Vol.2009-MPS-76 No.3
Vol.2009-BIO-19 No.3
2009/12/17

Improving Model-based Reinforcement Learning with Multitask Learning

Jaak Simm,[†1] Masashi Sugiyama[†1]
and Hirotaka Hachiya[†1]

We introduce an extension to standard reinforcement learning setting called observational RL (ORL) where additional observational information is available to the agent. This allows the agent to learn the system dynamics with fewer data samples, which is an essential feature for practical applications of RL methods. We show that ORL can be formulated as a multitask learning problem. A similarity-based and a component-based multitask learning methods are proposed for learning the transition probabilities of the ORL problem. The effectiveness of the proposed methods is evaluated in experiments of grid world and object lifting tasks.

1. Introduction

Recently, there is an increasing interest for methods of planning and learning in unknown and stochastic environments. These methods are investigated in the field of *Reinforcement Learning* (RL) and have been applied to various domains, including robotics[6], AI for computer games, such as tetris[3]. However, one of the main limiting factors for RL methods has been their scalability to large environments, where finding good policies requires too many samples, making most RL methods impractical.

1.1 Transfer Learning in RL

One of the approaches for solving the scalability problem is to reuse the data from similar RL tasks by transferring data or previously found solutions to the new RL task. These methods have been a focus of the research lately and are called *transfer learning* methods. The transfer learning methods can be separated into value-based and model-based transfer learning methods, depending on what is being transferred between the RL tasks.

In value-based transfer learning the value functions of previously solved RL tasks are transferred to the new task at hand. A popular approach for transferring value functions is to use the previously found value functions as initial solutions for the value function of the new RL task. These methods are called starting-point methods, for example see the temporal-difference learning based approach by Tanaka et al.[10] and a comparative study of these methods by Taylor et al.[11]. For successful transfer, a good mapping of states and actions between the RL tasks is required. When a poor mapping is used the transfer can result in worse performance than doing the standard reinforcement learning without a transfer.

On the other hand, model-based transfer learning methods transfer the transition models and reward models from the solved RL tasks to new RL tasks. Similarly to the value-based transfer, the mapping between states and actions of the learned RL tasks and the target RL task is required. However, the requirements for the mapping are weaker than those in the case of value-based transfer and, thus, the transfer is also possible between less similar tasks. The reason is that the transition model and reward model only depend on a single transition from the current state whereas the value function depends on a sequence of rewards (and thus transitions) starting from the current state.

A model-based transfer method was proposed by Wilson et al.[12] that successfully estimates the prior probabilities of tasks. If the model of the new task is similar to previously encountered tasks, the data from the previous tasks can be used to estimate the transition and reward model for the new task. Thus, the new task can be learned with fewer samples.

However, these model-based and value-based transfer approaches still require almost full learning of at least one initial task. That is "previous tasks", which are used in transfer learning of new tasks, should have been learned with sufficient accuracy. If the tasks have large state spaces, then the initial learning will require a huge amount of data, which is not practical. This kind of setting where the tasks

†1 Tokyo Institute of Technology

are ordered is called *transfer learning*. In contrast, *multitask learning* is a setting where there is no initial task and all tasks are solved simultaneously. Another issue with the above reviewed methods is that the advantage of transferring between large RL tasks is problematic because a good mapping between them is not usually available.

1.2 Proposed Observational Idea

To tackle the above mentioned problems we propose a setting where the sharing does not occur between different RL tasks but between different regions (parts) of the same RL task. This is accomplished by allowing the agent to access additional *observational* data about the regions of state-action space of the RL task [⋆1]. The usefulness of the observational data is that it identifies the regions of the task that participate in the multitask learning. Moreover, the strength of the sharing between different regions depends on the similarity of their observations. The more similar the observations are, the stronger the sharing is. This kind of observational data is often available in practice, e.g., in the form of camera data or sensor measurements.

A motivating example for our observational framework is a mobile robot moving around on a ground, where there are two types of ground conditions: slippery and non-slippery. The robot knows its current location and thus, can model the environment using a standard Markov decision formulation, predicting the next location from the current location and the movement action (e.g., forward and backward). However, if the robot has access to additional sensory information about the ground conditions at each state, it could use that additional observation to share the data between similar regions and models of the environment more efficiently even when only a small amount of transition data is available. We call this kind of RL setting *Observational RL*.

In our observational setting there is no order for solving the tasks, meaning that all regions are solved simultaneously, i.e., as a multitask learning setup. Additionally, since the sharing takes place between regions of the whole problem, the mapping is essentially between smaller parts of the problem. Therefore, the problem of finding a good mapping is often mitigated.

⋆1 The idea of separating the RL task into partitions has been explored by Ravindran et al.[7] in the context of hierarchical RL. However, in contrast to the transfer learning setting considered here they assumed that the agent knows a common transition model for all partitions and the goal is to learn how to fit that model to each partition.

This paper extends the study of our proposed Observational RL framework and methods, initially published in[8]. Thus, we only shortly summarize the framework and focus more on new experimental results (Section 5) and discussion (Section 6).

2. Ordinary RL

The goal of reinforcement learning is to learn optimal actions in unknown and stochastic environment. The environment is specified as a Markov Decision Problem (MDP), which is a state-space-based planning problem defined by S, P_I, A, P_T, R and γ. Here S denotes the set of states, $P_I(s)$ defines the initial state probability, A is the set of actions, and $0 \leq \gamma < 1$ is the discount factor. The state transition function $P_T(s'|s,a)$ defines the conditional probability of the next state s' given the current state s and action a. At each step the agent receives rewards defined by function $R(s,a,s') \in \mathbb{R}$. The goal of RL is to find a policy $\pi : S \rightarrow A$ that maximizes the expected discounted sum of future rewards when the transition probabilities P_T and/or the reward function R is unknown.

3. Observational RL

In this section we formulate the setting of Observational RL (ORL). For better understandability, we first start with a simpler framework that already includes the main idea. Then, later extend it to a more general setting.

3.1 Basic Idea

The Observational RL setting extends the ordinary RL setting by allowing the agent to access additional observational information about the state-action space. For the basic case, consider that the agent has observations about each state. This means that for each state $s \in S$ the agent has some observation $o \in O$, where O is the set of observations. Thus, formally the observational

information can be defined as a function $\phi(s) \in O$ mapping each state to its observation. In many cases in practice we can treat some information as this kind of observational data, while the Markov property of the state still holds. For example, in the case of the mobile robot these observations could be sensor measurements about ground conditions and the state is the location of the robot. In other words, the transition probability is determined by the location of the robot without the information about the ground conditions.

The current paper focuses on the model-based RL approach[9], which consists of following two steps:

(1) Estimate the transition probabilities $P_T(s'|s,a)$ using transition data.
(2) Find an optimal policy for the estimated transition model by using a *dynamic programming* method, such as value iteration.

More specifically, the transition data consists of, possibly non-episodic[⋆1], samples $\{(s_t, a_t, s'_t)\}_{t=1}^{T}$, where s_t and a_t correspond to the current state and action of the t-th transition and s'_t is the the next state.

3.2 Formulation of ORL

In the previous formulation the observations were just connected to single states. It is useful to extend the formulation by connecting the observations to regions (i.e., subsets) of the state-action space $S \times A$. Let u denote a region an observation is connected to. We call u an observed region because it is a subset of state-action space $u \subset S \times A$. Thus, the basic ORL idea described above was just a special case when $u \in S$. There are two motivations for this extension. Firstly, it allows us to work with structural problems where one observation is connected to several states, e.g., a manipulation task of various objects by a robotic arm, where an observation is connected to an object, and thus to all states involved in the manipulation of that object. Secondly, this extension means that the observations are now also connected to actions. This allows one to have different observations for different actions and the sharing can depend on actions. For example, in the mobile robot case the movement actions (forward and backward) could participate in the sharing, whereas some other actions, such as picking up an object, could be left out from the sharing.

Now the observations function is $\phi : U \to O$ where U contains all observed regions. If there are N observations then, the observational data is $\{(u_n, o_n)\}_{n=1}^{N}$ where observation $o_n \in O$ corresponds to region $u_n \subset S \times A$. In this case the set of observed regions is $U = \{u_n\}_{n=1}^{N}$.

Compared to the basic idea in Section 3.1 the sharing takes now place between different regions, not just states. Therefore, we require that all observed regions have a common parameterization for their transition models. The transition probabilities of state-action pairs in u, i.e., $(s,a) \in u$, are modeled with $P_T(s'|s,a;\beta_u)$ and β_u is the parameter for the transition model of the region u.

4. Proposed Methods

In this section we give short overview of two proposed multitask-learning-based methods for ORL, for more details please see[8]. First of them is based on the similarity idea and the second one comes from the mixture-of-components multitask learning ideas.

4.1 Similarity-based ORL

The idea of the similarity-based ORL method is to add data from similar tasks directly to the likelihood function of the models for every observed region. Consider the single task estimation of maximum (log) likelihood for observed region u

$$\hat{\beta_u} = \operatorname*{argmax}_{\beta_u} \sum_{(s,a,s') \in D_u} \log P_T(s'|s,a;\beta_u), \tag{1}$$

where D_u is a set of transition data from observed region u. A straightforward extension of the single task estimation (1) is to add data from other tasks and weight them according to the similarity of the other tasks to the current task at hand. This can be expressed by

$$\hat{\beta_u} = \operatorname*{argmax}_{\beta_u} \sum_{v \in U} \sum_{(s,a,s') \in D_v} k_u(v) \log P_T(s'|s,a;\beta_u), \tag{2}$$

⋆1 Non-episodic means that there is no requirement that the next state of the t-th transition sample (i.e., s'_t) has to be equal to the starting state of the $(t+1)$-th transition sample (i.e., s_{n+1}).

where $k_u(v) \in [0,1]$ is the similarity of the observed region v to observed region u. Thus, data from observed regions that have high similarity $k_u(v)$ have a big effect on the estimation of the model of region u. In the case of a mobile robot, consider the estimation of the model for a region of slippery states u (e.g., an icy region). If the similarity function k_u assigns high similarity to other regions of slippery states (e.g., other icy regions or wet regions) and a low similarity value for non-slippery states then the similarity-based ORL method will provide an accurate estimate for β_u even if region u has few or no samples. A practical option for the similarity function is to use the Gaussian kernel between the observations of the regions or nearest neighbor similarity.

4.2 Component-based ORL

Here we introduce the idea of component-based multitask learning where the role of task features is to a priori determine the component the task belongs to. Let there be M components, then $P(m|\phi(u))$ denotes the probability that the task u with features $\phi(u)$ belongs to the component m (where $m \in \{1,\ldots,M\}$).

Let (s,a) be a state-action pair and $u \in U$ be such that $(s,a) \in u$, then the sharing between elements of U is formulated as a mixture of components for the transition probability:

$$P_T(s'|s,a) = \sum_{m=1}^{M} P_T(s'|s,a,m)P(m|\phi(u)), \tag{3}$$

where $P_T(s'|s,a,m)$ is the transition probability to state s' under component m for state-action pair (s,a) and $P(m|\phi(u))$ is the component membership probability mentioned above. In the example of a mobile robot, these components would comprise of states that have similar transition dynamics, e.g., one component could be a group of states where a certain moving action fails due to difficult ground conditions and another component represents states where the moving action succeeds.

The parameterized version of (3) is given by

$$P_T(s'|s,a,\beta,\alpha) = \sum_{m=1}^{M} P_T(s'|s,a,\beta_m)P(m|\phi(u),\alpha), \tag{4}$$

where β_m is the parameter for the transition model of component m and α is the parameter for the component membership probabilities. The estimates of both of these parameters will be determined by maximum likelihood estimation by employing EM method. It should be noted that any parameterization will work as long as the maximum likelihood estimation is computationally tractable. The choice of parameterization for $P(m|\phi(u),\alpha)$ depends on the type of observations, O. For details please see[8]. We follow standard approach for implementing the EM method. This includes using several restarts to the EM procedure to avoid local optima and using cross-validation to choose the number of components (M).

5. Experimental Results

In this section we present experimental results from two simulated domains: grid world with slippery ground conditions and a robot's object lifting tasks.

5.1 Slippery Grid World

We conducted experiments on a mobile robot task with discrete state and action space. The size of the state space of the grid world is 15×15 and there are 4 movement actions: left, right, up and down. There are two types of states, one type is slippery, where all movement actions fail with probability 0.8, keeping the robot at the same spot and the other type is non-slippery having probability of failure 0.15. The goal of the agent is to reach the goal state from the initial state. An example of the grid world is shown in Figure 1. The goal of the robot is to reach the goal state denoted with "G" starting from bottom left state "S". White squares are non-slippery and colored squares are slippery states. The observations about each state are two-dimensional real values of sensor measurements of water level and amount of loose gravel. Their distribution is depicted in Figure 2.

Due to the rather high rate of failure of actions in slippery states, the robot should avoid these states and thus, it is essential to accurately estimate the transition probabilities. The average performance over 50 runs for component-based and similarity-based ORL methods is reported in Table 1. Methods named 'Comp(n)' are component-based methods with n components. Thus, 'Comp(1)' actually just merges all observed regions as a unified task. For component-based methods, 'Comp(2)' and 'Comp(3)', we manually chose the regularization parameter of the logistic regression to be 10^{-3}. For similarity-based method 'Sim(fixed)'

Vol.2009-MPS-76 No.3
Vol.2009-BIO-19 No.3
2009/12/17

Table 1: KL-divergence of the estimated transition probabilities from the true model, for the slippery grid world experiment with *2-dimensional* observations. For each method the mean and standard deviation of its KL-divergence averaged over 50 runs are reported, for different data sizes $N = 50$, $N = 100$, $N = 150$, and $N = 200$. Bolded values in each column show methods whose performance is better than others using t-test with 1% confidence level.

Method	$N = 50$	$N = 100$	$N = 150$	$N = 200$
Comp(1)	0.375 ± 0.065	0.280 ± 0.023	0.255 ± 0.012	0.244 ± 0.013
Comp(2)	0.373 ± 0.102	$\mathbf{0.177 \pm 0.036}$	$\mathbf{0.117 \pm 0.034}$	$\mathbf{0.080 \pm 0.034}$
Comp(3)	0.422 ± 0.123	0.235 ± 0.069	0.164 ± 0.051	0.123 ± 0.045
Comp(CV)	$\mathbf{0.322 \pm 0.053}$	0.190 ± 0.051	$\mathbf{0.127 \pm 0.032}$	$\mathbf{0.094 \pm 0.035}$
Sim(fixed)	0.369 ± 0.046	0.207 ± 0.022	0.153 ± 0.015	0.125 ± 0.010
Sim(CV)	$\mathbf{0.338 \pm 0.028}$	0.211 ± 0.023	0.162 ± 0.021	0.132 ± 0.014
Single task	1.686 ± 0.004	1.628 ± 0.006	1.576 ± 0.008	1.526 ± 0.009

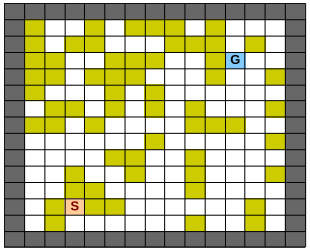


Fig. 1: Mobile robot in a grid-world with slippery and non-slippery states. Robot starts from an initial state at bottom left denoted with "S" and has to reach the goal state "G".

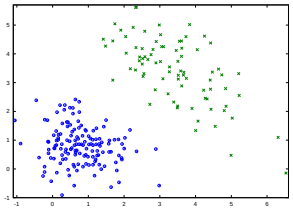


Fig. 2: Distribution of observations for non-slippery (blue circles) and slippery (green crosses) states. The horizontal axis displays the measured water level and the vertical axis displays the measured amount of loose gravel for each state.

the Gaussian kernel with a fixed width $\sigma = 2.5$ was used. The 'Comp(CV)' is the component-based ORL that uses 5-fold CV to choose the regularization parameter for logistic regression from the set $\{10^{-3}, 10^{-1}, 10^{0}\}$ and the number of components. Similarly, 'Sim(CV)' is the similarity-based ORL that uses 5-fold CV to choose the optimal width for the Gaussian kernel from the set $\{1.5, 3.0, 4.5, 6.0, 10.0\}$.

With 50 samples none of the other ORL methods perform better than 'Comp(1)' (unified task), suggesting that there are too few samples to successfully perform data sharing. However, all ORL methods outperform the 'Single task' implying that the use of data sharing in this case is valuable. As seen from Table 1 the cross-validation version of component-based method 'Comp(CV)' is performing almost as well as the best fixed parameter version.

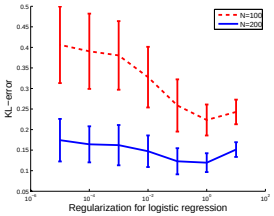
5.2 Grid World with High-dimensional Observations

We also tested the grid world example width high-dimensional observations. Now the observations were 10-dimensional. The first two dimensions were exactly the same as before, containing useful information about the states as depicted in Figure 2. The new 8 dimensions did not contain any information, i.e., the observations for slippery and non-slippery states were generated from the same distribution, which was a single 8-dimensional Gaussian with mean zero and covariance identity.

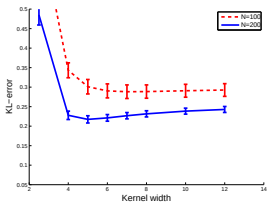
Table 2: KL-divergence of the estimated transition probability from the true model, for the slippery grid world experiment with *10-dimensional* observations. For each method the mean and standard deviation of its KL-divergence averaged over 50 runs are reported, for different data sizes $N = 50$, $N = 100$, $N = 150$, and $N = 200$. Bolded values in each column show methods whose performance is better than others using t-test with 1% confidence level.

Method	$N = 50$	$N = 100$	$N = 150$	$N = 200$
Comp(CV)	$\mathbf{0.395 \pm 0.085}$	$\mathbf{0.248 \pm 0.044}$	$\mathbf{0.190 \pm 0.054}$	$\mathbf{0.140 \pm 0.039}$
Sim(CV)	$\mathbf{0.398 \pm 0.047}$	0.285 ± 0.014	0.244 ± 0.014	0.222 ± 0.012
Comp(1)	$\mathbf{0.375 \pm 0.065}$	0.280 ± 0.023	0.255 ± 0.012	0.244 ± 0.013
Single task	1.686 ± 0.004	1.628 ± 0.006	1.576 ± 0.008	1.526 ± 0.009

Comparing Table 2 to Table 1 we can see that the performance of ORL methods is degraded compared to the problem with low-dimensional9 observation. As expected, the performance of the similarity-based approach, 'Sim(CV)', has worsened more than the performance of the component-based approach, 'Comp(CV)'. The effect the choice of the parameters has on the performance of both methods is depicted in Figure 3. As seen from Figure 3(a), too weak regularization for logistic regression results in poorer performance. Similarly, too small kernel width for the similarity-based ORL method has poor performance.



(a) Dependence of the performance of the component-based ORL on the regularization of logistic regression.



(b) Dependence of the performance of the similarity-based ORL on Gaussian width.

Fig. 3: Average KL-divergence from the true distribution in slippery grid world tasks with 10-dimensional observations for sample sizes $N = 100$ and $N = 200$. The averages and standard deviations were calculated from 50 runs.

Table 3: Value of the the policy found by using the estimated transition probabilities, for the slippery grid world experiment with 10-dimensional observations. For each method the mean and standard deviation of its value averaged over 50 runs are reported, for different data sizes $N = 50$, $N = 100$, $N = 150$, and $N = 200$. Bolded values in each column show methods whose performance is better than others using t-test with 1% confidence level.

Method	$N = 50$	$N = 100$	$N = 150$	$N = 200$
Comp(CV)	$\mathbf{0.648 \pm 0.101}$	$\mathbf{0.699 \pm 0.048}$	$\mathbf{0.724 \pm 0.043}$	$\mathbf{0.740 \pm 0.027}$
Sim(CV)	$\mathbf{0.659 \pm 0.014}$	0.667 ± 0.020	0.705 ± 0.033	0.727 ± 0.024
Comp(1)	0.639 ± 0.011	0.649 ± 0.005	0.652 ± 0.002	0.651 ± 0.001
Single task	-0.508 ± 0.121	-0.380 ± 0.177	-0.279 ± 0.192	-0.135 ± 0.201

Table 3 shows the value of the policies that were found from the transition probabilities learned by different methods for high-dimensional observations case. The component-based method outperforms similarity-based method for sample sizes 100, 150 and 200, but only slightly. Compared to the differences in the KL-error the similarity-based method would be expected to perform worse. It is probable that for the slippery grid world task the similarity-based method captures some important differences in the transition probabilities resulting in similar performance to the component-based method.

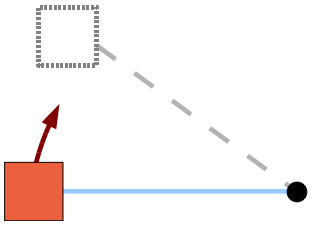


Fig. 4: Single motor robotic arm lifting an object to a goal angle using torque control. The object depicted with solid line shows initial position and the dashed line shows the goal state.

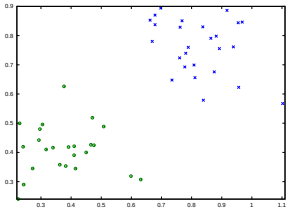


Fig. 5: Distribution of observations for light (green circles) and heavy (blue crosses) objects. The horizontal axis depicts the size of the object and the vertical axis depicts the color of the object (higher values corresponding to red and small values corresponding to blue color objects).

5.3 Results of Object Lifting Task

The goal for the task of object lifting by a single joint robotic arm is to lift an object from the starting position to the goal angle of 36 degrees, as depicted in Figure 4. The task involves 50 objects, randomly chosen at the start of the task. Half of the objects are light and the other half are heavy, having different state transition dynamics. Thus, ideally the sharing should happen only between the objects of the same type. Each object has 3-dimensional observation, consisting of object size, color and texture. The texture of the object does not contain any information regarding the object and is randomly generated from a single Gaussian distribution. The size and the color of objects contain some information regarding the object's nature. These two features are depicted in Figure 5 for the 50 objects.

The state vector consists of object number (from 1 to 50), joint angle and angle speed of the arm. The latter two are discretized, joint angle to 21 values of $0, 3, \ldots, 60$ degrees, and angle speed to 9 values of $0, \pm 3, \pm 7.5, \pm 14.25, \pm 24.4$ degrees per step. The transition model is based on simulated frictionless dynamics with gravitational force. If the simulation ends in a state between discretized values, the transition models assigns probabilities to the nearest discretized states

according to the distance (the closer the state the higher the transition probability). The reward function gives +1.0 for states with the goal joint angle (i.e., 36 degrees) and +0.5 for states that have joint angle next to the goal state, i.e., 33 and 39 degrees; all other states give zero rewards. The discount factor of 0.95 is used, combined with the maximum possible reward it implies that the expected value of a policy cannot be larger than 20. The arm is controlled by 9 actions corresponding to different torques applied to the joint. Thus, the transition dynamics for heavy and light objects are quite different and thus, also the optimal policy is different. There are totally 1701 state action pairs for one object, making totally 85050 state-action pairs for 50 objects.

Table 4 shows the accuracy of the transition model estimation for the two ORL methods ('Comp(CV)' and 'Sim(CV)'), unified ('Comp(1)') and single task ('Single task') methods. The data was collected uniformly over the state-action space. As can be seen the component-based ORL method significantly outperforms all other methods for sample sizes 16000 and above. The poor performance of the similarity-based method is due to the irrelevant element in observations (i.e., texture) and relatively few number of tasks (i.e., 50) to the number of data collected (8000 and more samples).

Similarly to transition model estimation results, the value of the learned policies shows a good performance for the component-based method, see Table 5. The expected value of the optimal policy for the object lifting task is 14.442 and with 32000 samples the component-based method achieves quite close value of 13.3. In contrast to the mobile robot task, the similarity-based method only slightly outperforms the unified task (significantly for 24000 and 32000 samples sizes, with confidence level 1%). However, the difference is minimal to provide big gains in the case of object lifting task. Nevertheless, both ORL methods and the unified task strongly outperform the single task method.

5.4 Summary of Experiments

The two proposed ORL methods showed clear advantages over the no sharing (i.e., single task) approach in the slippery grid world and the object lifting tasks. If observations are noisy, then the component-based method performs better than the similarity-based method if enough samples are available. In other words, the performance of the similarity-based method depends on the existence of a good similarity function, whereas the component-based method is able to learn the task similarities from data.

Table 4: KL-divergence of the estimated transition probability from the true model, for the experiment of object lifting by robotic arm. For each method the mean and standard deviation of its KL-divergence averaged over 50 runs are reported, for different data sizes $N = 8000$, $N = 16000$, $N = 24000$, and $N = 32000$. Bolded values in each column show methods whose performance is better than others using t-test with 1% confidence level.

Method	$N = 8000$	$N = 16000$	$N = 24000$	$N = 32000$
Comp(CV)	$\mathbf{1.754 \pm 0.023}$	$\mathbf{0.944 \pm 0.031}$	$\mathbf{0.616 \pm 0.011}$	$\mathbf{0.447 \pm 0.005}$
Sim(CV)	$\mathbf{1.754 \pm 0.023}$	1.153 ± 0.016	0.905 ± 0.010	0.772 ± 0.006
Comp(1)	$\mathbf{1.754 \pm 0.023}$	1.156 ± 0.016	0.916 ± 0.010	0.790 ± 0.006
Single task	4.357 ± 0.002	4.161 ± 0.004	3.975 ± 0.006	3.800 ± 0.006

Table 5: The expected value of the the policy found by using the estimated transition probabilities, for the experiment of object lifting. For each method the mean and standard deviation of the value of the policy averaged over 50 runs are reported, for different data sizes $N = 8000$, $N = 16000$, $N = 24000$, and $N = 32000$. Bolded values in each column show methods whose performance is better than others using t-test with 1% confidence level.

Method	$N = 8000$	$N = 16000$	$N = 24000$	$N = 32000$
Comp(CV)	$\mathbf{6.110 \pm 0.987}$	$\mathbf{12.203 \pm 0.592}$	$\mathbf{12.850 \pm 0.451}$	$\mathbf{13.319 \pm 0.434}$
Sim(CV)	$\mathbf{6.229 \pm 0.867}$	7.292 ± 0.884	7.961 ± 0.689	8.402 ± 0.537
Comp(1)	$\mathbf{6.110 \pm 0.987}$	6.953 ± 0.765	7.331 ± 0.918	7.721 ± 0.503
Single task	1.172 ± 0.084	1.830 ± 0.088	2.691 ± 0.141	3.533 ± 0.129

6. Discussion and Conclusion

We proposed the ORL framework for taking advantage of additional observational data to share transition data for the learning of the transition probabilities. The two proposed methods for ORL show good performance in experiments, obtaining more accurate estimates for the transition probabilities and consequently resulting in better policies.

Up to now we only considered situation where reward function $R(s, a, s')$ was

known by the agent. Similarly to the transition probabilities it is possible to use the ORL framework to speed up the learning of the reward function. For that we need to assume a common parameterization for the probabilities of the rewards for all regions, i.e., for region u the conditional probability density function of rewards is $P(r|s,a,s';\omega_u)$, where $(s,a) \in u$, $r \in \mathbb{R}$ is reward and ω_u is the parameter for the reward function.

If the maximum likelihood estimation for $P(r|s,a,s';\omega_u)$ is computationally tractable we can use both proposed ORL methods for estimating the parameters for the reward function. The performance and the properties of the two methods for reward function estimation are expected to be similar to those of transition probability estimation.

In the current paper we did not discuss the topic of exploration in relation to ORL. One future research topic is to investigate whether Rmax[2] style exploration-exploitation can be introduced for ORL. Although the standard exploration policies based on Rmax can be used in ORL they are too conservative because they require visiting each state and action pair many times. Efficient exploration should avoid exploring state-actions where data is available from other regions. For efficient exploration in ORL setting we could use recently introduced approximate Bayesian exploration methods, proposed by Kolter and Ng[5] and by Asmuth et al.[1], both having polynomial resource guarantees. However, to use these two Bayesian exploration methods we need to extend our methods to allow sampling the posterior of the transition probabilities, instead of just providing the maximum likelihood estimate.

Another possible research area is extending ORL to factored MDPs[4] by allowing observations to be linked to specific state factors and, thus, allowing us to share data between similar factors. For example, the slippery grid world with multiple mobile robots is a setting where sharing information between different factors, i.e., different robots, is useful. In this example the sharing between different factors means sharing data between similar robots that are in similar locations. Thus, in addition to the information about state-action regions (i.e., locations) the observations should include information about the factors (i.e., robots).

References

1) John Asmuth, Lihong Li, MichaelL. Littman, Ali Nouri, and David Wingate. A Bayesian sampling approach to exploration in reinforcement learning. In *Proceedings of The 25th Conference on Uncertainty in Artificial Intelligence (UAI-09)*, June 2009.
2) RonenI. Brafman and Moshe Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231, 2003.
3) Alexander Gro, Jan Friedland, and Friedhelm Schwenker. Learning to play tetris applying reinforcement learning methods. In *ESANN*, pages 131–136, 2008.
4) MichaelJ. Kearns and Daphne Koller. Efficient reinforcement learning in factored MDPs. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 740–747, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.
5) J.Zico Kolter and AndrewY. Ng. Near-Bayesian exploration in polynomial time. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 513–520, New York, NY, USA, 2009. ACM.
6) J. Peters, S. Vijayakumar, and S. Schaal. Reinforcement learning for humanoid robotics. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, Karlsruhe, Germany, September 2003.
7) Balaraman Ravindran and AndrewG. Barto. Relativized options: Choosing the right transformation. In *Proceedings of the Twentieth Internatinal Conference on Machine Learning (ICML 2003)*, pages 608–615, Menlo Park, CA, August 2003. AAAI Press.
8) J.Simm, M.Sugiyama, and H.Hachiya. Observational reinforcement learning. In *Technical Report on Information-Based Induction Sciences 2009 (IBIS2009)*, pages 120–127, October 2009.
9) RichardS. Sutton and AndrewG. Barto. *Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1998.
10) Fumihide Tanaka and Masayuki Yamamura. Multitask reinforcement learning on the distribution of MDPs. In *Computational Intelligence in Robotics and Automation, 2003*, volume3, pages 1108–1113, July 2003.
11) MatthewE. Taylor, Peter Stone, and Yaxin Liu. Value functions for RL-based behavior transfer: A comparative study. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*, pages 880–885, July 2005.
12) Aaron Wilson, Alan Fern, Soumya Ray, and Prasad Tadepalli. Multi-task reinforcement learning: a hierarchical Bayesian approach. In *ICML '07: Proceedings of the 24th International Conference on Machine learning*, pages 1015–1022, New York, NY, USA, 2007. ACM.