Slides
http://www.pomdp.net

Advanced Cutting Edge Research Seminar

Dialogue Management using Reinforcement Learning

Assistant Professor
Koichiro Yoshino

Nara Institute of Science and Technology
Augmented Human Communication Laboratory
PRESTO, Japan Science and Technology Agency

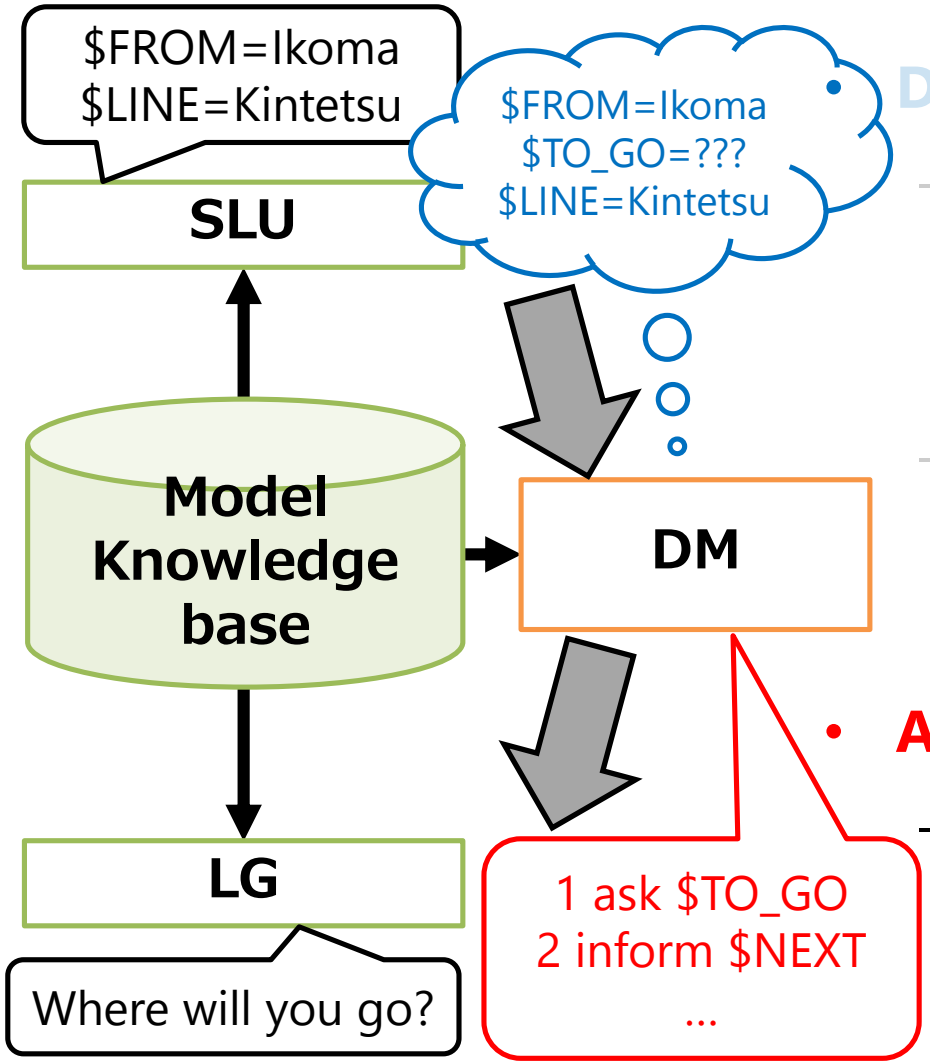
Course works

1. **Basis of spoken dialogue systems**
   - Type and modules of spoken dialogue systems
2. **Deep learning for spoken dialogue systems**
   - Basis of deep learning (deep neural networks)
   - Recent approaches of deep learning for spoken dialogue systems
3. **Dialogue management using reinforcement learning**
   - Basis of reinforcement learning
   - Statistical dialogue management using intention dependency graph
4. **Dialogue management using deep reinforcement learning**
   - Implementation of deep Q-network in dialogue management

Dialogue management using reinforcement learning

- **Task definition of dialogue management (re-visiting)**
  - In Markov decision process
  - Problems to apply reinforcement learning
  - Intention dependency graph
- **Q-learning**
  - Algorithm of Q-learning
- **User simulator**
  - To mimic the user behavior in learning process of Q-learning
  - Data driven user simulators
- **Reward functions**
  - Task completion, user satisfaction, inversed reinforcement learning

Task of dialogue management



Dialogue state tracking

- Decide the current belief (confidence of states) with belief update
  - $b^t \approx P(o^t|s^t)\sum_{s_i} P(s_j^t|s_i^{t-1})b^{t-1}$
- Decide with RNN (LSTM)
  - $h^t = \tanh(W_{Xh}X^t + W_{hh}h^{t-1} + c_h)$
  - $b^t = \mathrm{softmax}(W_{hb}h^t + c_b)$

- **Action decision**
  - Mapping between belief and action
    - $\pi(b, a)$ or $\pi(b) = a$

An example task of dialogue system

- **Let's consider a dialogue system that can control the car-audio with voice**
  - There are two actions: "play music" and "control volume"

```
Play_music{
$ARTIST=Beatles
$ALBUM=Let it be
}
```

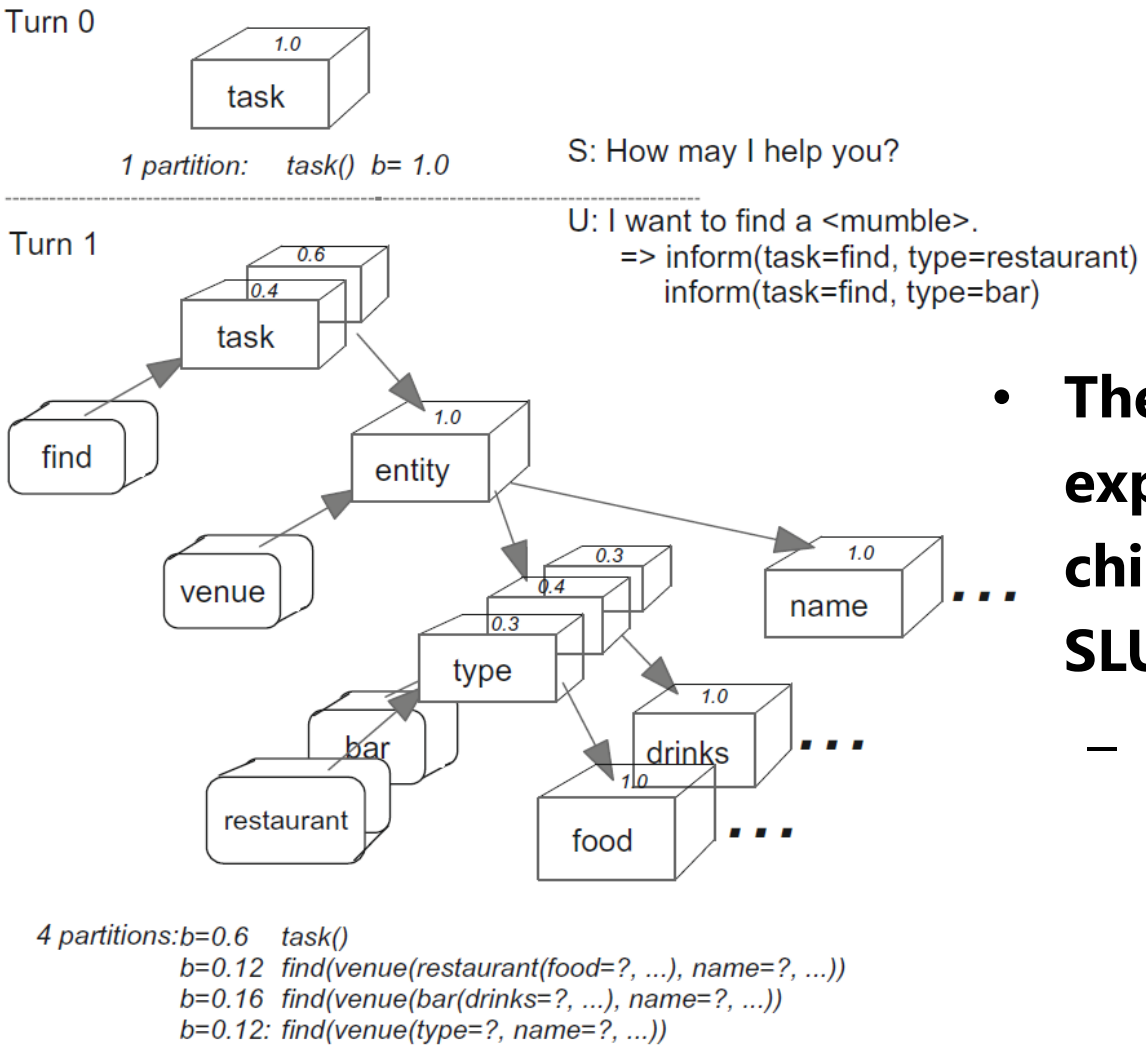
```
Control_volume{
$VOLUME=+1
}
```

- **How do you design the space of user state?**
  - Consider any possible states of each slot

    → State space will be extremely large: but it will be accurate if the model works: **Hidden Information State model**
  - Consider binary flag of each slot

    → State space will be smaller: **Intention Dependency Graph**

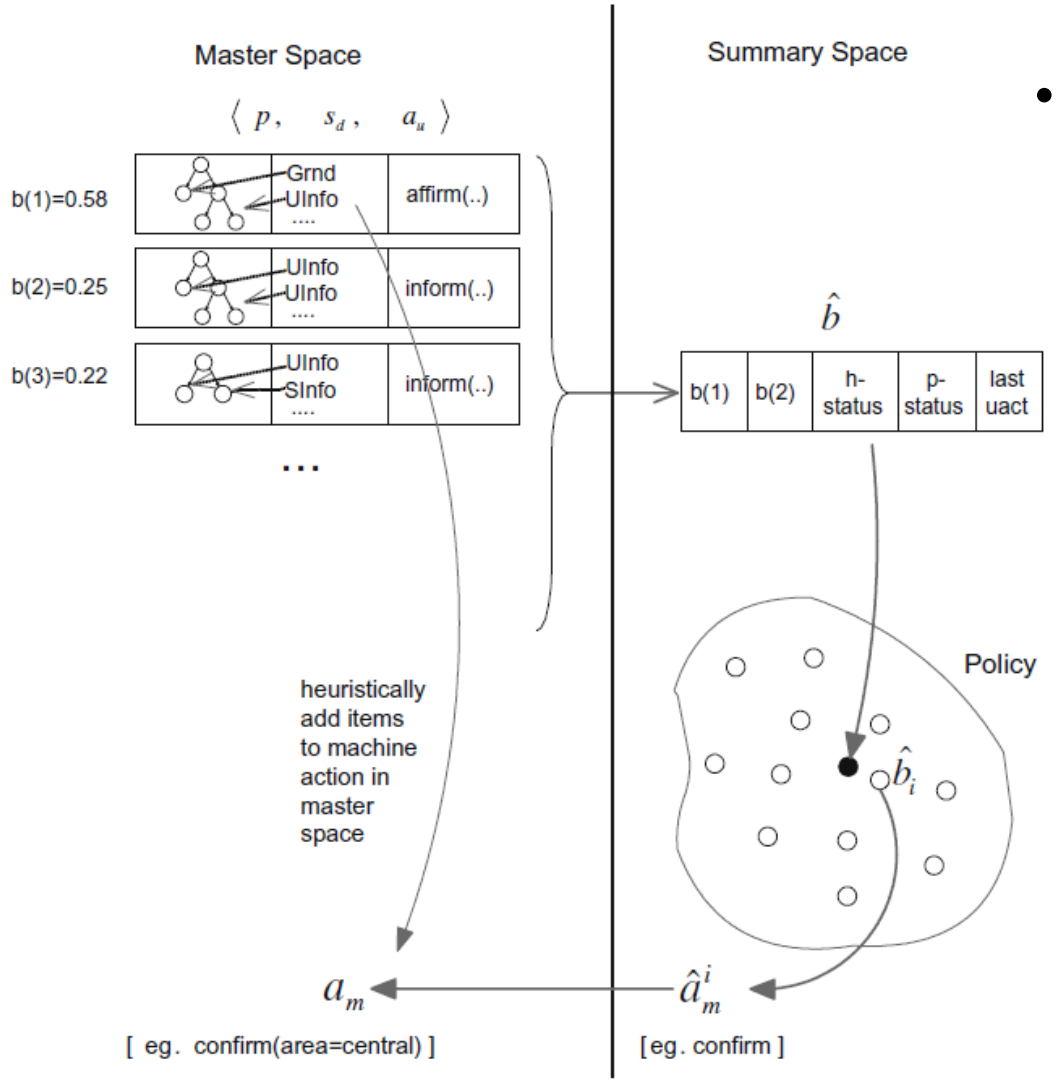
Hidden Information State model



The hidden information state model: a practical framework for POMDP-based spoken dialogue management. Young et al., Computer Speech & Language, Vol.24, No.2, pp.150-174, 2010.

- **The system dynamically expand the corresponding children according to the SLU results**
  - This solution reduce the required data for DST and action decision

Master space and summary space



- **The state space of dialogue state is mapped into the space of summary space**
  - Summary space only decide the action
  - Slot values to be used in the action are filled with some values in master space

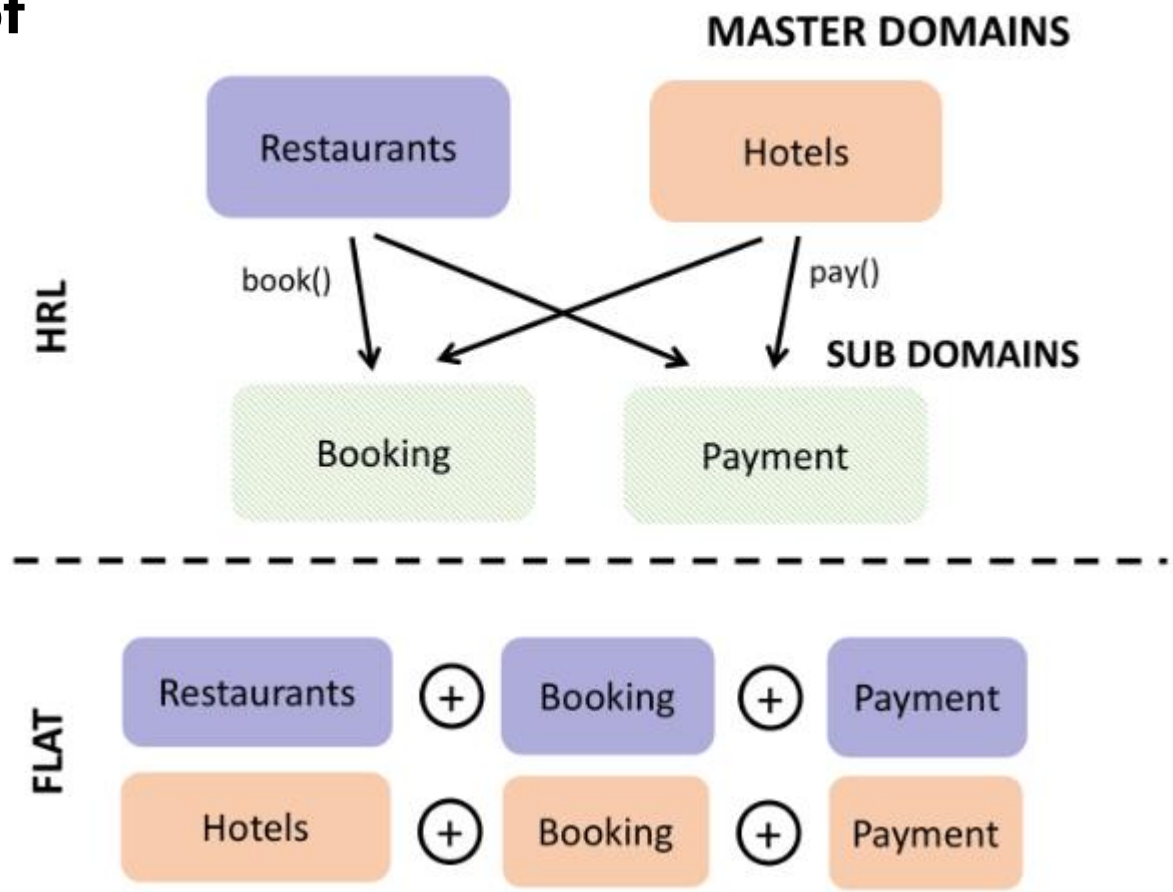
Benefits to use task knowledges or structures

- **Applies for new domain system**
  - How do we apply statistical dialogue management?
    - Data collection is required
  - Knowledges or structures make it possible to work the system without any data for learning
    - We run the system first, and adapt to the collected data
- **Gives weights to unseen states or sequence**
  - It is impossible to prepare dialogue data that covers every possible situation
- **Adaptation for unseen states or new domain is still a large problem and some works focus on this problem**

Approach for new domain

- **Express master domains with combinations of sub domains**

Sub-domain Modelling for Dialogue Management with Hierarchical Reinforcement Learning. Budzianowski et al., Proc. SIGDIAL 2017



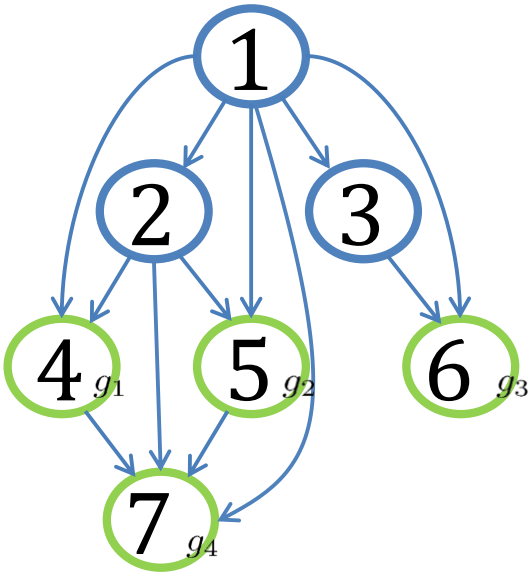
But to learn the model…

We learn on simpler model

Intention Dependency Graph

- **Tasks: car audio control**
  - Play any music or control volume

1. ROOT[] (=no specified request)
   - Root note
2. PLAY_MUSIC[artist=null, album=null]
3. PLAY_MUSIC[artist=$artist_name, album=null]
4. PLAY_MUSIC[artist=null, album=$album_name]
5. PLAY_MUSIC[artist=$artist_name, album=$album_name]
   - Possible PLAY_MUSIC states (null slots depends on filled slots)
6. CONTROL_VOLUME[value=null]
7. CONTROL_VOLUME[value=$up_or_down]
   - Possible CONTROL_VOLUME states



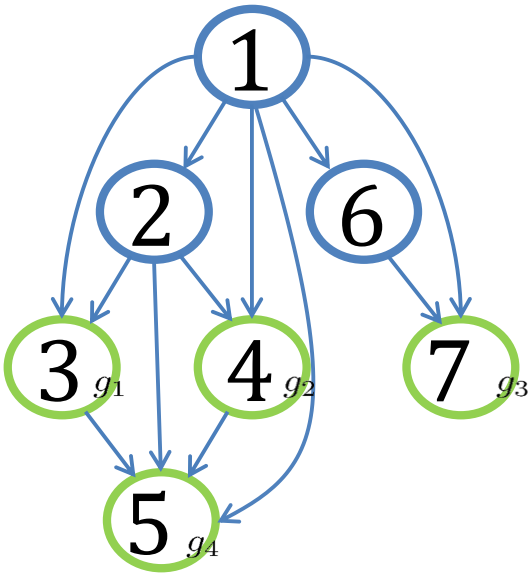
Statistical dialogue management using intention dependency graph. Yoshino et al., In Proc. IJCNLP, pp.962-966, 2013.

How do we develop the IDG?

- **Combinations of binary flag of each slot is used as a state**
  - We consider any states even if slot-values are NULL
  - It is possible to make nodes that have actual values (e.g. ARTIST=Beatles)
    - As hidden information state modeling
    - The resultant model can manage the dialogue more accurate, but it requires much more data to learn the model
- **We define state transitions according to the filling process of slots**
  - NULL states depend on filled states
  - But it is not tree (Directed Acyclic graph; DAG)
- **Define frames that can be a task goal**
  - It is possible to define any states as goals

Actions for IDG

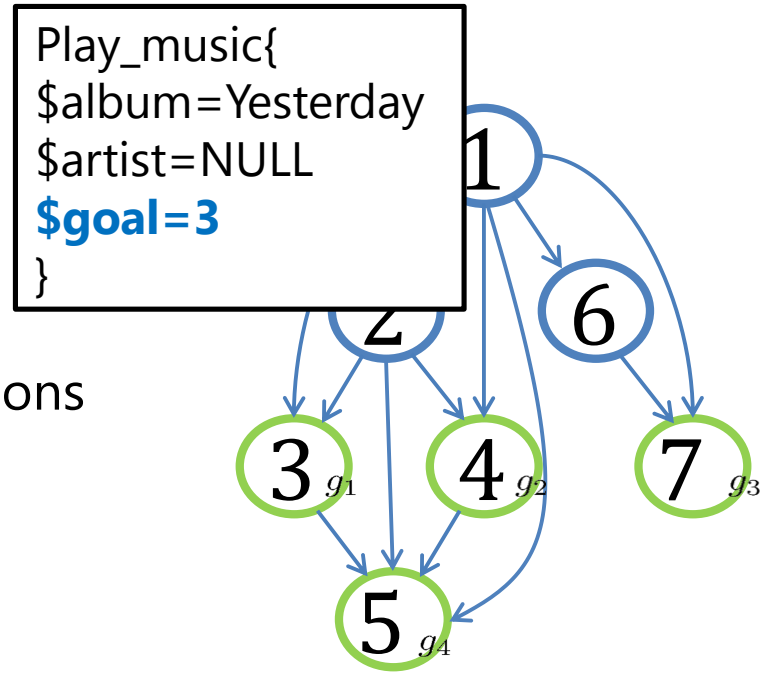
- **Task: Play music, control volume**
  1. root
  2. Play music (w. o. artist and album)
  3. Play music (w. artist name)
  4. Play music (w. album name)
  5. Play music (w. both of artist and album name)
  6. Change volume (w. o. value)
  7. Change volume (w. value)



- **Define goal actions for goals**
  - 3, 4, 5, 7＝Do the defined task (play or control)
- **Define actions for non-goal nodes**
  - 1, 2, 6=Request for the slot to be fulfilled

How do we know the goal?

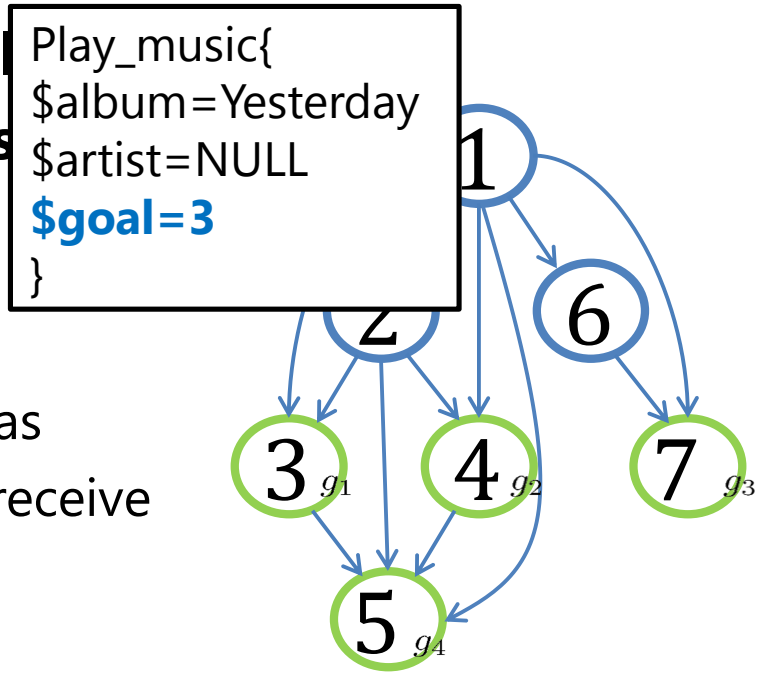
- **In case of user intention=3**
  - Should the system say "OK, I play the album of "yesterday"" or "Could you tell me the artist name?" (to node 5)
    - Depends on contexts and task relations



- **Set up user goals in dialogue states**
  - The system estimate the goal as a part of dialogue state given sequence of observation (w belief update or RNN)
  - Use states will be much more complicated with combinations of goals and existing states

Other actions

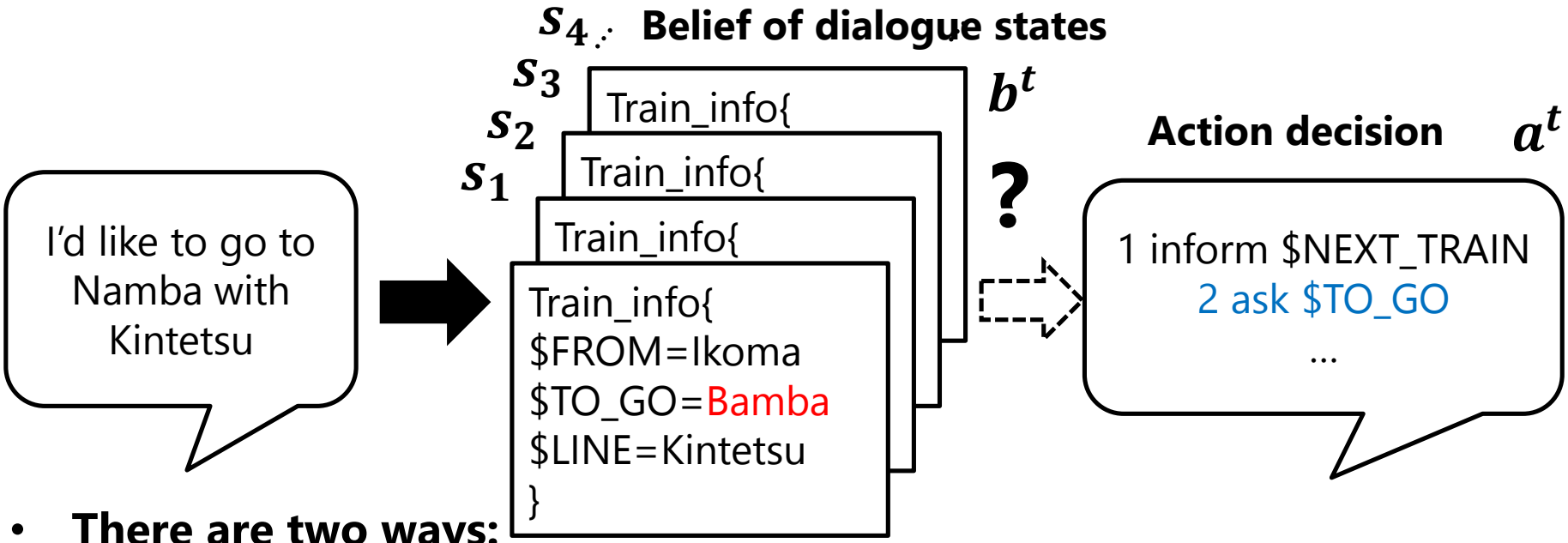
- **We already defined do a task (in goal and request information (in all nodes**
  - What is other possible actions?
- **Confirm**
  - If the system do the confirmation such as "could you say again?", the system can receive the observation again
    - Usually in higher confidence



- **Other possible actions:**
  Confirm the current intention, Confirm the goal, etc…

Visit again the problem of action decision with reinforcement learning

- **Decide action a^t given belief b^t**



- **There are two ways:**
  - **Find the best policy (policy gradient)**
  - **Find the best Q-function (Q-network)**
    - **To understand the model, let's see the example in s^t**

How do we get the good policy?

- **Maximize the expected future reward (value function)**
  - $V^{\pi^*}(s^t) = \max_{\pi} V^{\pi}(s^t)$

    $= \max_{a} \sum_{s^{t+1}} P\left(s^{t+1} | s^t, a^t\right) \left(R\left(s^t, \pi(s), s^{t+1}\right) + \gamma V^{\pi^*}\left(s^{t+1}\right)\right)$

    a
  - $Q^{\pi^*}(s, a) = \sum_{s^{t+1}} P\left(s^{t+1} | s^t, a^t\right) \left(R\left(s^t, a^t, s^{t+1}\right) +$

Q-learning

- **Premise: if we can calculate $Q(s,a)$ for every pair, we should decide action a according to $\max_a Q(s,a)$**
- **Problem: we don't know $P(s^{t+1}|s^t,a^t)$ to calculate $Q(s,a)$**
- **Solution: approximate the $P(s^{t+1}|s^t,a^t)$ with sampling**
  - $Q(s^t,a^t)$

    $$\xleftarrow[update]{} (1-\alpha)Q(s^t,a^t) + \alpha\left(R(s^t,a^t,s^{t+1}) + \gamma \max_{a^{t+1}} Q(s^{t+1},a^{t+1})\right)$$
  - Back-propagate the reward from the end of the sample

  - We will try to build a dialogue manager by using this algorithm in the next class

Q-learning algorithm (ε-greedy)

Initialize every pair of $Q(s,a)$

set ε (0<ε<1)

while update < threshold

observe s^t

if rand() < ε

the system takes the action a^t according to $\max_{a_m^t} Q(s^t, a^t)$

else

randomly select a^t

decide s^{t+1}

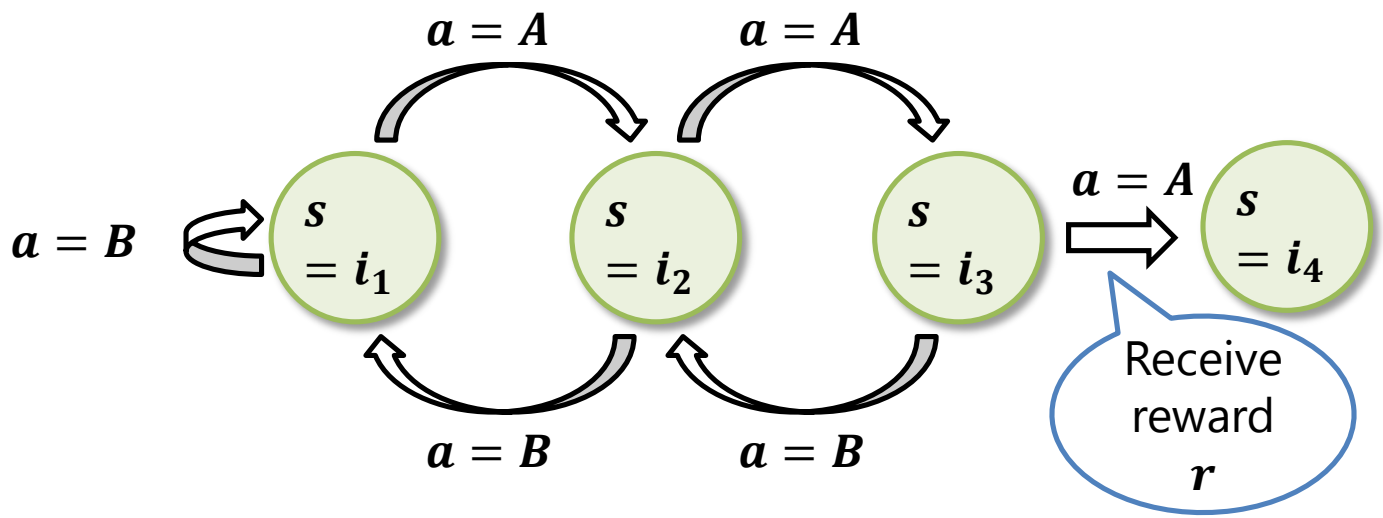
receive reward $R\left(s^t, a^t, s^{t+1}\right)$

$Q(s^t, a^t)$

$$\xleftarrow[update]{} (1-\alpha)Q(s^t, a^t) + \alpha\left(R\left(s^t, a^t, s^{t+1}\right) + \gamma \max_{a^{t+1}} Q(s^{t+1}, a^{t+1})\right)$$

end

Example: algorithm behavior of Q-learning



- **Let's think the simple example of Q-learning**
  - There are 4 $\boldsymbol{s}$ (1, 2, 3, 4) and 2 $\boldsymbol{a}$ (A, B)
  - The system can receive if $\boldsymbol{s^t}, \boldsymbol{a^t}, \boldsymbol{s^{t+1}} = \boldsymbol{i_3}, \boldsymbol{A}, \boldsymbol{i_4}$
    - The task will finish if the state reach to $\boldsymbol{i_4}$
  - Every state transition probabilities $\boldsymbol{P}\left(\boldsymbol{s^{t+1}} | \boldsymbol{s^t}, \boldsymbol{a^t}\right)$ are 1.0

Example: algorithm behavior of Q-learning

$s = i_1$ $s = i_2$ $s = i_3$

α: learning rate
γ: discount factor

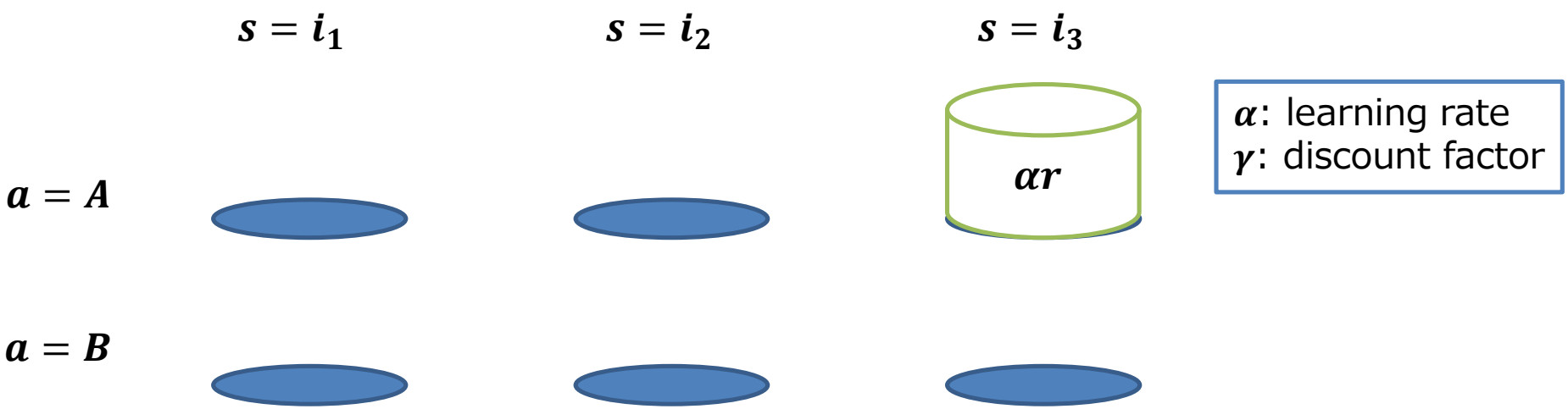
$a = A$

$a = B$

1. Initialize Q-value for every combination

- e.g. =0

Example: algorithm behavior of Q-learning



1. Initialize Q-value for every combination

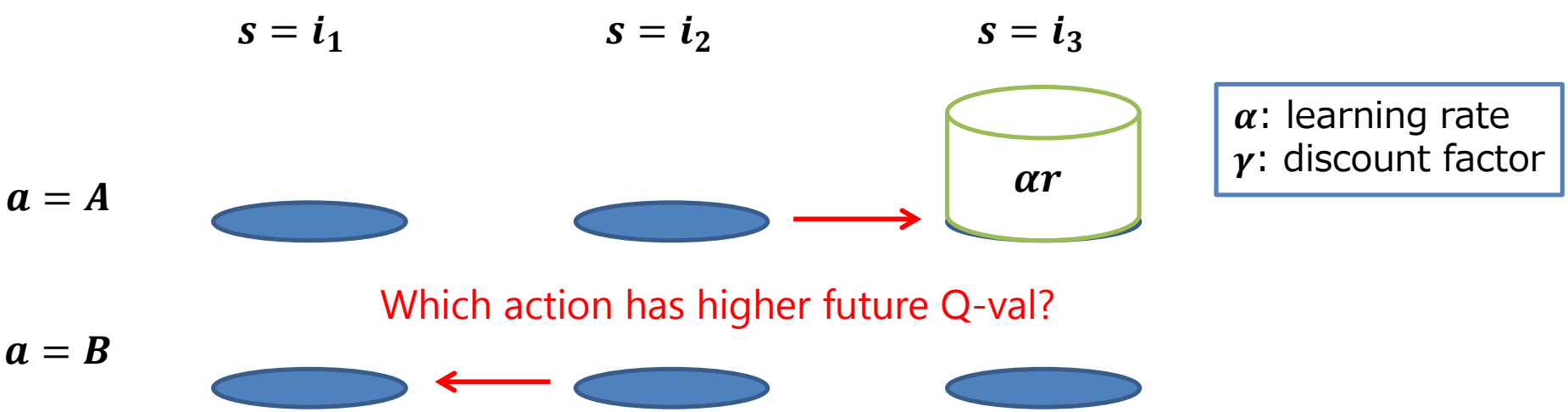
- e.g. =0

2. After some actions, the system received r in $s^t, a^t, s^{t+1} = i_3, A, i_4$

- The reward is discounted with learning rate

$$Q(s^t, a^t) \xleftarrow[update]{} (1-\alpha)Q(s^t, a^t) + \alpha\left(R(s^t, a^t, s^{t+1}) + \gamma \max_{a^{t+1}} Q(s^{t+1}, a^{t+1})\right)$$

Example: algorithm behavior of Q-learning



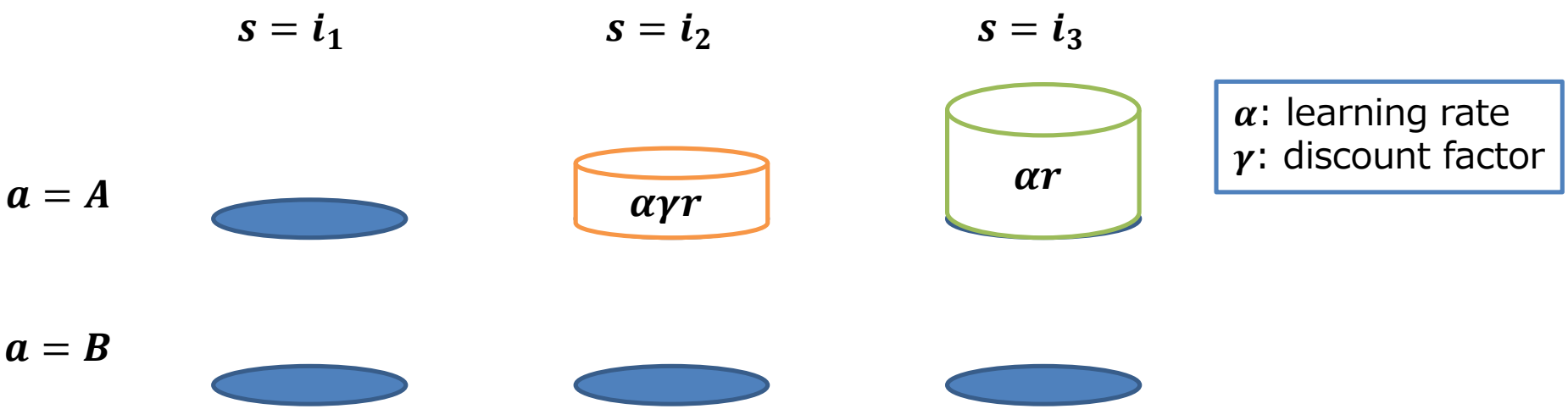
2. After some actions, the system received r in $s^t, a^t, s^{t+1} = i_3, A, i_4$

– The reward is discounted with learning rate

$$Q(s^t, a^t) \xleftarrow[update]{} (1-\alpha)Q(s^t, a^t) + \alpha\left(R(s^t, a^t, s^{t+1}) + \gamma \max_{a^{t+1}} Q(s^{t+1}, a^{t+1})\right)$$

3. When the system reach to $s = i_2$ in the next trial the system selects according to $\max_{a^{t+1}} Q(s^{t+1}, a^{t+1})$

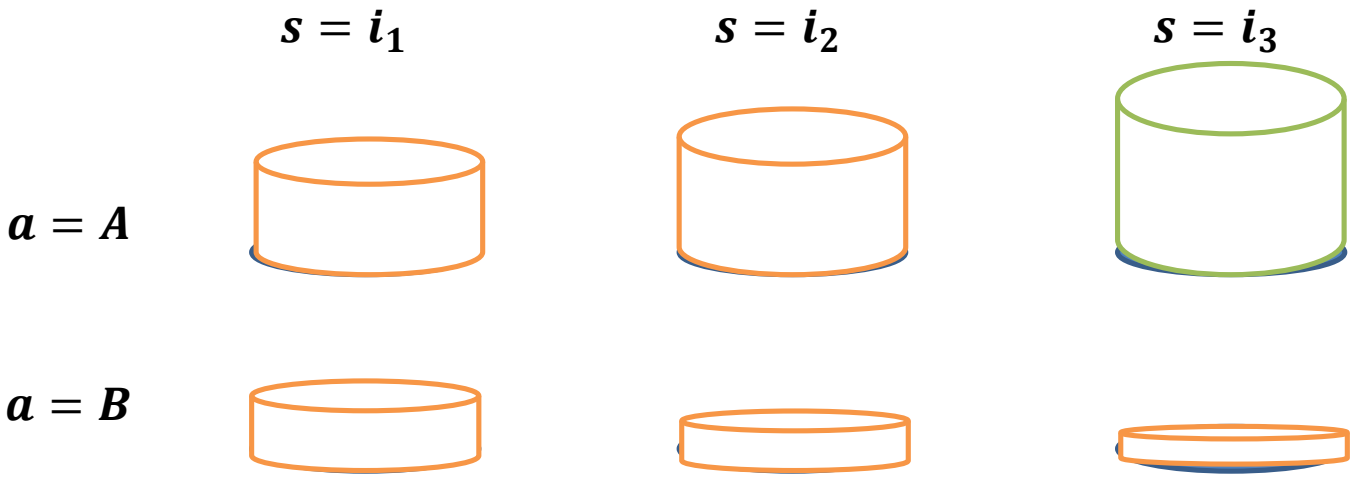
Example: algorithm behavior of Q-learning



3. When the system reach to $s = i_2$ in the next trial the system selects according to $\max_{a^{t+1}} Q(s^{t+1}, a^{t+1})$

4. The Q-value is stacked on the current $Q(s^t, a^t)$ with discount factor

Example: algorithm behavior of Q-learning



α: learning rate
γ: discount factor

3. When the system reach to $s = i_2$ in the next trial the system selects according to $\max_{a^{t+1}} Q(s^{t+1}, a^{t+1})$

4. The Q-value is stacked on the current $Q(s^t, a^t)$ with discount factor

5. After the large number of trials, Q-values for each combination will be converged

Problem of Q-learning

Initialize every pair of $\mathrm{Q(s,a)}$

set ε (0<ε<1)

while update < threshold

observe s^t

if rand() < ε

the system takes the action a^t according to $\max_{a_m^t} Q(s^t, a^t)$

else

randomly select a^t

decide s^{t+1}

receive reward $R(s^t, a^t, s^{t+1})$

$Q(s^t, a^t)$

$$\xleftarrow[update]{} (1-\alpha)Q(s^t, a^t) + \alpha\left(R(s^t, a^t, s^{t+1}) + \gamma \max_{a^{t+1}} Q(s^{t+1}, a^{t+1})\right)$$

end

How do we decide s?
How to define rewards?

User simulator

- **User simulator decides s at each turn**
  - Very simple model works on $\boldsymbol{P}(\boldsymbol{s}^{t+1}|\boldsymbol{s}^{t},\boldsymbol{a}^{t})$
    - But we can calculate actual $\boldsymbol{Q}^{\pi^*}(\boldsymbol{s},\boldsymbol{a})$ without Q-learning, if we know $\boldsymbol{P}(\boldsymbol{s}^{t+1}|\boldsymbol{s}^{t},\boldsymbol{a}^{t})$:P
- **There are several approaches as other modules**
  - Heuristics

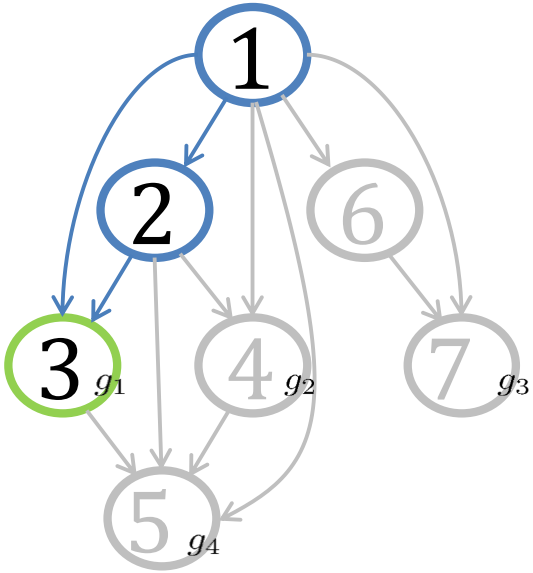
    Agenda-based user simulation for bootstrapping a POMDP dialogue system, Schatzmann et al., In Proc. NAACL, 2007
    - Agenda-based: The simulator assumes goal and agenda
    - IDG system: The simulator assumes goal and transitions to goals
  - Data driven approaches:
    - N-gram based utterance generator,

      Data-driven user simulation for automated evaluation of spoken dialog systems. Jung et al., CSL, 2009
    - neural network based

      A Sequence-to-Sequence Model for User Simulation in Spoken Dialogue Systems. Asri et al., In Proc INTERSPEECH, 2016

If we use simple assumption in IDG…

- **The user firstly decide their goal g**
  - The goal is 3 in the example
- **The user says the first utterance as any node between the goal node and the root node**
  - It will be 1, 2 or 3 in the example
- **If the system confirms with the user, the user repeats the previous utterance (node) again**
  - Required if the system assumes belief (=confidence of each state)



Reward definition

- **Task completion**
  - This is the most simple and easy to understand
  - E.g. system receives +10 rewards in task successes, -10 penalties in task fails and -1 penalty on each turn
- **User satisfaction**
  - Regression result of the user satisfaction
  - It works comparably to the task completion

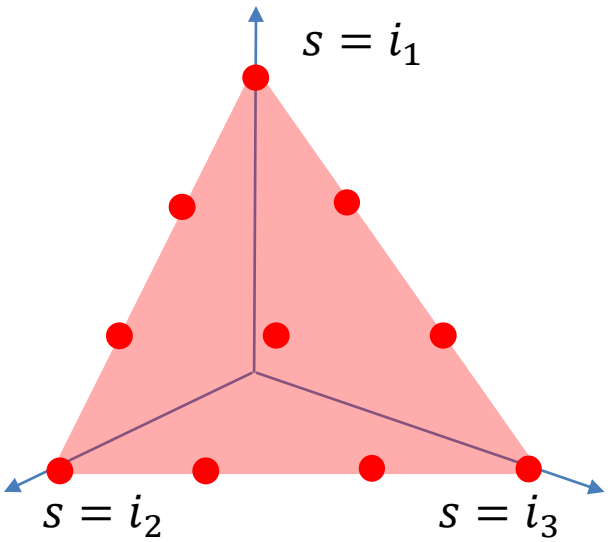
Reward-Balancing for Statistical Spoken Dialogue Systems using Multi-objective Reinforcement Learning. Stefan et al., In Proc SIGDIAL 2017

- **Inversed reinforcement learning**
  - Calculate the reward from the expert data (dialogue data of wizard of Oz; dialogue system acted by human)

User simulation in dialogue systems using inverse reinforcement learning. Chandramohan et al., In Proc INTERSPEECH 2011

Other remaining problems

- **How do we decide belief point to be sampled as $\boldsymbol{b}$?**
  - It is hard to Update any $\boldsymbol{Q}(\boldsymbol{b}, \boldsymbol{a})$, because $\boldsymbol{b}$ is not a point as $\boldsymbol{s}$
    - It will be hyper plain
- **Grid-based value iteration**
  - Decide belief points with grid
- **Point-based value iteration**
  - Decide belief points from sampling of data
- **Regression (Q-network)**
  - If we can develop a regression to calculate $\boldsymbol{Q}(\boldsymbol{b}, \boldsymbol{a})$, it can calculate $\boldsymbol{Q}(\boldsymbol{b}', \boldsymbol{a})$ (if the model is successfully trained)



Solutions for implementation problems in the IDG example

- **States are defined from dialogue frames, and simple actions are set up**
  - Of course, you can develop much more complicated intention graph
- **User simulator can be developed under very simple assumption**
  - Page 28
- **Belief update will be conventional belief update or RNN**
  - $\boldsymbol{b^t \approx P(o^t|s^t)\sum_{s_i} P\left(s_j^t|s_i^{t-1}\right)b^{t-1}}$
  - $\boldsymbol{h^t = \tanh(W_{Xh}X^t + W_{hh}h^{t-1} + c_h)}$
  - $\boldsymbol{b^t = \mathrm{softmax}(W_{hb}h^t + c_b)}$
- **Reward functions are task completion**
- **Problem of belief point sampling will be solved by Q-network**