情報論的学習理論テクニカルレポート 2009
Technical Report on Information-Based Induction Sciences 2009 (IBIS2009)

Observational Reinforcement Learning

Jaak Simm, Masashi Sugiyama, and Hirotaka Hachiya*

Abstract: 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.

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, AI for computer games, such as tetris, racing games and fighting games. 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 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 and Yamamura [4] and a comparative study of these methods by Taylor et al. [5]. 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. This difference can be seen from an example of transferring knowledge from a previous task where the agent is able to obtain a big positive reward after opening the door and moving around in the room behind the door. However, if the new RL task gives a negative reward after the agent enters the room, the value-based transfer is not useful, probably even worsening the performance. On the other hand, model-based transfer could transfer the knowledge that the opening of the door allows to enter the room and if the agent has already learned that the room contain negative rewards in the new task it can infer the negative value of the actions that open the door and enter that room. In summary, the advantage of model-based transfer over value-based transfer is in cases where actions in different tasks have similar results, e.g., the same action opens the door, but the value of the action is different between the tasks.

A model-based transfer method called was proposed by Wilson et al. [6] that successfully estimates the probabilistic prior 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. A similar approach has also been applied to partially observable environments [2].

However, these model-based and value-based transfer approaches still require almost full learning of at

*Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan 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 realistic. This kind of setting where the tasks 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 usually not 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. 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.

In our proposed setting, the model-based sharing is more natural than the value-based sharing, as the value of the states often depends on the global location of the region, and thus the value of similar regions is not expected to be same. In the mobile robot example described above, the probabilities of moving forward would be similar in locations with similar ground conditions, but the value of going forward in these locations depends on where the robot makes a transition to after executing the forward action. For this reason, from here we only focus on the model-based multitask learning in the setting of ORL.

1.3 Outline

In the next section we formally introduce the setting of ordinary RL. The notions of observations and similarity will be formalized in Section 3. After that we propose two methods for solving the Observational RL problem in Section 4. Their performance is evaluated experimentally in Section 5. Finally, we conclude in 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 $\pi : S \to A$ that maximizes the expected discounted sum of future rewards when the transition probabilities P_T and/or the reward function R is unknown. The discounted sum of future rewards is $\sum_{t=0}^{\infty} \gamma^t r_t$, where r_t is the reward received at step t. In this paper we focus on the case where the transition probabilities are unknown, but the reward function is known, due to space constraints. The extension of the proposed methods to an unknown reward function is straight-forward.

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 [1]. This means that for each

[1] Observations are separated from the state information because they do not necessarily satisfy the Markov property.

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. For example, in the case of the mobile robot these observations could be sensor measurements about ground conditions at each location.

The general idea of ORL is to use these additional observations for speeding up the learning, thus, requiring fewer samples to find good policies. ORL will be effective if the states that have similar observations have similar transition structure. If the transition structure has nothing in common applying ORL-based methods will not be able to improve the performance. On the other hand, if similar observations imply similar transition structure, then ORL-methods should have strong advantages.

The current paper focuses on the model-based RL approach [3], 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[2], 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. Thus, the idea is to use observational data expressed by ϕ to have more accurate estimates of the transition probabilities P_T.

To take advantage of observational information we have to require that the agent assumes a common parameterization for the transition models for all states. In other words, transition probabilities for all states are modelled with the same parametric form $P_T(s'|s,a;\beta_s)$ where β_s is the parameter for the transition model for state s. For example, in the case of discrete MDPs, we can use a multinomial parameterization. This common parameterization implicitly defines the mapping between the actions and next states of different states. Thus, it is similar to the mappings used in other transfer learning methods discussed in Section 1.1.

Similarly to other transfer learning methods the choice of mapping (in the case of ORL the choice of the parameterization) greatly affects the performance. Use of improper parameterization will negate all advantages of data sharing and could even worsen the performance, depending on what method is used for solving the ORL problem.

Next we formalize the ORL framework that extends the described basic idea.

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 and as 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 $\phi : U \to O$ 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 $U = \{u_n\}_{n=1}^{N}$.

Additionally, we require that states can belong at most to a single observed region, this means that $u_i \cap u_j = \emptyset$, for $i \neq j$. However, there is no requirement that all state-action pairs belong to an observed region. The state-action pairs that do not belong to any observed region do not benefit from the observational information. This extension allows the agent to consider models where all regions of the state-action space are not equipped with observations or certain parts of the state space are different, e.g., there is a maze with corridors and rooms and the agent only has observations about the rooms.

Next we propose two methods for solving the ORL problem.

4 Proposed Methods

First of the methods 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 similarity-based ORL method is to add data from similar tasks directly to the likelihood function of the models for every observed region. Consider

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

the single task estimation of maximum (log) likelihood for observed region u

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

where D_u is a set of transition data from observed region u. A straight-forward 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

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

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 just use the Gaussian kernel between the observations of the regions, expressed as

$$k_u(v) = \exp(-\|\phi(u) - \phi(v)\|^2/\sigma^2), \qquad (3)$$

where σ is the width of the kernel. This parameter could be chosen using cross-validation and it controls how much multitask effect distant tasks have on the current task at hand.

The only constraint on k_u is that it should give value 1 for the region itself, i.e., $k_u(u) = 1$. No other properties are required. Thus, we also allow non-symmetric and non-positive definite similarities.

One disadvantage of similarity-based ORL is it suffers from the curse of dimensionality if the observations are high-dimensional. In this case it means that all tasks will become dissimilar to each other due to the high-dimensionality of observations. Therefore, next we will introduce a more sophisticated method that is based on mixture-of-components, which uses a probabilistic framework to model the multitasking problem of ORL and thus could be expected to mitigate the above mentioned problem.

4.2 Component-based ORL

In this section we introduce a component-based multitask learning method for learning transition probabilities $P_T(s'|s,a)$ for ORL.

Consider again the example of a mobile robot that is moving along a difficult terrain that has obstacles and varying ground conditions. The robot knows its location and speed at each step. That knowledge allows the robot to learn the state transition probabilities for each action. However, if the robot has access to additional observations about the states (using sensors or a camera), then using such observational information could allow the robot to estimate the transition probabilities in fewer samples than by just using robots location and speed.

Recall that in ORL the agent has access to observations, i.e., the agent knows function $\phi(u) \in O$. For example, for the mobile robot the set of observations could contain measurements about the ground type (e.g., gravel or tarmac) or visual information about the obstacles around a particular location. As already mentioned, in terms of multitask learning an observed region $u \in U$ is a task and $\phi(u)$ specifies its features.

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 $m \in \{1,\dots,M\}$).

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:

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

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.

Given the number of components M and data about transitions and observations, we want to find the maximum likelihood estimate for (4). To do that we first need to assume a parametric form for its elements. The parameterized version of (4) is given by

$$P(s'|s,a,\beta,\alpha) = \sum_{m=1}^{M} P(s'|s,a,\beta_m)P(m|\phi(u),\alpha), \qquad (5)$$

where β_m is the parameter for the transition model of component m and α is the parameter for component membership probabilities. The estimates of both of these parameters will be determined by maximum likelihood estimation. It should be noted that any parameterization will work as long as its maximum likelihood estimation is tractable. The choice of parameterization for $P(m|\phi(u), \alpha)$ depends on the type of observations, O. For discrete observations an option is to use a Naive Bayes model:

$$P(m|o, \alpha) = \alpha_{m,0} \prod_{k=1}^{K} \alpha_{m,k,o_k}, \tag{6}$$

where o is observation, i.e., $o = \phi(u) = (o_1, \ldots, o_K)^T$. Parameter $\alpha_{m,0}$ controls the overall probability of component m and α_{m,k,o_k} controls the probability of component m for regions whose observation's k-th dimension is equal to o_k. Since parameters are multiplied together, the model assumes that each dimension independently affects the component probability.

For continuous observations following parameterization can be used:

$$P(m|\phi(u), \alpha) = \frac{\exp(\langle \alpha_m, \phi(u) \rangle)}{\sum_{\bar{m}=1} \exp(\langle \alpha_{\bar{m}}, \phi(u) \rangle)}, \tag{7}$$

where $\phi(u)$ denotes the observation for u, $\alpha_m \in \mathbb{R}^K$, i.e., observations are K-dimensional real values, and $\langle \cdot, \cdot \rangle$ is inner product. This parameterization corresponds to multi-class logistic regression problem.

Because of its complicated form, the maximum likelihood estimate for (5) cannot be found using straightforward optimization. A standard approach doing maximum likelihood estimation on such problems is to use an EM-based method [1]. To do that we introduce a latent indicator variable

$$\mathbf{z} \in \{0, 1\}^M, \tag{8}$$

which denotes the true component for u. Thus, only one of the elements of $\mathbf{z}$ is equal to one and all others are equal to zero. Using $\mathbf{z}$ we can rewrite the mixture (5) as

$$\begin{aligned} & P(s', \mathbf{z}|s, a, \beta, \alpha) \\ & = \sum_{m=1}^{M} z_m P(s'|s, a, \beta_m) P(m|\phi(u), \alpha) \qquad (9) \\ & = \prod_{m=1}^{M} \left[P(s'|s, a, \beta_m) P(m|\phi(u), \alpha) \right]^{z_m}, \qquad (10) \end{aligned}$$

where the summation form is transformed into a product form, which allows us to easily handle the log likelihood. This latent variable formulation allows us to use the EM algorithm for determining a maximum likelihood solution for β and α.

The outline of the EM-method is

1. Start with initial values for parameters β and α.

2. Calculate the posterior probabilities of the latent variables, given the parameters β and α (E-step).

表 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 0.1% confidence level.

Method	$N = 50$	$N = 100$
Comp(1)	0.375 ± 0.065	0.280 ± 0.023
Comp(2)	0.373 ± 0.102	$\mathbf{0.177 \pm 0.036}$
Comp(3)	0.422 ± 0.123	0.235 ± 0.069
Comp(CV)	$\mathbf{0.322 \pm 0.053}$	0.190 ± 0.051
Sim(fixed)	0.369 ± 0.046	0.207 ± 0.022
Sim(CV)	$\mathbf{0.338 \pm 0.028}$	0.211 ± 0.023
Single task	1.686 ± 0.004	1.628 ± 0.006

(a) $N = 50$ and $N = 100$

Method	$N = 150$	$N = 200$
Comp(1)	0.255 ± 0.012	0.244 ± 0.013
Comp(2)	$\mathbf{0.117 \pm 0.034}$	$\mathbf{0.080 \pm 0.034}$
Comp(3)	0.164 ± 0.051	0.123 ± 0.045
Comp(CV)	$\mathbf{0.127 \pm 0.032}$	$\mathbf{0.094 \pm 0.035}$
Sim(fixed)	0.153 ± 0.015	0.125 ± 0.010
Sim(CV)	0.162 ± 0.021	0.132 ± 0.014
Single task	1.576 ± 0.008	1.526 ± 0.009

(b) $N = 150$ and $N = 200$

3. Find β and α that maximize the expectation of the regularized data likelihood (M-step).

4. If the solution has converged stop, otherwise go to step 2.

Due to space restriction we leave out the details of E-step and M-step and only present the conclusions. E-step can be performed analytically by just applying the Bayes law. The M-step for transition models can be performed analytically for discrete and Gaussian models and M-step for observation-based component membership parameter α can be effectively computed by convex optimization based methods.

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.

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. If the robot moves at the edge squares it receives a negative reward of -1 and is reset to the starting state. The robot receives reward $+1$ when it reaches goal state, after which it is again reset to the initial position. Other states do not give any reward.

The observations about each state are two-dimensional real values of sensor measurements. The first dimension shows the measurement of the depth of the water layer covering the ground at that location and the second dimension the amount of loose gravel. Both measurements are noisy and for the experiments are generated randomly from two Gaussian distributions, one for slippery states and another for non-slippery states. The two Gaussians are quite separated, as can be seen from Figure 2.

The average performance over 50 runs for the component-based and the similarity-based ORL methods is reported in Table 1. The table reports average KL-divergence values of the estimated transition probabilities from the true transition probabilities. Methods named 'Comp(n)' are component-based methods with n components. Thus, 'Comp(1)' actually just merges all observed regions as a unified task. The results in Table 1 use transition data that is collected uniformly over the state and action space, this allows us to compare the pure performance of different methods without the side effects of non-uniform data collecting policy. Secondly, in this experiment the methods used manually-chosen parameters to show the performance of the methods without the problem of choosing optimal parameters. 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)' the Gaussian kernel with a fixed width $\sigma = 2.5$ was used. The single task method that does not use observations and 'Comp(1)' do not have any extra parameters.

Table 1 also reports the performance of methods using cross-validation(CV) for the choice of the parameters. 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 $\{1.5, 3.0, 4.5, 6.0, 10.0\}$.

Firstly, we can use the performance of the unified task ('Comp(1)') as a good comparison point in Table 1, because unifying all tasks is not expected to provide good results when a large number of samples are available. All ORL methods outperform the 'Single task' implying that the use of data sharing in this case is valuable, even with just 50 samples. As expected the component-based method using 2 components 'Comp(2)' is performing the best overall with 100 or more samples. The performance of 'Comp(3)' and 'Sim(fixed)' is slightly worse than 'Comp(2)', but still clearly outperforming the unified task and single task methods, validating their usefulness in this experiment.

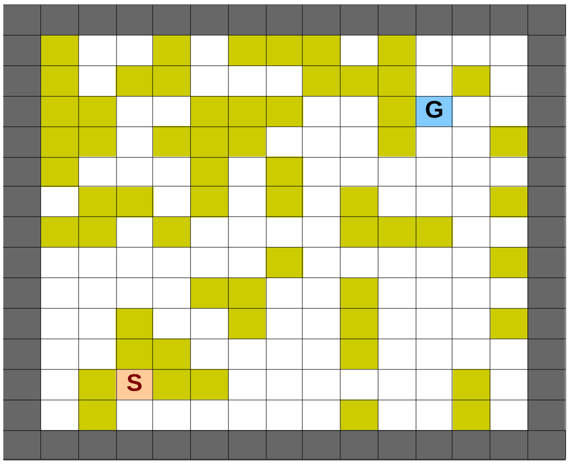


図 1: Mobile robot in a grid-world with slippery and non-slippery states. Robot starts from an initial state at bottom left denoted with "S" and has to reach the goal state "G".

Also, 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. Actually in the case $N = 50$ the CV method is outperforming the fixed methods, because the regularization that was used in the fixed cases (10^{-3}) is too small, resulting in poor performance of the EM-based method, if only 50 samples are available. The effect of the regularization of logistic regression is depicted in Figure 3(a) for sample sizes 100 and 200. For both sample sizes if the regularization is not too big the component-based ORL has good performance.

Similarly, the 'Sim(CV)' method is very close to the fixed width case and the performance of similarity-based ORL is not very sensitive to the chosen Gaussian widths unless a too small width is chosen as seen from Figure 3(b). These results suggest that CV can be used for tuning the parameters of component-based and similarity-based ORL.

Table 2 shows the value of the policies that were found from the transition probabilities learned by different methods. The two ORL methods have similar performance and obtain significantly higher value than unified task ('Comp(1)') and single task. They

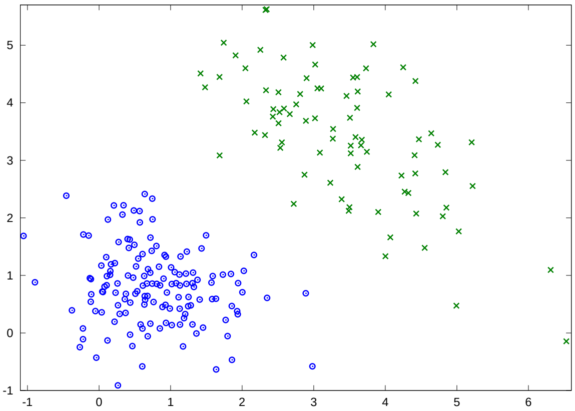
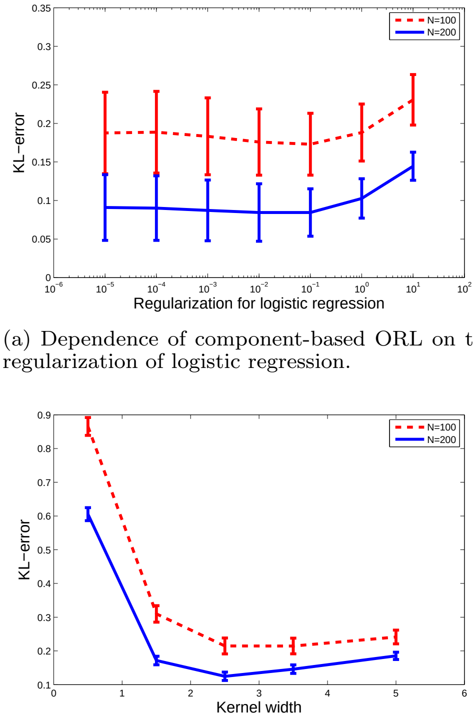


図 2: Distribution of observations for non-slippery (blue circles) and slippery (green crosses) states. The horizontal axis displays the measured water level and the vertical axis displays the measured amount of loose gravel for each state.



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

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

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

are quite close to the value of optimal policy, which is 0.799 in this task. The good performance of 'Comp(1)' is explained be the fact that in their nature the slippery and non-slippery states are similar, because all 4 actions result in similar outcomes, just the probabilities of these outcomes differ.

表 2: Value of the the policy found by using the estimated transition probabilities, for the slippery grid world experiment with *2-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 0.1% confidence level.

Method	$N = 50$	$N = 100$
Comp(CV)	$\mathbf{0.716 \pm 0.054}$	$\mathbf{0.746 \pm 0.023}$
Sim(CV)	$\mathbf{0.715 \pm 0.036}$	$\mathbf{0.749 \pm 0.021}$
Comp(1)	0.638 ± 0.009	0.649 ± 0.006
Single task	-0.503 ± 0.117	-0.370 ± 0.179

(a) $N = 50$ and $N = 100$

Method	$N = 150$	$N = 200$
Comp(CV)	$\mathbf{0.754 \pm 0.016}$	$\mathbf{0.757 \pm 0.018}$
Sim(CV)	$\mathbf{0.756 \pm 0.005}$	$\mathbf{0.757 \pm 0.004}$
Comp(1)	0.652 ± 0.002	0.651 ± 0.001
Single task	-0.237 ± 0.196	-0.102 ± 0.179

(b) $N = 150$ and $N = 200$

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 zero mean and identity covariance.

The results of high-dimensional grid world experiments for component-based and similarity-based ORL methods with CV are shown in Table 3. The sets of model parameters used by CV are the same as in the previous experiment. For comparison the results for 'Comp(1)' and 'Single task', are also presented in the table and as they do not use observations, we just report again their performance from the previous experiment.

Comparing Table 3 to Table 1 shows that the performance of both ORL methods is degraded compared to the problem with low-dimensional 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 similarity-based approach just slightly outperforms the unified task 'Comp(1)' for sample sizes $N = 150$ and $N = 200$. Although component-based ORL also has weaker performance compared to the low-dimensional observation case, it is performing still

表 3: 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 0.1% confidence level.

Method	$N = 50$	$N = 100$
Comp(CV)	$\mathbf{0.395 \pm 0.085}$	$\mathbf{0.248 \pm 0.044}$
Sim(CV)	$\mathbf{0.398 \pm 0.047}$	0.285 ± 0.014
Comp(1)	$\mathbf{0.375 \pm 0.065}$	0.280 ± 0.023
Single task	1.686 ± 0.004	1.628 ± 0.006

(a) $N = 50$ and $N = 100$

Method	$N = 150$	$N = 200$
Comp(CV)	$\mathbf{0.190 \pm 0.054}$	$\mathbf{0.140 \pm 0.039}$
Sim(CV)	0.244 ± 0.014	0.222 ± 0.012
Comp(1)	0.255 ± 0.012	0.244 ± 0.013
Single task	1.576 ± 0.008	1.526 ± 0.009

(b) $N = 150$ and $N = 200$

表 4: 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 0.1% confidence level.

Method	$N = 50$	$N = 100$
Comp(CV)	$\mathbf{0.648 \pm 0.101}$	$\mathbf{0.699 \pm 0.048}$
Sim(CV)	$\mathbf{0.659 \pm 0.014}$	0.667 ± 0.020
Comp(1)	$\mathbf{0.639 \pm 0.011}$	0.649 ± 0.005
Single task	-0.508 ± 0.121	-0.380 ± 0.177

(a) $N = 50$ and $N = 100$

Method	$N = 150$	$N = 200$
Comp(CV)	$\mathbf{0.724 \pm 0.043}$	$\mathbf{0.740 \pm 0.027}$
Sim(CV)	$\mathbf{0.705 \pm 0.033}$	$\mathbf{0.727 \pm 0.024}$
Comp(1)	0.652 ± 0.002	0.651 ± 0.001
Single task	-0.279 ± 0.192	-0.135 ± 0.201

(b) $N = 150$ and $N = 200$

rather well and clearly outperforms other methods for $N = 150$ and $N = 200$.

Table 4 shows the value of the policies that were found from the transition probabilities learned by different methods for high-dimensional observations case. As expected, compared to the case of low-dimensional observations (see Table 2) both ORL methods have weaker performance. The component-based method slightly outperforms similarity-based method, significantly only for $N = 100$. This suggests that although the KL-error of the similarity-based method is much higher than the component-based method, it still captures useful structure in the transition probabilities resulting in almost similar performance in the grid world task.

In summary, both ORL methods show good performance in the grid world task and the curse of dimensionality has mild effect on their performance.

6 Conclusions

The results of the grid world task show that the proposed ORL framework is suitable in cases useful observations are available about the state-action space. The two proposed method were shown to effectively employ the additional observations to speed up the learning of the transition probabilities.

Our next step is to apply the proposed methods to a more challenging task of object lifting by robotic arm where the robot has observations about the objects. Additionally, our future work is to investigate the relationship of ORL to the studies of Bayesian RL and Partially Observable MDP (POMDP).

参考文献

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.

[2] Hui Li, Xuejun Liao, and Lawrence Carin. Multitask reinforcement learning in partially observable stochastic environments. *The Journal of Machine Learning Research*, 10:1131–1186, 2009.

[3] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1998.

[4] Fumihide Tanaka and Masayuki Yamamura. Multitask reinforcement learning on the distribution of MDPs. In *Computational Intelligence in Robotics and Automation, 2003*, volume 3, pages 1108–1113, July 2003.

[5] Matthew E. 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.

[6] 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.