What-If Prediction via Inverse Reinforcement Learning

Masahiro Kohjima, Tatsushi Matsubayashi, and Hiroshi Sawada
NTT Service Evolution Laboratories, NTT Corporation
1-1 Hikari-no-oka, Yokosuka-shi, Kanagawa, 239-0847 Japan
Email: {kohjima.masahiro, matsubayashi.tatsushi, sawada.hiroshi}@lab.ntt.co.jp

Abstract

What happens if a new street is constructed in a city? What happens if a certain traffic regulation is executed in an exhibition hall? It is important to answer such questions in order to identify "good" operation scenarios for improving city and event comfort. In this paper, we propose a new method on a framework of inverse reinforcement learning (IRL) that can answer these and similar questions. Given *any* scenario among executable scenario candidates, the proposed method predicts the impact on people under the condition that the scenario is executed. The proposed method consists of three steps: parameter estimation, scenario integration, and prediction. In the parameter estimation step, our new IRL algorithm estimates both cost (reward) function and transition probability from past transition logs. Note that it is not necessary that the scenario to be conducted is executed in the past. In the scenario integration step, the estimated parameters are updated by scenario information, and prediction is conducted in the final step. We evaluate the effectiveness of the proposed method by experiments on synthetic and real car probe data.

Introduction

People living in a large city always suffer from congestion. People are caught in traffic jams on the way to work, and need to wait in a long queue to attend a popular event such as sports festivals and product/technology exhibitions. It is desirable to solve or at least reduce the congestions in order to improve the comfort and safety of people and enhance their enjoyment of the event.

One of the major difficulties in easing congestion is that we cannot make trial operations because of the cost and risks. For example, it is too expensive to construct a new street in a city just for a trial. As another example, overcrowding may threaten the safety of visitors if new traffic regulations are set in an exhibition hall. Therefore, it is essential to predict the effects of operations without trying them in the real world. If such predictions are possible, we can compare several *scenarios*, each consisting of set of operations to be executed, based on the prediction results. "Good" scenarios are those among the executable *scenario candidates* that reduce congestion. However, it is very difficult to predict people's transition behavior in a scenario even if past transition data are available.

In this paper, we tackle the problem of predicting the transition of people when any of the scenario candidates is executed. Since this is a kind of virtual prediction problem that involves answering the question, *what will happen if a certain scenario is executed?*, we call this the *what-if prediction problem*.

We consider the setting that (i) past transition logs and (ii) information of the scenario under test are available. The past transition logs do not need to contain past instances in which the scenario was executed. We design a method for the what-if prediction problem in order to satisfy three requirements (R1)(R2)(R3). (R1) The method needs to extract parameters that are invariant regardless of which scenario is executed from past transition logs. This requirement makes what-if predictions possible. (R2) The method needs to merge the information of the scenario to be conducted. Otherwise, we cannot expect precise predictions. (R3) The method needs to conduct prediction without gathering new data in order to find "good" scenarios without executing operations.

Keeping the above requirements in mind, we developed a new method for what-if prediction. Our main idea is the use of *Inverse reinforcement learning* (IRL) (Ng and Russell 2000). IRL is a method that estimates a cost (reward) function for a certain class of Markov decision process (MDP) (Puterman 2005) from agent's optimal behavior. "Inverse" means that the input and output have the reverse relationship to that in standard reinforcement learning (RL) (Sutton and Barto 1998), which estimates optimal behavior given a cost function. We extend the formulation of IRL to satisfy the three requirements. (R1) Cost function is regarded as the most succinct, robust and transferable definition of RL tasks in IRL literature (Abbeel and Ng 2004). By estimating the cost function from past transition data, we can expect the extraction of invariant parameters. (R2) RL has two types of parameters, cost function and transition probability of an environment. Even if the cost function is invariant, the behavior of agent can be drastically changed by modifying the transition probability. Merging scenario information with transition probability allows the 2nd requirement to be satisfied. (R3) The IRL method only estimates the cost function in general. However, transition probabilities which define the state transition given an action are needed to output predictions. Therefore, we propose

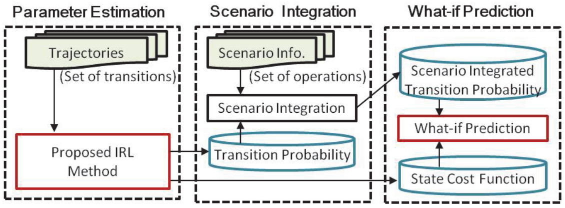


Figure 1: System overview of what-if prediction. Proposed IRL method is a core algorithm.

a new IRL algorithm that can estimate both a cost function and transition probability.

Figure 1 summarizes the procedure of what-if prediction. There are three steps: parameter estimation, scenario integration, and what-if prediction. In the 1st step, our new IRL algorithm estimates both a cost function and transition probability from past transition logs. The transition probability is updated using scenario information in the 2nd step and predictions is conducted in the final step.

Our new IRL method is based on the proposal of Dvijotham and Todorov (Dvijotham and Todorov 2010). This method need not solve a forward problem (i.e., RL problem) repeatedly in an inner loop of the reward estimation process, unlike the well known IRL methods (Ng and Russell 2000)(Ramachandran and Amir 2007) (Ziebart et al. 2008). This success is achieved by the use of a new class of MDP called Linearly Solvable MDP (LMDP) (Todorov 2006). Therefore, our IRL method uses the framework of LMDP. We extend LMDP to formulate a new IRL problem and construct a new IRL method that estimates both a cost function and transition probability. Use of LMDP also contributes to reduce the number of parameters of transition probability in comparison with standard MDP.

The problem setting of this study is related to multi-agent simulations (MAS) which evaluate scenarios by using handmade simulators (Macal and North 2010). MAS is effective in addressing a problem in which the movement of agents, e.g., people, can be easily modeled. For example, in evacuating a building on fire, people rush for the exit. In fact, Ueda et al. recently proposed a method that uses MAS to identify good scenarios (Ueda et al. 2015). However, people's transitions in a city and exhibition hall have more variety. Some people may just be wandering and may stop at a shop/exhibition-booth that catches his/her eye. Our approach has a complementary relation to MAS.

The rest of this paper is organized as follows. In §2, we define an extended variant of LMDP called Shared-Parameter LMDP (SP-LMDP). §3 presents the formulation and the algorithm of proposed IRL method on SP-LMDP. §4 details a way of merging scenario information and §5 is devoted to the numerical experiments. Finally, §6 concludes the paper.

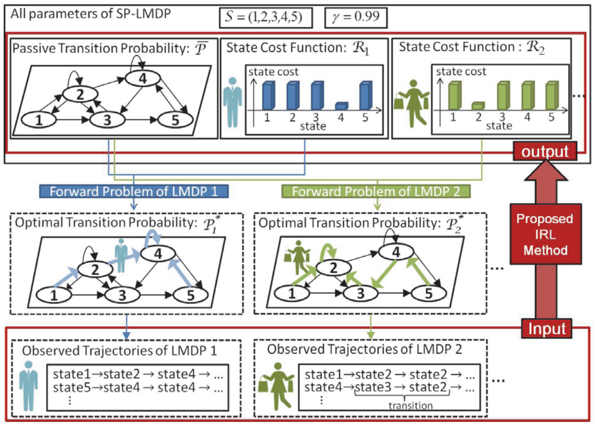

All parameters of SP-LMDP
$S = (1,2,3,4,5)$
$\gamma = 0.99$
Passive Transition Probability: $\bar{\mathcal{P}}$
State Cost Function: $\mathcal{R}_1$
State Cost Function : $\mathcal{R}_2$
state cost
state
1
2
3
4
5
output
Forward Problem of LMDP 1
Forward Problem of LMDP 2
Optimal Transition Probability: $\mathcal{P}_1^*$
Optimal Transition Probability: $\mathcal{P}_2^*$
Proposed IRL Method
Input
Observed Trajectories of LMDP 1
state1→state2 → state4 → ...
state5→state4 → state4 → ...
Observed Trajectories of LMDP 2
state1→state2 → state2 → ...
state4→state3 → state2 → ...
transition

Figure 2: Forward and inverse problem of SP-LMDP.

Shared-parameter LMDP (SP-LMDP)

Definition of LMDP and SP-LMDP

Linearly solvable MDP (LMDP) (Todorov 2006) is defined by the quadruple $\{\mathcal{S}, \bar{\mathcal{P}}, \mathcal{R}, \gamma\}$, where $\mathcal{S} = \{1, 2, \cdots, S\}$ is a finite set of *states* and S is the number of states. $\bar{\mathcal{P}} = \{\bar{p}_{jk}\}_{j,k=1}^{S}$ indicates *passive transition probabilities*, each element of which defines the transition probability from state j to state k when an action is not executed. $\mathcal{R} = \{r_j\}_{j=1}^{S}$ is a *state cost function* and r_j denotes the *state cost* of state j. $\gamma \in [0, 1)$ is a *discount factor*. Note that this work focuses on the "infinite horizon discounted cost" case (Puterman 2005). However, its application to other settings is straight-forward.

We define a new type of LMDP which is defined as its collection that share states $\mathcal{S}$, passive transition probability $\bar{\mathcal{P}}$, and discount factor γ. Each LMDP has its own state cost function, $\mathcal{R}_i$, where i is the index of the LMDP. We call this *shared-parameter LMDP* (SP-LMDP). We formulate our IRL method as an inverse learning problem to estimate passive transition probability and all state cost functions in SP-LMDP. Figure 2 shows all the parameters of SP-LMDP. Formally, SP-LMDP is defined by the quadruple $\{\mathcal{S}, \bar{\mathcal{P}}, \boldsymbol{\mathcal{R}}, \gamma\}$, where $\boldsymbol{\mathcal{R}} = (\mathcal{R}_1, \cdots, \mathcal{R}_I)$ is a set of *state cost function* and $\mathcal{R}_i = \{r_{ij}\}_{j=1}^{S}$. I is the number of the state cost functions. Note that the setting where multiple state cost functions are defined is considered in IRL literature, e.g., (Babes et al. 2011).

In both LMDP and SP-LMDP, *action* $\boldsymbol{a}$ is represented as a continuous value $\mathbb{R}^S$ dimensional vector and the *action transition probability* from state j to state k when action $\boldsymbol{a}_j = \{a_{jk}\}_{k=1}^{S}$ is executed is defined as

$$p_{jk}(\boldsymbol{a}_j) = \bar{p}_{jk} \exp(a_{jk}). \tag{1}$$

Note that action executed in state j must belong to $\mathcal{A}_j = \left\{\boldsymbol{a}_j \in \mathbb{R}^S | \sum_k p_{jk}(\boldsymbol{a}_j) = 1\right\}$ so that the summation of the probabilities equals one[1]. Therefore, the transition probability itself can be controlled by an action. For example, increasing a_{jk} increases the probability of the transition from state j to k. In order to execute a certain action, it is necessary to pay the action cost defined by *action cost function*. The *action cost* when action $\boldsymbol{a}_j$ is executed in state j is defined as

$$q_j(\boldsymbol{a}_j) = KL(\boldsymbol{p}_j(\boldsymbol{a}_j) || \boldsymbol{p}_j(\boldsymbol{0})), \tag{2}$$

[1] More precisely, $\bar{p}_{jk} = 0 \rightarrow a_{jk} = 0$ since a_{jk} doesn't affect the action transition probability $p_{jk}(\boldsymbol{a}_j)$ when $\bar{p}_{jk} = 0$.

where $KL(\cdot||\cdot)$ is Kullback-Leibler divergence and $\boldsymbol{p}_j(\boldsymbol{a}) = \{p_{jk}(\boldsymbol{a})\}_{k=1}^S$. Thus, action cost increases as $p_{jk}(\boldsymbol{a})$ deviates further from passive transition $\bar{p}_{jk}$. Note that when the action is a zero vector, $\boldsymbol{a} = \boldsymbol{0}$, $p_{jk}(\boldsymbol{0})$ equals the passive transition probability $\bar{p}_{jk}$ and action cost $q_j(\boldsymbol{0}) = 0$.

Forward Problem of SP-LMDP

The forward problem of the i-th LMDP in SP-LMDP can be solved independently following the method used for LMDP. Let $\boldsymbol{\pi}_i = \{\boldsymbol{a}_{ij}\}_{j=1}^S$ be a policy on the i-th LMDP whose element $\boldsymbol{a}_{ij}$ indicates the action executed in state j. The *value function* of policy π_i, $\boldsymbol{v}_i^{\pi_i} = \{v_{ij}^{\pi_i}\}_{j=1}^S$, is defined such that element v_{ij}^{π} indicates the expected sum of future cost from state j following policy $\boldsymbol{\pi}_i$ on the i-th LMDP,

$$v_{ij}^{\pi_i} = \lim_{T\to\infty} \mathbb{E}_{\boldsymbol{d}_T}\left[\sum_{t=1}^{T} \gamma^{t-1}\left\{r_{is_t} + q_{s_t}(\boldsymbol{a}_{is_t})\right\}\Big| s_1 = j\right],$$

where $\mathbb{E}_{\boldsymbol{d}_T}$ denotes the expectation over trajectory $\boldsymbol{d}_T = \{s_t\}_{t=1}^T$, the finite time step transitions from $t = 1$ to T. s_t denotes the visit state at time t, and $\boldsymbol{d}_T$ follows probability $P(\boldsymbol{d}_T|\bar{\mathcal{P}}, \boldsymbol{\pi}_i) = p_{s_1}^{ini}\prod_{t=1}^{T-1} p_{s_t s_{t+1}}(\boldsymbol{a}_{is_t})$. p^{ini} is the initial state distribution.

The forward problem of the i-th LMDP is to obtain optimal policy $\boldsymbol{\pi}_{\boldsymbol{i}}^* = \{\boldsymbol{a}_{ij}^*\}_{j=1}^S$, i.e. the one that minimizes the expected sum of the future cost. The optimal action in state j is given by

$$\begin{aligned}\boldsymbol{a}_{ij}^* &= \underset{\boldsymbol{a}_{ij}\in\mathcal{A}_j}{\arg\min}\left\{r_{ij} + q_j(\boldsymbol{a}_{ij}) + \gamma\sum\nolimits_{k=1}^{S} p_{jk}(\boldsymbol{a}_{ij})v_{ik}\right\}\\ &= -\gamma v_{ij} - \log\left\{\sum\nolimits_k \bar{p}_{jk}\exp(-\gamma v_{ik})\right\}, \qquad (3)\end{aligned}$$

where $\boldsymbol{v}_i = \{v_{ij}\}_{j=1}^S$ is the optimal value function of the i-th LMDP, $v_{ij} = \min_\pi v_{ij}^\pi$, which satisfies the following optimal equation:

$$v_{ij} = r_{ij} - \log\left\{\sum\nolimits_k \bar{p}_{jk}\exp(-\gamma v_{ik})\right\}. \qquad (4)$$

This optimal function can be efficiently obtained by power iteration (Todorov 2006). Inserting Eq. (3) into Eq. (1), *optimal transition probability*, the transition probability when the optimal action is executed, can be written as

$$p_{ijk}^* = p_{ijk}(\boldsymbol{a}_{ij}^*) = \frac{\bar{p}_{jk}\exp(-\gamma v_{ik})}{\sum_\ell \bar{p}_{j\ell}\exp(-\gamma v_{i\ell})}. \qquad (5)$$

We emphasize that the above form of optimal transition probability is a direct consequence of LMDP unlike Bayesian IRL, which uses the value function as a potential function (Ramachandran and Amir 2007).

Proposed IRL method

This subsection details the proposed IRL method; it can estimate both state cost and passive transition probability for PS-LMDP. This type of IRL problem has not been well studied in IRL literature except for the work for partially observable setting (Makino and Takeuchi 2012). We denote the all transition logs which are used for estimation as $\mathcal{D}$ and the number of transitions from state j to state k in the i-th LMDP as n_{ijk}. Our IRL method is naturally derived by considering that each transition data is generated by the probability defined in Eq. (5) which has the parameter $\mathcal{V}, \bar{\mathcal{P}}$; the probability of all transition $\mathcal{D}$ is given by

$$P(\mathcal{D}|\mathcal{V}, \bar{\mathcal{P}}) = \prod_i \prod_{j,k\in\mathcal{S}}\left\{\frac{\bar{p}_{jk}\exp(-\gamma v_{ik})}{\sum_\ell \bar{p}_{j\ell}\exp(-\gamma v_{i\ell})}\right\}^{n_{ijk}}. \qquad (6)$$

Our algorithm is designed to minimize the sum of negative log-likelihood term $-\log P(\mathcal{D}|\mathcal{V}, \bar{\mathcal{P}})$ and regularization term $\Omega(\mathcal{V}, \bar{\mathcal{P}})$ which is defined as

$$\Omega(\bar{\mathcal{P}}, \mathcal{V}) = -\sum\nolimits_{j,k}(\alpha - 1)\log\bar{p}_{jk} + \frac{\beta}{2}\sum\nolimits_{i,j} v_{ij}^2. \qquad (7)$$

α and β are weight parameters. Note that this regularization is equivalent to putting a Dirichlet prior on $\bar{\mathcal{P}}$ and a Gaussian prior on $\mathcal{V}$. Then, the objective function is given by

$$\begin{aligned}\mathcal{L}(\mathcal{V}, \bar{\mathcal{P}}) = &\sum_{i,j}\left\{n_{i\cdot j}\gamma v_{ij} + n_{ij\cdot}\log\Big(\sum_{k'}\bar{p}_{jk'}\exp(-\gamma v_{ik'})\Big)\right\}\\ &- \sum_{j,k}(n_{\cdot jk} + \alpha - 1)\log\bar{p}_{jk} + \frac{\beta}{2}\sum_{i,j} v_{ij}^2, \qquad (8)\end{aligned}$$

where dot index means that the corresponding index is summed out: $n_{\cdot jk} = \sum_i n_{ijk}$, $n_{i\cdot k} = \sum_j n_{ijk}$, $n_{ij\cdot} = \sum_k n_{ijk}$. We construct an algorithm that iteratively updates $\mathcal{V}$ and $\bar{\mathcal{P}}$. After $\mathcal{V}$ and $\bar{\mathcal{P}}$ are estimated, state cost $\mathcal{R}_i$ can be computed using the estimated $\boldsymbol{v}_i$ and $\bar{\mathcal{P}}$ by Eq. (4). Pseudo code of the proposed algorithm is shown in Algorithm 1.

Update of Value Function: For the minimization with respect to $\mathcal{V}$, any unconstrained optimization method such as Newton method can be applied. For the gradient-based method, the 1st partial derivative is given by

$$\frac{\partial\mathcal{L}(\mathcal{V}, \bar{\mathcal{P}})}{\partial v_{i\ell}} = \gamma n_{i\cdot\ell} - \gamma\sum\nolimits_j n_{ij\cdot}p_{ij\ell}^* + \beta v_{i\ell}. \qquad (9)$$

Note that the objective function is convex while $\bar{\mathcal{P}}$ is fixed. In the experiment section that follows, we use the LBFGS method.

Update of Passive Transition Probability: For the minimization with respect to $\bar{\mathcal{P}}$, we use Lagrange multipliers to obtain the necessary condition of the limiting point. The Lagrange function is defined as $\mathcal{F}(\bar{\mathcal{P}}, \lambda) = \mathcal{L}(\mathcal{V}, \bar{\mathcal{P}}) + \sum_j \lambda_j\big(\sum_k \bar{p}_{jk} - 1\big)$, where λ is a Lagrange coefficient. By solving the above, necessary conditions are given by the following non-linear simultaneous equation $F_{jk}(\bar{\boldsymbol{p}}_j) = 0\ (\forall j, k)$, where

$$F_{jk}(\bar{\boldsymbol{p}}_j) = \sum\nolimits_i n_{ij\cdot}p_{ijk}^* - n_{\cdot jk} + (S\bar{p}_{jk}-1)(\alpha-1). \qquad (10)$$

Note that $\bar{p}_{jk}$ is also included in p_{ijk}^* as defined in Eq. (5). Then, $\bar{\mathcal{P}}$ is updated to the value that satisfies Eq. (10).

Note that if passive transition $\bar{\mathcal{P}}$ is known and fixed, and the number of LMDP, I, equals 1, $\alpha = 1.0$, $\beta = 0.0$, the proposed method reduces to the method by Dvijotham and Todorov (Dvijotham and Todorov 2010). In addition, if $\bar{\mathcal{P}}$ is fixed to a uniform distribution, it is equivalent to *maximum entropy IRL* (Ziebart et al. 2008), which was also proven by Dvijotham and Todorov (Dvijotham and Todorov 2010).

Algorithm 1 Proposed IRL Algorithm

```
Input: $\mathcal{D}, \gamma, \alpha, \beta$,  Output: $\bar{\mathcal{P}}, \mathcal{V}, \boldsymbol{\mathcal{R}}$
Initialize $\bar{\mathcal{P}}$
repeat
    Minimize Eq. (8) w.r.t. $\boldsymbol{v}_i$ and update $\boldsymbol{v}_i$ $(\forall i)$.
    Solve non-linear simultaneous equation $F_{jk}(\bar{\boldsymbol{p}}_j) = 0$
    for all $k$ and update $\bar{\boldsymbol{p}}_j$ to its solution $(\forall j)$.
until Converge
Compute $\boldsymbol{\mathcal{R}}$: $r_{ij} = v_{ij} + \log\Big(\sum_k \bar{p}_{jk} \exp(-\gamma v_{ik})\Big)$
```

Scenario Integration & What-if Prediction

This subsection provides a way to use the estimated state cost function and passive transition for what-if prediction. The remainder of the procedure of our system (Fig. 1) consists of scenario integration and what-if prediction.

Integration of scenario information: First, given the output of the proposed IRL method and scenario information, we update passive transition probability by merging the scenario. We denote the scenario integrated (passive) transition probability as $\bar{\mathcal{P}}^{\mathrm{sc}}$. As shown in Table 1, we assume that scenario information is given by table format which tells us which state/edge becomes what kind of condition, e.g., keep out, one way and so on. Therefore, the update rules are intuitive. For example, if state s_k becomes keep out, then, the passive transition probability is updated such that $\bar{p}_{jk} = 0$ for all j. Thus what we need to be concerned with is "normalization" to satisfy the sum-to-one constraint. While various types of normalization procedure are available, in the later experiment, we use the following normalization which uses softmax function while zero probability remain unchanged: $\bar{p}^{\mathrm{sc}}_{jk} \propto \bar{p}_{jk} \exp(-\bar{p}_{jk})$.

What-if transition probability: The final step is what-if prediction. The assumption made here is that the (estimated) state cost function is consistent, i.e., unchanged by scenario information [2]. Thus, we define the what-if transition probability as the optimal transition probability of PS-LMDP $\{\mathcal{S}, \bar{\mathcal{P}}^{\mathrm{sc}}, \boldsymbol{\mathcal{R}}, \gamma\}$, where $\boldsymbol{\mathcal{R}}$ is the state cost function estimated by the IRL method. Since the optimal transition probability of PS-LMDP is given by Eq. (5), what-if transition probability, p^{if}_{ijk}, is defined as

$$p^{\mathrm{if}}_{ijk} = \frac{\bar{p}^{\mathrm{sc}}_{jk} \exp(-\gamma v^{\mathrm{sc}}_{ik})}{\sum_\ell \bar{p}^{\mathrm{sc}}_{j\ell} \exp(-\gamma v^{\mathrm{sc}}_{i\ell})}, \tag{11}$$

where $\boldsymbol{v}^{\mathrm{sc}}$ denotes the value function of the above PS-LMDP. Figure 3 shows an example explaining why we adopt this definition. From the definition of the value function, the values of $\boldsymbol{v}^{\mathrm{sc}}$ are changed from the previous values before scenario integration. Therefore, the magnitude relation of its value between states can be changed (See, red circle area in Fig. 3), and so the prediction of a person's typical trajectory can be changed since transition to lower value state likely to occur. This enables us to conduct what-if prediction.

Table 1: Scenario information and update rules of $\bar{\mathcal{P}}$.

scenario information	corresponding update
Edge s_j-s_k: keep-out	set $\bar{p}_{jk}$=$\bar{p}_{kj}$=0 and normalize.
State s_k: keep-out	set $\bar{p}_{jk}$=0 and normalize, for all j.
State s_j: one way to s_k	set $\bar{p}_{jk}$=1 and $\bar{p}_{jk'}$=0 $(\forall k' \neq k)$.

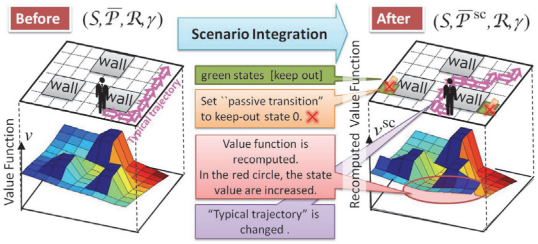

Before
$(\mathcal{S}, \bar{\mathcal{P}}, \mathcal{R}, \gamma)$
Scenario Integration
After
$(\mathcal{S}, \bar{\mathcal{P}}^{\mathrm{sc}}, \mathcal{R}, \gamma)$
wall
wall
wall
Typical trajectory
Value Function
v
green states [keep out]
Set ``passive transition" to keep-out state 0.
Value function is recomputed. In the red circle, the state value are increased.
"Typical trajectory" is changed .
Recomputed Value Function
v^{sc}

Figure 3: An illustrative example of value function and typical trajectory before and after scenario integration.

Experiment

Synthetic Data: We evaluate the what-if prediction performance of our method using synthetic data and real car probe data. In the first experiment, we construct a 10×10 grid world as shown in Fig. 4 (a). Passive transition probability from each state is set to a uniform probability for the up and down, left and right states (if some of them involve walls or obstacles, we consider self-transition). We also prepare four state cost functions, $\mathcal{R}_1, \cdots, \mathcal{R}_4$, and the state cost is set to 0 only for the corresponding goal state shown in Fig. 4 (a) and is set to 1 for the other states. We prepare the scenario information which sets the 4 states to keep out as shown in Fig. 4 (b). Under this scenario, it becomes impossible to go through the left or right corridor. By solving the forward problem with true state cost and transition probability, we compute true optimal transition probability; we use this probability to generate training data and validation data. Training data is used as the input of the proposed method and validation data is used to choose the optimal hyper parameter. After that, by computing true what-if transition probabilities by scenario integration, we generate test data. In order to evaluate the performance while varying the amount of training data, an equal number of one step transition data was collected in all states; we set $n_{ij\cdot} = 5, 10, 20, 40, 100$.

Real Car Probe Data: For the second experiment, we used real car probe data provided by NAVITIME JAPAN Co, Ltd. This is a collection of GPS trajectory of users who used a car navigation application on smartphones from 2015.4.13 to 2015.5.17 in Kanagawa Prefecture, Japan. In particular, we used the trajectories in the area of Minato-Mirai-21 dis-

[2] This assumption may be broken in extreme situations, such as a huge natural disaster since the behavior of people will changed drastically during an evacuation. However, we consider that this assumption is reasonable in many cases because, for example, crowd control and traffic regulation plans don't effect people's final destination and attraction of the certain place of the city, i.e., cost of the state is consistent.

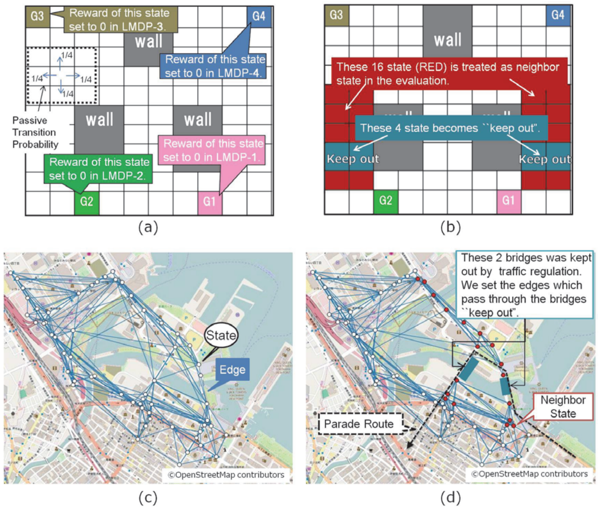

G3
Reward of this state set to 0 in LMDP-3.
G4
wall
Reward of this state set to 0 in LMDP-4.
1/4
1/4
1/4
1/4
Passive Transition Probability
wall
wall
Reward of this state set to 0 in LMDP-1.
Reward of this state set to 0 in LMDP-2.
G2
G1
(a)
G3
G4
wall
These 16 state (RED) is treated as neighbor state in the evaluation.
These 4 state becomes ``keep out''.
Keep out
Keep out
G2
G1
(b)
State
Edge
©OpenStreetMap contributors
(c)
These 2 bridges was kept out by traffic regulation. We set the edges which pass through the bridges ``keep out''.
Neighbor State
Parade Route
©OpenStreetMap contributors
(d)

Figure 4: Setting for experiment. (a)(b) Gridworld for synthetic data experiment and (c)(d) landmark graph for car probe data experiment. Training data and validation data are obtained in settings (a)(c). Test data is obtained in setting (b)(d).

trict in Yokohama, since an annual parade [3] was held and traffic regulation was executed on 2015.5.3 (Sun.). We use the log of this day as the test data. We also use the log of the holiday between 2015.4.13 to 2015.5.1 (5 days in total) for training data and the log of 2015.5.2 for validation data. This allows us to evaluate what-if prediction performance. Since map matching algorithms have already been applied to the original trajectories, each point on the trajectory is tied to street id information. However, in order to remove noisy transitions derived from GPS noise or the failure of map matching, we apply the landmark graph construction algorithm (Yuan et al. 2010) in order to obtain an abstract street network as shown in Fig. 4(c). We convert the GPS trajectories to transition data between the nodes (states) of this graph. We also prepare scenario information based on the traffic regulation of the event day as shown in Fig. 4(d). In order to consider the time zone dependency of transitions, we use the logs of 10:00-12:59, 14:00-16:59, 17:00-19:59 as the logs of LMDP1, 2 and 3, respectively. We use the log of the parade day at 10:00-12:59 as test data since the traffic regulation is conducted only in that period.

Evaluation Measure: We use the negative log likelihood metric to evaluate what-if prediction performance. The negative test log likelihood is defined as $(1/\mathcal{T})\sum_{i=1}^{I}\sum_{j,k\in\mathcal{S}} -n_{ijk}^{\text{test}}\log\hat{p}_{ijk}^{\text{if}}$, where $\mathcal{T}$ is the number of test data sets and n_{ijk}^{test} indicates the number of transitions from state j to state k in the i-th LMDP. For investigating the effect of scenario integration, we also show the log likelihood performance of the transition from neighbor states, which is near to the keep-out state, shown as red-colored states in Fig. 4(b)(d), because the transition from these states will change drastically.

Baseline Method: For the comparison, we use three existing methods as the baselines: Random, Markov and MaxEnt (Ziebart et al. 2008). In all methods, we use the adjacency information E_j for all j, which denotes the set of states reachable from state j by one step transition. The transition probability of Random and Markov is computed as $p_{ijk}^{\text{random}} = 1/|E_j|$ and $p_{ijk}^{\text{markov}} = (n_{ijk}+\alpha)/(n_{ij\cdot}+\alpha|E_j|)$, respectively. $|\cdot|$ denotes the number of elements in the set. The probability of MaxEnt is computed by Algorithm 1 with fixed passive transition probability $\bar{p}_{jk} = 1/|E_j|$. The proposed method also uses the adjacency information by modifying regularization term Eq. (7). Note that hyperparameters of all methods are set to the one yielding the best performance for the validation data. For fair comparison, the existing method also use scenario information. The probability of Random and Markov is updated following the passive transition update rules of the previous section. The update for MaxEnt is analogous to that of our method.

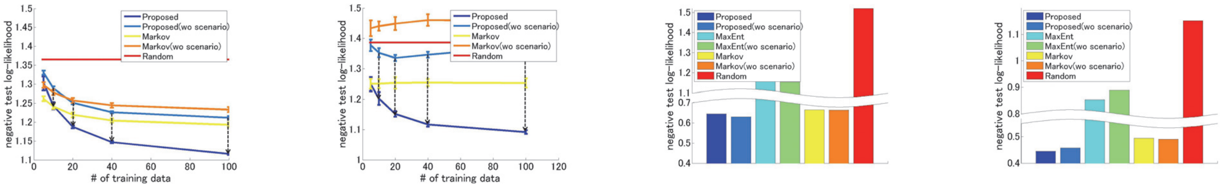
Synthetic Data Result: Figure 5 (a)(b) shows the results of the synthetic data experiment [4]. Figure 5 (a) indicates that the proposed method has performance competitive with that of Markov when the amount of training data does not exceed 10. As the amount of training data increases, our method outperforms the baseline methods. Comparing the proposed method w/o scenario integration, the degree of improvement increases with the amount of data. This is because the estimation accuracy of the value function and state cost function improved as the amount of training data increases. The above interpretations are also supported by Fig. 6(a). Value function with 20 and 40 training data seems to yield good estimations the true values; the value function with 10 training data is not accurate enough and its value is not changed by scenario integration. Figure 5 (b) also shows similar results except that the degree of improvement attained with the use of scenario information is large. This is reasonable since transitions near the keep-out state tend to be drastic as shown in Figure 6(a).

Car Probe Data Result: Figure 5 (c)(d) show the results of the car probe data experiment; they indicate that the proposed method outperforms the baseline methods. We can also confirm the validity of the proposed method from Fig. 6 (b)[5] since the state with lower value corresponds to attractive spots that attract drivers. Figures 6 (c) also show that, under the traffic regulation, the estimated value function seems to reflect the fact that drivers try to avoid the regulation area. These results imply the effectiveness of the proposed method for what-if prediction. Without using transition logs under the traffic regulation, we can predict probe car transitions under the traffic regulation. Moreover, we also find that the experimental results suggest future research directions. Comparing Fig. 5 (c) and Fig. 5 (d), the performance improvement attained by scenario integration can be confirmed only in the neighbor states result. This re-

[3] http://www.yokohamajapan.com/upcoming-events/63rd-yokohama-parade-international-costume-parade-2015/

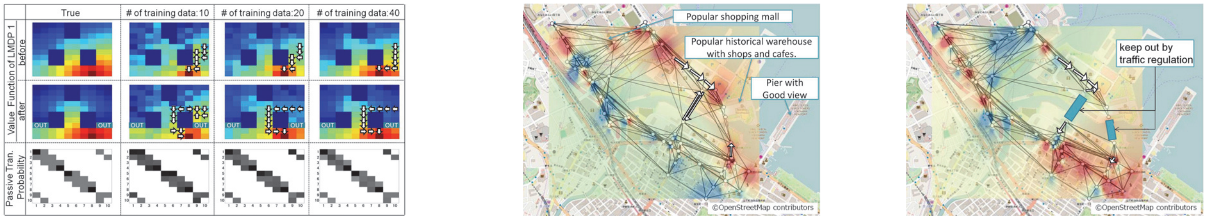
[4] Since true passive transition is uniform in many states, we make no comparison with MaxEnt to ensure a fair comparison.

[5] The figures are drawn using QGIS and interpolation plugin.



(a) All states (b) Neighbor states (c) All states (d) neighbor states

Figure 5: (a)(b) Result of synthetic data varying the amount of training data $n_{ij\cdot} = 5, 10, 20, 40, 100$. Average and standard deviation of 20 experiments are shown. Dotted arrow indicates the improvement by scenario integration. (c)(d) Result of car probe data.



(a) synthetic data (b) car probe data (before) (c) car probe data (after)

Figure 6: (a) The true and estimated value functions of LMDP 1 before/after scenario integration and passive transition probability among states 1∼10: synthetic data experiment. (b) (c) the estimated value functions before/after scenario integration: car probe data experiment. Red/Blue color indicate the value is small/large. White arrow indicates the transition that is most likely to occur from corresponding state.

sult is very similar to that of synthetic data experiment with a small amount of training data. This means that, in order to fully enjoy the power of scenario integration, the parameter estimation accuracy of IRL needs to be improved.

Conclusions and Future work

In this paper, we tackled the what-if prediction problem. We proposed a methodology to tackle the problem and a new IRL method that estimates both a cost function and transition probability. We confirmed the effectiveness of the method by comparing it in experiments with existing methods.

We list three future works. First, we need to construct an IRL method that works well with small amounts of data. Applying a Bayesian framework to handle the uncertainty of the estimation or the use of auxiliary information may be an effective approach. Second, we need to extend the method in order to deal with the case that state cost function is *not* consistent. Such extension broaden the application area of the method. Third, we need to interact with researchers in e.g. urban engineering for solving congestion in a city by using our method to identify good scenarios. By fusing their knowledge with our technology, we hope to improve the method and to contribute making the world less congested.

References

Abbeel, P., and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proc. ICML*, 1. ACM.

Babes, M.; Marivate, V.; Subramanian, K.; and Littman, M. L. 2011. Apprenticeship learning about multiple intentions. In *Proc. ICML*, 897–904.

Dvijotham, K., and Todorov, E. 2010. Inverse optimal control with linearly-solvable mdps. In *Proc. ICML*, 335–342.

Macal, C. M., and North, M. J. 2010. Tutorial on agent-based modelling and simulation. *Journal of simulation* 4(3):151–162.

Makino, T., and Takeuchi, J. 2012. Apprenticeship learning for model parameters of partially observable environments. In *Proc. ICML*, 1495–1502.

Ng, A. Y., and Russell, S. 2000. Algorithms for inverse reinforcement learning. In *Proc. ICML*.

Puterman, M. L. 2005. Markov decision processes: Discrete stochastic dynamic programming.

Ramachandran, D., and Amir, E. 2007. Bayesian inverse reinforcement learning. In *Proc. IJCAI*, 2586–2591.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*. MIT press Cambridge.

Todorov, E. 2006. Linearly-solvable markov decision problems. In *Proc. NIPS*, 1369–1376.

Ueda, N.; Naya, F.; Shimizu, H.; Iwata, T.; Okawa, M.; and Sawada, H. 2015. Real-time and proactive navigation via spatio-temporal prediction. In *Proc. UbiComp*, 1559–1566.

Yuan, J.; Zheng, Y.; Zhang, C.; Xie, W.; Xie, X.; Sun, G.; and Huang, Y. 2010. T-drive: driving directions based on taxi trajectories. In *Proc. SIGSPATIAL*, 99–108. ACM.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum entropy inverse reinforcement learning. In *AAAI*, 1433–1438.