CURIOSITY-DRIVEN REINFORCEMENT LEARNING FOR DIALOGUE MANAGEMENT

Paula Wesselmann, Yen-Chen Wu

Cambridge University Engineering Department
Trumpington Street, Cambridge CB2 1PZ, UK

*Milica Gašić**

Saarland University
66123 Saarbrücken, Germany

ABSTRACT

In this paper we describe the use of curiosity rewards for dialogue policy learning of goal oriented dialogues via reinforcement learning. Using curiosity improves state-action space exploration and helps overcome reward sparsity. Additionally, for goal oriented dialogues it makes sense to perform inherently curious actions in order to gain knowledge about the user goal. We show that intrinsic curiosity rewards can replace random ϵ-greedy exploration and stabilize training. The best results are achieved when curiosity rewards are combined with ϵ-greedy exploration.

***Index Terms*—** curiosity-driven, reinforcement learning, dialogue management, intrinsic rewards, exploration

1. INTRODUCTION

Initially dialogue systems were build using inflexible handcrafted decision rules, which brings limitations and does not allow for these systems to behave intelligently. Reinforcement learning (RL) allows learning from interaction by maximizing rewards [1], thus removing the need for manual rules. The problem is that these rewards are often noisy, sparse or completely non-existent. Therefore, the use of *intrinsic rewards*, generated by the system itself, is proposed [2], [3].

An agent learning via RL, learns to behave my maximizing the expected sum of rewards. Usually rewards are given to the system externally, reinforcing good behavior and/or punishing bad behavior. In the context of dialogue systems the rewards come from the users, who rate the quality of the dialogue. Users often do not like to give feedback and have different perceptions of good and bad, which results in sparse and inconsistent rewards. In order to learn desirable behavior it is beneficial for a *spoken dialogue system* (SDS) to have an intrinsic source of reward. Intrinsic curiosity is a consistent reward signal and also helps replace inefficient ϵ-greedy methods for exploration. Curiosity-driven learning does not require random exploration, since the agent acts with the aim to explore new belief-states and such exploration is key for data efficient learning.

*This work was partly funded by a Google Faculty award and an Alexander von Humboldt Sofja Kovalevskaja award. The code is available at `pydial.org`

We combine the current work on deep RL for dialogue policy optimization for goal oriented dialogues [4] with work on self-supervised state prediction errors as intrinsic curiosity reward signal [5]. We use belief-state prediction error as an intrinsic reward signal for the dialogue management (DM) module of the *PyDial* dialogue system [6] with ACER policy learning as introduced by [4]. If the system is able to correctly predict what belief-state it will be in after performing an action, that action was not curious and no reward is given. But if the system learns something new, that is, it was not able to predict the next belief-state, the prediction error is given as a reward for being 'curious'.

2. DIALOGUE MANAGEMENT

The DM is the core component responsible for the system's behavior. It is made up of two units, the **Belief Tracker** and the **Policy**. The DM can be described as the brain of the SDS, where belief tracking is responsible for memory, and policy for making decisions, sending signals on what actions to take, which are then executed.

The policy regulates what actions are executed given the system's current knowledge or *belief-state*. The DM has to deal with uncertainty coming from the automatic speech recognition (ASR) and natural language understanding (NLU) units of the SDS as well as from the users directly. Therefore the states are belief-states, accounting for uncertainly in the system. Policy π is a deterministic decision rule mapping a belief-state b_t into action $a_t = \pi(b_t)$. In every turn of the dialogue the dialogue tracker updates its belief about the users goal and its memory about the dialogue, the belief-state b. The current belief-state b_t is the input for the policy to generate a response to the user, a_t. The policy is learned via deep RL, and is chosen to maximize the total reward; that is we chose the policy with the optimal Q-function:

$$Q(b,a) = \mathbb{E}\Big[\sum_{k=0}^{T-t} \gamma^k r_{t+k} | b_t = b, a_t = a\Big] \tag{1}$$

$$\pi^*(b) = \arg\max_a Q^*(b,a). \tag{2}$$

where T is the last turn of the dialogue, t is the current turn and γ is the discount factor for future rewards. The

Q-function returns the expected sum of rewards given belief-state b and action a. $\pi^*(b)$ is the optimal policy, meaning it gives the best action a to take when in belief-state b, in order to maximize the expected return.

The *actor critic with experience replay* (ACER) method [4], approximates both, the policy-function π (actor) and Q-function (critic) as Deep Neural Networks (DNNs). This is an off-policy actor critic method that alternates between actor improvement, which aims to improve the current policy, and critic evaluation, which evaluates the current policy with the Q-function. Experience replay is used to increase the sample-efficiency of the algorithm. The importance sampling ensures the accuracy of the estimates and the trust region policy optimization (TRPO) stabilizes the learning. In its original form it utilizes ϵ-greedy exploration.

3. INTRINSIC MOTIVATION

Human learning and development often is not goal oriented, but driven by intrinsic motivation such as curiosity. Curiosity is a motivation to explore the unknown, searching for new knowledge and continuous improvement. Curious or exploratory behavior enables an agent to learn about its environment and relationship between the agent's actions, its current belief-state, and its environment. Therefore, by being curious an agent is able to gain new knowledge and skills, even when the rewards are rare or deceptive [7]. Intrinsic curiosity for RL was first introduced in [8]. Since then different metrics to measure curiosity have been introduced and curiosity-driven RL is successfully used for many different machine learning problems.

4. RELATED WORK

State prediction error is a popular metric used for intrinsic curiosity rewards in different RL applications [8, 9, 10, 5]; it is used to overcome reward sparsity and for efficient state-action space exploration for tasks such as gaming and learning robotic motor skills, but has not been applied to dialogue policy learning yet. Closely related measures of curiosity used as intrinsic reward signals are prediction uncertainty [11, 12, 13] and improvement [14]. Both approaches also train a forward model, learning about the environment while training the policy. Other measures of curiosity include count based exploration with tabular RL and pseudo state visitation counts derived from density models [15], allowing to generalize count-based exploration to non-tabular cases.

5. CURIOSITY REWARDS FOR DIALOGUE MANAGEMENT

Intrinsic motivation is generated by the agent itself, using an Intrinsic Curiosity Module (ICM). The intrinsically generated reward can be used in combination with extrinsic reward signals coming from the environment. The ICM is a part of the DM and outputs a reward signal at every dialogue turn (as illustrated in Fig. 1). The ICM takes action a_t, belief-state b_t, and belief-state b_{t+1} as inputs. Output of the ICM is the state prediction error, which can be used as reward for curiosity.

The ICM is adapted from [5]. This section partly mirrors their method for "Self-supervised prediction for exploration". The ICM (Fig. 1) consists of an inverse model, predicting the action a_t given states b_t and b_{t+1}, which is used to optimize the belief-state feature encoding and a forward model predicting future state $\phi(b_{t+1})$ given action a_t and state $\phi(b_t)$. The prediction error is the L^2-norm of difference between $\hat{\phi}(b_{t+1})$ and $\phi(b_{t+1})$ (see Equation 5).

5.1. Inverse Model

Using an inverse model to predict states in a learned feature space was proposed by [5]. It deals with the prediction of raw images that include random, unpredictable features. Learning a feature space helps to focus on features that are essential in order to make a good prediction. This feature space is learned by training a DNN with two sub-modules: first encoding raw state b_t into a feature vector $\phi(b_t)$ and second taking $\phi(b_t)$ and $\phi(b_{t+1})$ as feature encoded inputs and predicting action a_t taken to move from belief state b_t to b_{t+1}:

$$\hat{a}_t = g(\phi(b_t), \phi(b_{t+1}); \theta_I) \tag{3}$$

Function g is learned by training a NN, where $\hat{a}_t$ is the predicted estimate of action a_t and the network parameters θ_I are trained to minimize the discrepancy between predicted and actual actions, $L_I(\hat{a}, a_t)$. In our case the output of g is a soft-max distribution across all possible summary actions.

When using curiosity rewards for belief state-action space exploration in DM, a good feature representation is more general, e.g. a feature represents that the food type has been informed, rather than the specific type of food that has been informed. The system cannot learn to predict which food the user is going to request, but it can learn that after requesting information about which food a user wants, there is a high probability that the system will be informed about the food type. On the other hand, when using curiosity rewards to encourage curious actions, the full belief-state vector is needed and for every new dialogue we want the agent to be curious about the new user goal and specific slots/ features.

5.2. Forward Model

Core element of the ICM is the *forward model*, a NN that is trained to predict the next state b_{t+1} in the feature space, given the feature encoding of the current belief-state $\phi(b_t)$ and the action a_t executed in this state:

$$\hat{\phi}(b_{t+1}) = f(\phi(b_t), a_t; \theta_F) \tag{4}$$

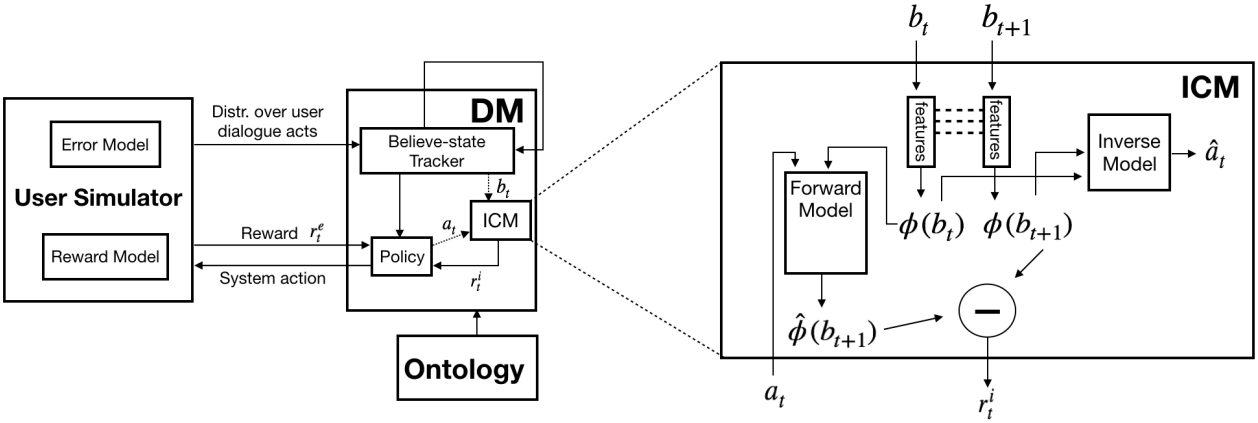


Fig. 1: Illustrated formulation for self-supervised prediction as curiosity in context with the DM. In belief-state b_t the agent interacts with the user by executing an action a_t sampled from policy π to get to state b_{t+1}. The ICM encodes belief-states b_t and b_{t+1} into features $\phi(b_t)$ and $\phi(b_{t+1})$, that are trained to predict a_t (inverse model). a_t and $\phi(b_t)$ are inputs for the forward model predicting the feature representation $\hat{\phi}(b_{t+1})$ of b_{t+1}. The prediction error is used as intrinsic reward signal r_t^i which can be used in addition to external rewards r_t^e. (this model is adapted from [5])

where $\hat{\phi}(b_{t+1})$ is the predicted estimate of $\phi(b_{t+1})$. The network parameters θ_F are trained to optimize:

$$\min_{\theta_F} L_F\big(\phi(b_{t+1}), \hat{\phi}(b_{t+1})\big) = \frac{1}{2}||\hat{\phi}(b_{t+1}) - \phi(b_{t+1})||_2^2 \quad (5)$$

The intrinsic reward signal is the prediction error multiplied by a scaling factor η, $\eta > 0$:

$$r_t^i = \frac{\eta}{2}||\hat{\phi}(b_{t+1}) - \phi(b_{t+1})||_2^2 = \eta L_F \quad (6)$$

Forward and inverse dynamic losses are jointly optimized:

$$\min_{\theta_I, \theta_F} \Big[(1-\beta)L_I + \beta L_F\Big] \quad (7)$$

where $0 \leq \beta \leq 1$ is weighting the inverse model loss against the forward model loss.

5.3. Intrinsic Curiosity Module without feature encoding

In addition to the ICM described above, we use a simpler ICM using the same principle of belief-state prediction error as intrinsic reward, but without feature encoding and therefore without the need for an inverse model. The *forward model* now uses raw belief-sates b_t and b_{t+1} directly. Equation 4 for the NN becomes: $\hat{b}_{t+1} = f(b_t, a_t; \theta_F)$ and the parameters θ_F are trained to minimize $L_F(b_{t+1}, \hat{b}_{t+1})$.

5.4. Not-informative action penalty

Instead of rewarding curious actions, one can penalize actions that have predictable outcomes, as it is the case when the system repeats itself. For this approach the prediction error is calculated as the cosine loss:

$$L_F = 1 - \frac{\phi(b_{t+1}) \bullet \phi(\hat{b}_{t+1})}{||\phi(b_{t+1})|| \cdot ||\phi(\hat{b}_{t+1})||} \quad (8)$$

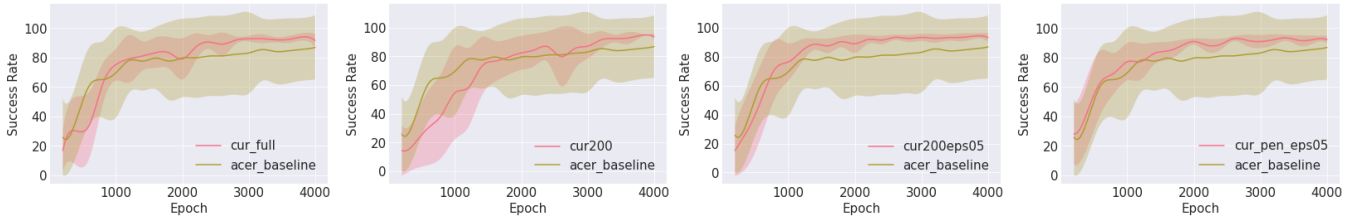
where the dot-product of the belief-state $\phi(b_{t+1})$ and its prediction $\phi(\hat{b}_{t+1})$ is divided by the product of their magnitudes. A fixed penalty is assigned when L_F falls below threshold w.

5.5. Pre-trained Curiosity vs. Jointly Trained Curiosity

When jointly training the curiosity model and policy, in the early stages of training, curiosity rewards are high for all actions. In the later stages, curiosity rewards will only remain for actions that prompt unpredictable reactions from the user. We want the curiosity reward to be used for efficient exploration. The reward given while the ICM learns to make predictions and learning a feature space is not accurate. Training the dialogue policy and ICM together means that dialogue policy learning is trying to optimize those random rewards at the beginning of training, slowing down policy learning. We propose pre-training the ICM on a small data-set to make learning more data efficient.

6. EXPERIMENTS

We compare performance of the same policy learning algorithm with and without intrinsic curiosity rewards. We also compare results to the state-of-the-art GP-SARSA algorithm [16]. All dialogues are in the Cambridge Restaurant domain (where users can access information about restaurants in Cambridge, UK), the domain includes 100 venues with 6 slots each, out of which 3 are requestable by the system. The SDS in this domain has a belief-state vector of size 268 and



(a) no feature encoding (b) with feature encoding (c) curiosity + ϵ-greedy (d) non-info. action penalty

Fig. 2: Learning curves of the success rates for four different settings for the use of intrinsic curiosity rewards.

	4000 trg dialogues		1000 pre-trg +3000 trg	
	GP-SARSA	ACER	no feat.	cur+eps
Reward	9.8	10.2	11.4	11.4
SR (%)	90.8	86.9	92.9	**93.3**
Turns	8.4	7.2	7.1	7.3

Table 1: Final rewards, success rates and number of turns.

can perform 16 summary actions including actions such as `inform, request_food` or `bye`. We use a simulated user with semantic error rate (SER) of 15%. The simulated user is comprised of a behavior component to produce a semantic act, and an error simulator, to produce a list of semantic hypotheses derived from the semantic act, simulating the error coming from the ASR and NLU channels in a SDS [17]. The external reward received for a completed dialogue that successfully reached the user goal (i.e. provided the desired information about venue along with additional details such as phone number or address) is $r_e = 20 - n$, where n is the number of turns in a dialogue. Unsuccessful dialogues only receive the turn penalty $r_e = -n$. When using the curiosity methods, total reward received is $r_t = r_e + r_i$.

Four broad settings are evaluated: (a) no feature encoding; (b) feature encoding; (c) curiosity + ϵ-greedy;and (d) ϵ-greedy + not-informative action penalty. The training is done in 20 intervals of 200 dialogues each, after every interval the current policy is tested with 500 dialogues. Where ϵ-greedy exploration is used, it is set at $\epsilon = 5$, meaning that percentage of the time the action is chosen at random. No random actions are performed during testing. In the plots the shaded area depicts the mean $\pm$ the standard deviation over 10 different random seeds. (Note that average rewards for the baseline and different curiosity methods are not comparable.)

(a) **No feature encoding:** We apply the prediction error for predicting raw belief-states as curiosity reward. Results (Fig. 2a) show that using this curiosity method is somewhat less data efficient then the ACER baseline, but converges to a higher final success rate of 92.9% (see Table 1) with less instability in the final results. The final average number of turns in a dialogue is 7.1.

(b) **With feature encoding:** By adding feature encoding for the belief-state prediction, variance in success rates during the first 2000 dialogues of training increases, but final results are stable (Fig. 2b) at around 93%. Feature encoding does not seem to improve predictions, but rather makes the model more complex and less data efficient. The final average number of turns in a dialogue is 7.3.

(c) **Curiosity + ϵ-greedy:** When using the curiosity reward for exploration in addition to ϵ-greedy exploration, learning becomes more stable and data efficient (Fig. 2c). This method achieves with 93.3% the highest final average success rate out of our experiments (see Table 1). The final average number of turns in a dialogue is 7.3.

(d) **ϵ-greedy + not-informative action penalty:** The idea of giving a penalty for accurate predictions is to solve the observed problem of the system unnecessary repeating itself and asking questions the user has already informed the system about especially in the early stages of learning. Results (Fig. 2d) show that the learning becomes more data efficient. The final average success rate is 92.1%, similar to the other curiosity reward methods. The final average number of turns in a dialogue is 7.1, the lowest together with method (a).

7. CONCLUSION

This work focuses on the application of intrinsic curiosity rewards and penalties related to belief-state prediction error for dialogue management in addition to external rewards given for the completion of successful dialogues. We are able to show an improvement in success rates when using curiosity over ϵ-greedy exploration or as additional exploration tool. Experimenting with intrinsic reward modeling, we find fixed intrinsic penalties for non-curious actions to be more efficient than rewards for curious actions. This shows that it is easier to learn to avoid specific behavior than to learn to behave in new undiscovered ways. Future steps would be to optimize the scaling factor for curiosity rewards (or penalties) and improve the belief-state prediction model. It would also be interesting to see how this method works with real user interactions.

8. REFERENCES

[1] Steve Young, Milica Gasic, Blaise Thomson, and Jason D. Williams, "POMDP-based statistical spoken dialog systems: A review," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1160–1179, may 2013.

[2] Marlos C. Machado and Michael Bowling, "Learning Purposeful Behaviour in the Absence of Rewards," *ICML-16 Workshop on Abstraction in Reinforcement Learning*, 2016.

[3] Tejas D. Kulkarni, Ardavan Saeedi, Simanta Gautam, and Samuel J. Gershman, "Deep Successor Reinforcement Learning," *arXiv e-prints*, p. arXiv:1606.02396, June 2016.

[4] Gellert Weisz, Pawel Budzianowski, Pei-Hao Su, and Milica Gašić, "Sample efficient deep reinforcement learning for dialogue systems with large action spaces," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 26, no. 11, pp. 2083–2097, Nov. 2018.

[5] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," *ArXiv e-prints*, May 2017.

[6] Stefan Ultes, Lina M. Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gasic, and Steve Young, "PyDial: A Multi-domain Statistical Dialogue System Toolkit," in *Proceedings of ACL 2017, System Demonstrations*, Vancouver, Canada, July 2017, pp. 73–78, Association for Computational Linguistics.

[7] Pierre-Yves Oudeyer, "Computational theories of curiosity-driven learning," *CoRR*, vol. abs/1802.10546, 2018.

[8] Jrgen Schmidhuber, "Adaptive confidence and adaptive curiosity," Tech. Rep., Institut fur Informatik, Technische Universitat Munchen, Arcisstr. 21, 800 Munchen 2, 1991.

[9] Bradly C. Stadie, Sergey Levine, and Pieter Abbeel, "Incentivizing exploration in reinforcement learning with deep predictive models," *CoRR*, vol. abs/1507.00814, 2015.

[10] Jonathan Sorg, Satinder Singh, and Richard L. Lewis, "Variance-based rewards for approximate bayesian reinforcement learning," in *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, Arlington, Virginia, United States, 2010, UAI'10, pp. 564–571, AUAI Press.

[11] Rein Houthooft, Xi Chen, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel, "Vime: Variational information maximizing exploration," in *Advances In Neural Information Processing Systems 29*, D. D. Lee, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., pp. 1109–1117. Curran Associates, Inc., 2016.

[12] Susanne Still and Doina Precup, "An information-theoretic approach to curiosity-driven reinforcement learning," *Theory in Biosciences*, vol. 131, no. 3, pp. 139–148, jul 2012.

[13] Christopher Tegho, Pawel Budzianowski, and Milica Gašić, "Benchmarking uncertainty estimates with deep reinforcement learning for dialogue policy optimisation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6069–6073.

[14] Manuel Lopes, Tobias Lang, Marc Toussaint, and Pierre yves Oudeyer, "Exploration in model-based reinforcement learning by empirically estimating learning progress," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., pp. 206–214. Curran Associates, Inc., 2012.

[15] Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos, "Unifying count-based exploration and intrinsic motivation," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2016, pp. 1471–1479.

[16] Milica Gašić and Steve Young, "Gaussian processes for POMDP-based dialogue manager optimization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 1, pp. 28–40, jan 2014.

[17] Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young, "Agenda-based user simulation for bootstrapping a pomdp dialogue system," in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, Stroudsburg, PA, USA, 2007, NAACL-Short '07, pp. 149–152, Association for Computational Linguistics.