous Mental Development (TAMD)*, vol. 2, no. 2, pp. 65–69, 2010.

[2] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[3] T. Hester and P. Stone, "Real time targeted exploration in large domains," in *International Conference on Development and Learning (ICDL)*, August 2010.

[4] T. Hester, M. Quinlan, and P. Stone, "RTMBA: A real-time model-based reinforcement learning architecture for robot control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[5] L. Kocsis and C. Szepesvári, "Bandit based Monte-Carlo planning," in *European Conference on Machine Learning (ECML)*, 2006.

[6] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[7] G. Konidaris and A. G. Barto, "Building portable options: Skill transfer in reinforcement learning," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[8] A. Baranes and P. Y. Oudeyer, "R-IAC: Robust Intrinsically Motivated Exploration and Active Learning," *IEEE Transactions on Autonomous Mental Development (TAMD)*, vol. 1, no. 3, pp. 155–169, Oct. 2009.

[9] R. Brafman and M. Tennenholtz, "R-Max - a general polynomial time algorithm for near-optimal reinforcement learning," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2001.

[10] J. Schmidhuber, "Curious model-building control systems," in *International Joint Conference on Neural Networks*. IEEE, 1991.

[11] Ö. Şimşek and A. G. Barto, "An intrinsic reward mechanism for efficient exploration," in *ICML*, 2006, pp. 833–840.

[12] C. Watkins, "Learning from delayed rewards," Ph.D. dissertation, University of Cambridge, 1989.

[13] S. P. Singh, R. L. Lewis, A. G. Barto, and J. Sorg, "Intrinsically motivated reinforcement learning: An evolutionary perspective," *IEEE Transactions on Autonomous Mental Development (TAMD)*, vol. 2, no. 2, pp. 70–82, 2010.