
FROM LANGUAGE TO GOALS-
INVERSE REINFORCEMENT
LEARNING FOR VISION-BASED
INSTRUCTION FOLLOWING
Hyungrak Kim

Content

*Abstraction

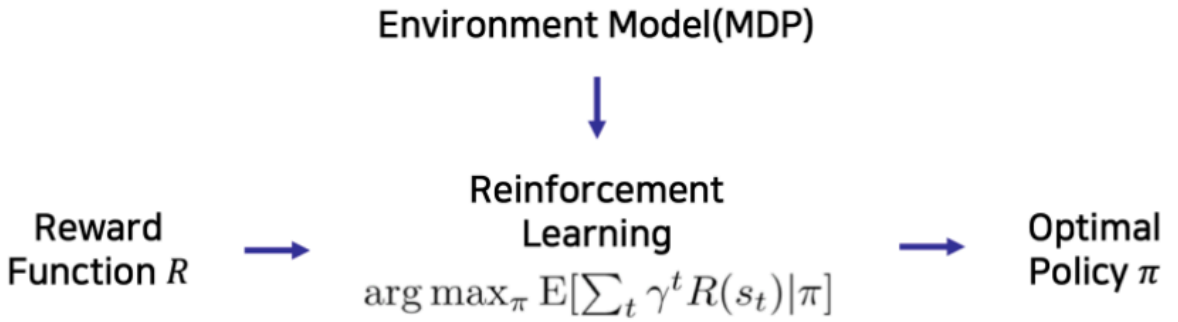
- Robot이나 자율주행자동차에서 Reward Function을 만들기 어려움
- Reward가 Sparse하고, 그 Reward자체가 적절한지 의문
- Grounding Language 를 이용해서 IRL을 하여 Reward function을 만드는 것을 제안
- 기존 Language Condition Policy 보다 Language-Condition Reward가 새로운 환경에 더 적합함
- 따라서 Language Condition Reward Learning(LC-RL)를 제안해서 Reward function을 만듬

1. Introduction

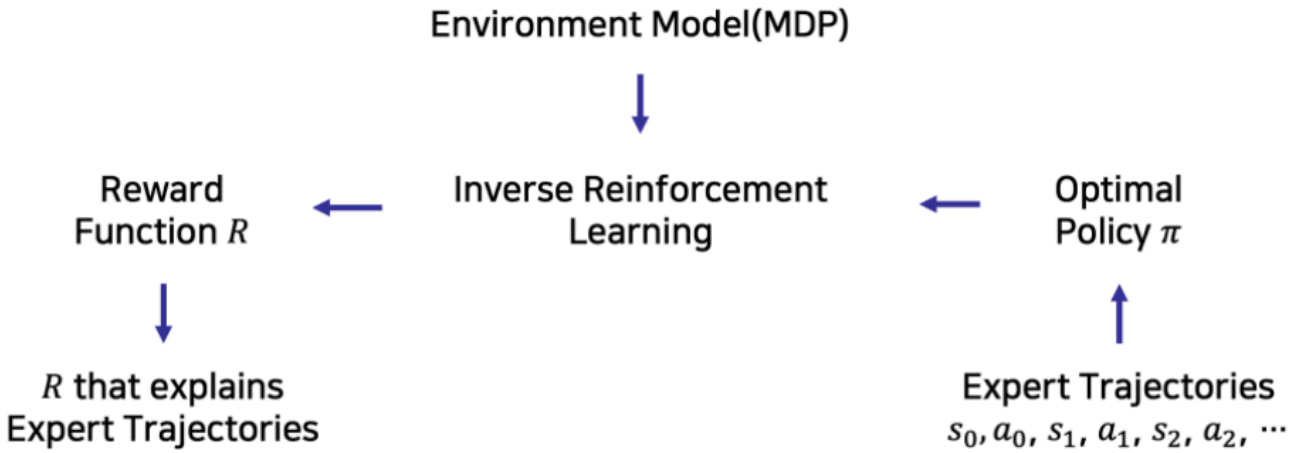
- Reward는 강화학습에서 학습을 하게 해주는 가장 중요한 Fact
- 기존 Reward를 학습하기 위해서 어려운 문제가 많다.
- 특히 State뿐 아니라 Language command를 이해한 Goal task 를 해결하기 위해서 어떻게 해야하는지 고민
- Language command를 Reward를 주는데 활용하고 싶음
- IRL에서 Natural Language command를 적용해서 Reward Function을 만듬
  - IRL –> Reward inference
- MaxEnt IRL 사용
  - 학습에 Dynamic programming 사용
- Evaluation Data
  - SUNGG Data set 사용
    - Pick and Place Task
- Contribution:
  - First language-conditioned inverse reinforcement learning to environments with image observations and deep neural networks,
  - Reward generalize to novel task and Environmet

1. Introduction

- Reinforcement Learning



- Inverse Reinforcement Learning



2. Related Work

- The Principle Maximum Entropy :
  - 어떠한 Distribution의 Entropy를 최대화 하면 최악의 Policy를 고를 확률이 줄어듬
  - Task에 맞는 Reward를 만드는 Function을 잘 만들수 있겠다는 말

- MaxEnt IRL : Reward Function을 최대화하는 Parameter θ 를 찾겠다

- behavioral cloning

- GAIL

3. Background

- MaxEnt IRL

$$\max_{\theta} E_{\tau \sim \mathcal{D}} [\log p_\theta(\tau)]$$

- Optimal Trajectories는 exponentiated returns에서 Probabilities Proportion으로 관찰됨

$$p(\tau) \propto \exp\{r(\tau)\}$$

- Reward Function

$$\nabla_\theta E[\log p_\theta(\tau)] = \sum_{s,a} (\rho^{\mathcal{D}}(s,a) - \rho^*_\theta(s,a)) \nabla_\theta r_\theta(s,a) \ ,$$

- $\rho^{D}(s, a)$ represents the state-action marginal of the demonstrations
- $\rho^{*}\theta\ (s, a)$ represents the state-action marginal of the optimal policy under reward $r^{\theta}\ (s, a)$.

4. Multi-Task IRL

- Multi-Task IRL

$$\max_{\theta} E_{\xi}[E_{\tau_\xi}[\log p_\theta(\tau_\xi, c_\xi)]]$$

- c: context

Algorithm 1 Language-Conditioned Reward Learning (LC-RL)

1: Obtain expert demonstrations and language describing the goal.
2: Initialize reward function r_θ.
3: **for** step t in $\{1, \ldots, \mathrm{N}\}$ **do**
4: Sample task ξ, demonstrations d_ξ , and language $\mathcal{L}_\xi$.
5: Compute optimal $q^*(s, a)$ using q-iteration and $\rho^*(s, a)$ using the forward algorithm.
6: Update reward r_θ with the gradient $(\rho^{d_\xi}(s, a) - \rho^*(s, a))\nabla r_\theta(o, a, \mathcal{L}_\xi)$
7: **end for**

5. Language Condition Reward Learning

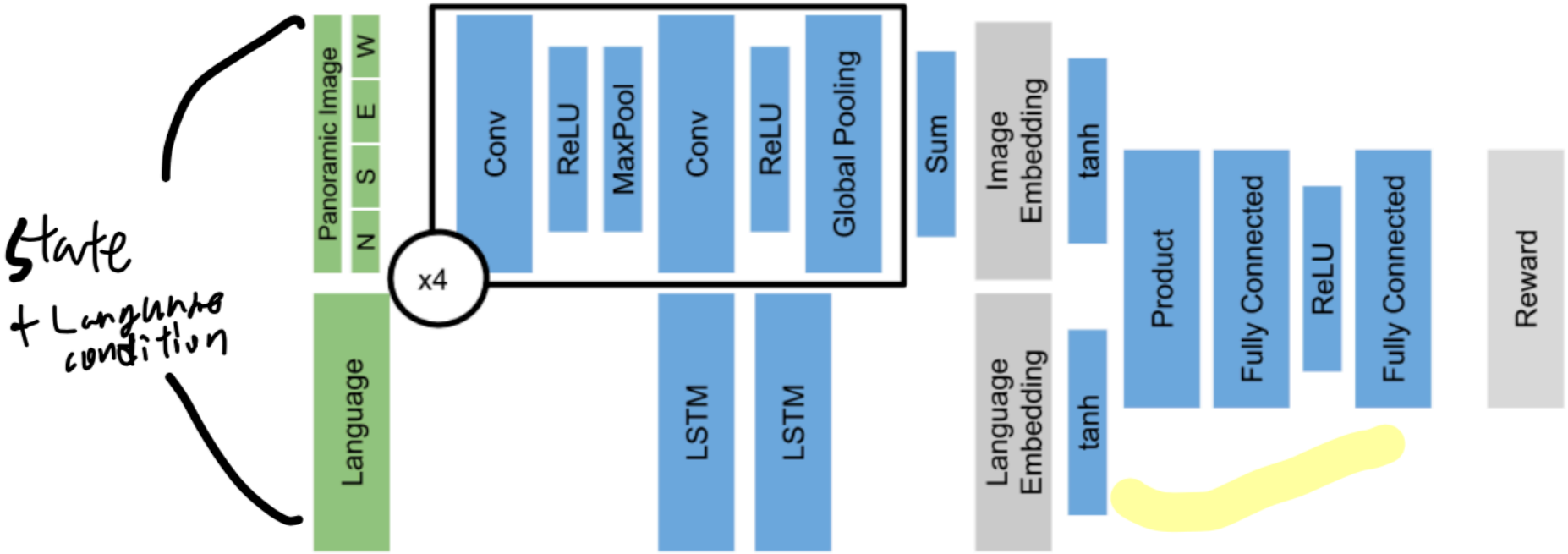


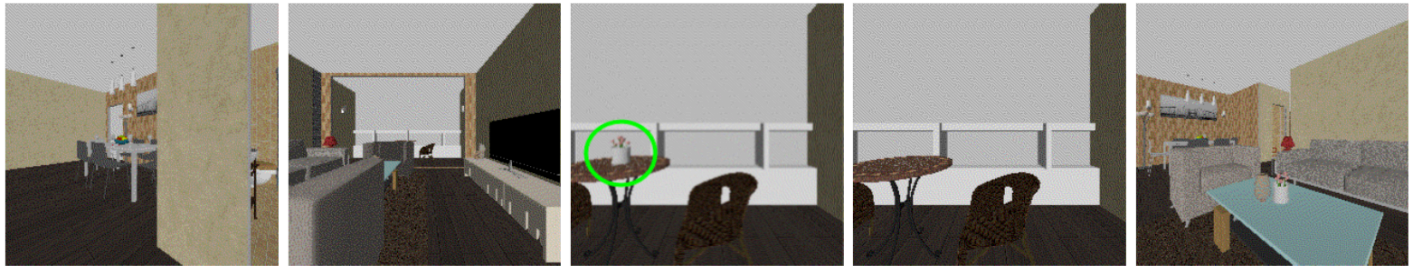
Figure 2: Our reward function architecture. Our network receives as input a panoramic semantic image (4 views) and a language command represented as a sequence of one-hot word vectors, and outputs a scalar reward.

6. Evaluation

- Task
  - (1)Navigation
  - (2)PICk

Figure 3: An example task. The green segment corresponds to the solution of a NAV task, ”go to the cup”, where the cup is circled in green. The green plus blue segments represents a path for the PICK task, ”move the cup to the bed”, where the bed is circled in blue.

Figure 4: Example first-person RGB (left) and semantic (right) images from the bedroom inside the house depicted in Figure 3. We only use the semantic labels as input to our model.



6. Evaluation

- Dataset
  - Task(Training): Same House, novel combinations of object and location
  - House(Test, Dev): new Houses
- 1413 Task(716 PICK, 697 NAV)
  - 14 object, 76 difference house Layout
  - 1004 task(71%) training set
  - 236(17%) task test set
  - 173(12%) house test set

6. Evaluation

Table 1: Success rates (in percentages) across task categories. Each result is averaged over 3 seeds. Test-Task refers to testing on novel tasks within the same houses as training, whereas Test-House refers to testing novel tasks in novel houses. The AGILE method is described in (Bahdanau et al., 2018)

	Train			Test-Task			Test-House		
	PICK	NAV	Total	PICK	NAV	Total	PICK	NAV	Total
Optimal Policy Cloning	20.7	61.6	40.3	10.1	29.4	19.6	0.0	17.2	8.5
AGILE	0.0	40.9	18.0	0.0	34.1	16.8	0.0	30.6	15.1
GAIL-Exact	59.4	**73.5**	**66.9**	49.1	**50.4**	49.8	23.5	35.4	28.3
LC-RL (ours)	**63.8**	69.7	**66.9**	**56.7**	47.8	**51.9**	**32.1**	**39.4**	**36.4**
Reward Reg. (Oracle)	87.0	85.0	86.1	82.5	67.0	74.1	70.6	62.3	65.7

6. Evaluation

Table 2: Success rates (in percentages) on using DQN to re-optimize learned rewards. For reference, we also include Q-iteration results (labeled QI) from Table 1 as an oracle comparison.

	Shaping	Train	Test-Task	Test-House
LC-RL (DQN)	Yes	12.9	14.1	14.9
	No	8.0	7.7	8.0
LC-RL (QI)	-	66.9	51.9	36.4
Reward Regression (DQN)	Yes	54.7	58.9	57.5
	No	7.5	8.3	9.2
Reward Regression (QI)	-	86.1	74.1	65.7

ject (during PICK tasks) is held by the agent or not. Each indicator is transformed by an embedding lookup, and all embeddings are element-wise multiplied along with with the language and image embeddings in the original architecture.

6. Evaluation

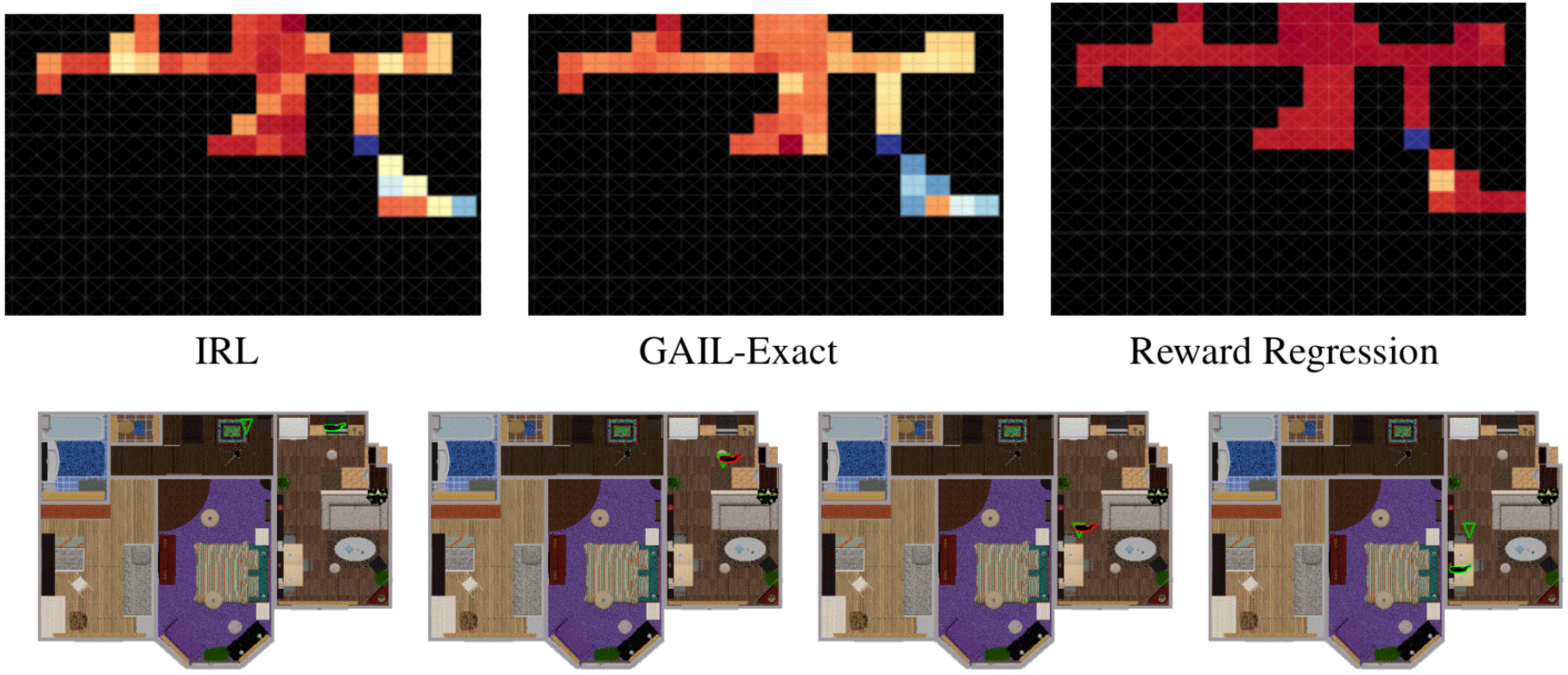


Figure 7: Learned rewards and a corresponding birds-eye view rollout for the task ”move pan to living room”.

7. Conclusion

- GAIL보다 조금 좋다는 평가
- 새로운 Language Condition 을 IRL에 접목하는 방법론 제시
- 그래서 실제 서비스에 적용할 수 있을까??
- IRL필요하긴 하지만, 사람보다 성능이 좋아야하는데 그렇게 보이지 않는다

Q&A

Thank you