Reward Estimation in Reinforcement Learning

Christian Geishauser

Dialogue Systems and Machine Learning Group

05.06.2020

Content

- Introduction to Reinforcement Learning (RL)
  - What is Reinforcement Learning?
- Inverse Reinforcement Learning (IRL)
  - Guided Dialogue Policy Learning (Takanobu et al. 2019)
- Intrinsic Reward Learning
  - What can learned intrinsic rewards capture? (Zheng et al. 2019)
- Learning in interaction with real users
  - On-line active reward learning (Su et al. 2016)

Introduction to Reinforcement Learning

Introduction to RL

- The introduction is partly inspired by David Silver‘s lectures at UCL
- https://www.davidsilver.uk/teaching/

Introduction to RL

What makes reinforcement learning different from other machine learning paradigms?

- There is no supervisor, only a **reward** signal
- Feedback is delayed, not instantaneous
- Agent‘s actions affect the subsequent data it receives
  - Agent creates its own data

Introduction to RL

- An agent interacts with an environment in discrete time steps

Introduction to RL

- An agent interacts with an environment in discrete time steps
- At each time step t the agent:
  - Observes state s_t
  - Executes action a_t
  - Receives scalar reward r_t

Introduction to RL

- An agent interacts with an environment in discrete time steps

- At each time step t the agent:
  - Observes state s_t
  - Executes action a_t
  - Receives scalar reward r_t
- The environment
  - Receives action a_t
  - Emits state s_{t+1}
  - Emits scalar reward r_t

Rewards

- A **reward** r_t is a scalar feedback signal
- Indicates how well agent is doing at time step t
- The agent tries to maximise cumulative reward $R_t = \sum_{i=t}^{\infty} \gamma^{i-t} r_i$
- $\gamma \in [0, 1]$ trades off immediate and future reward

Reward Hypothesis

- Reinforcement Learning is based on the **reward hypothesis**

> All goals can be described by the maximisation of expected cumulative reward

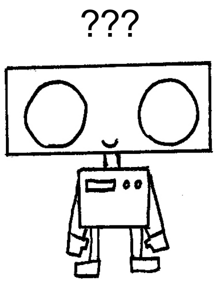
- Reward definition is central for the definition of the goal
- Reward design influences learning significantly

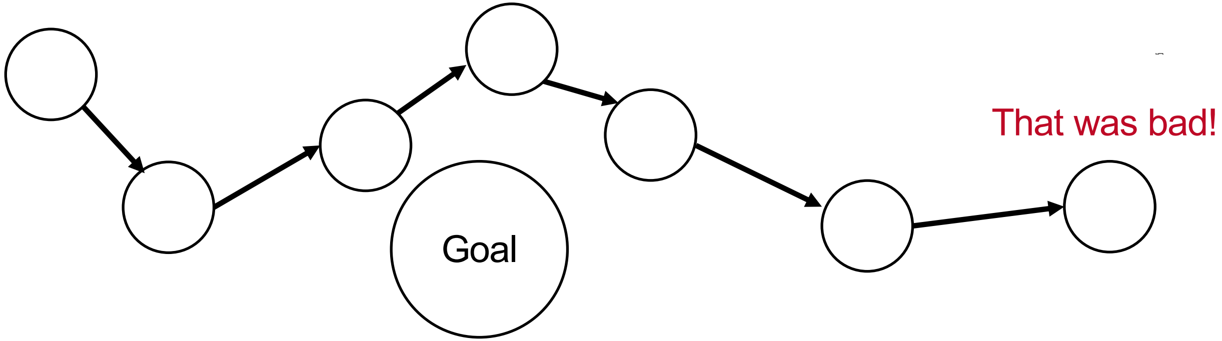
Sequential Decision Making

- Actions may have long term consequences
  - Get a pet
- Reward may be delayed
  - Grade of the exam
- It may be better to sacrifice immediate reward to gain more long-term reward
  - Write thesis instead of going to the rhein

Credit Assignment Problem

- What actions in the trajectory contributed to the outcome?
  - What actions should be reinforced?
  - What actions should be avoided?





Reward for Dialog Policy Optimization

- Small constant negative reward in each turn to keep dialog short
- Huge reward at the end of the dialog, indicating whether goal of the user was completed
  - Was all information provided?
  - Was the correct entity booked?

- Problems:
  - Credit Assignment problem
  - Agent may end the session too quickly, for instance booking a hotel without confirming with the user
  - Agent gets stuck in local minimum by just keeping the dialog as short as possible

Inverse Reinforcement Learning

Inverse Reinforcement Learning

- Instead of learning an optimal behaviour by maximizing hand-crafted reward
- Try to extract a reward function from observed behaviour of an agent
  - Find reward function that expert is implicitly optimizing
  - Behaviour can come from a human for instance
- Once a reward function is found, find an optimal policy to it by using RL

Possible Advantages of IRL

- Transfer learning
  - Even when observed agent is very different to target agent (different action sets for humans and robots) the reward function contains relevant information
  - Transferred reward function can be more robust than transferred policy

- Extends applicability of RL to problems where a reward function is difficult to define manually
  - Model animal behaviour
  - Autonomous driving

- Reward function might be dense, alleviating the credit assignment problem

Issues with IRL

Heinrich Heine
Universität Düsseldorf

- Policy can be optimal for many reward functions (e.g. all zeros)
  - -> Ambiguity in solution
- IRL algorithms assume that observed behaviour is optimal
  - Humans are not perfect
- Difficult to evaluate a learned reward

Maximum Entropy IRL (MEIRL)

Heinrich Heine
Universität Düsseldorf

- IRL is essentially an ill-posed problem as multiple reward functions can explain the expert‘s behaviour
- Ziebart et al., 2008 propose Maximum Entropy IRL to resolve that problem
- Maximum entropy IRL models distribution over trajectories
  - Models demonstrations using a Boltzmann distribution
  - $p_\theta(\tau) = \frac{1}{Z} \exp(-c_\theta(\tau))$, τ being a trajectory, $c_\theta(\tau) = \sum_t c_\theta(s_t, a_t)$ being a cost function
  - Parameters are optimized to maximize the likelihood of demonstrations
  - Probability of trajectory is proportional to the exponential of its cost

Equivalence between GANs and MEIRL

- Finn et al., 2016 draw a strong connection between GANs and MEIRL
  - GANs applied to IRL problems optimize the same objective as MEIRL

Guided Dialog Policy Learning (GDPL)

- Guided Dialog Policy Learning (Takanobu et al., 2019)

- Applies adversarial inverse reinforcement learning for dialogue policy optimization
  - Learns reward estimator and policy simulatenously

Guided Dialog Policy Learning (GDPL)

Heinrich Heine Universität Düsseldorf

- Reward estimator $f_\omega(\tau)$ is optimized using Maximum Entropy IRL
  - maximizes log likelihood of observed human behaviour

Guided Dialog Policy Learning (GDPL)

- Reward estimator $f_\omega(\tau)$ is optimized using Maximum Entropy IRL
  - maximizes log likelihood of observed human behaviour

- $\omega^* = argmax_\omega \mathbb{E}_{\tau \sim D}[f_\omega(\tau)]$ Maximize log likelihood of expert demonstration
- $f_\omega(\tau) = \log p_\omega(\tau) = \log \frac{e^{R_\omega(\tau)}}{Z_\omega}$ Log of Boltzmann distribution
- $R_\omega(\tau) = \sum_t \gamma^t r_\omega(s_t, a_t)$ Energy of sample τ
- $Z_\omega = \sum_\tau e^{R_\omega(\tau)}$ Partition function, used for normalization

Guided Dialog Policy Learning (GDPL)

Heinrich Heine Universität Düsseldorf

- $\omega^* = argmax_\omega \mathbb{E}_{\tau\sim D}[f_\omega(\tau)]$ Maximize log likelihood of expert demonstration
- $f_\omega(\tau) = \log p_\omega(\tau) = \log \frac{e^{R_\omega(\tau)}}{Z_\omega}$ Log of Boltzmann distribution
- $R_\omega(\tau) = \sum_t \gamma^t r_\omega(s_t, a_t)$ Energy of sample τ
- $Z_\omega = \sum_\tau e^{R_\omega(\tau)}$ Partition function, used for normalization

High value $f_\omega(\tau)$ $\longleftrightarrow$ High return $R_\omega(\tau)$

Guided Dialog Policy Learning (GDPL)

- Policy π_θ is encouraged to mimic human dialog behaviour
  - $J_\pi(\theta) = -KL[\pi_\theta(\tau)||p_\omega(\tau)] = \mathbb{E}_{\tau\sim\pi}[f_\omega(\tau) - \log \pi_\theta(\tau)]$
  - Policy should construct trajectories that resemble expert demonstrations

Guided Dialog Policy Learning (GDPL)

- Policy π_θ is encouraged to mimic human dialog behaviour
  - $J_\pi(\theta) = -KL[\pi_\theta(\tau)||p_\omega(\tau)] = \mathbb{E}_{\tau\sim\pi}[f_\omega(\tau) - \log \pi_\theta(\tau)]$
  - Policy should construct trajectories that resemble expert demonstrations

- Reward estimator should distinguish real human sessions from generated ones
  - $J_f(\omega) = -KL[p_D(\tau)||p_\omega(\tau)] + KL[\pi_\theta(\tau)||p_\omega(\tau)]$

Be close to data distribution

Adversarial Learning

Guided Dialog Policy Learning (GDPL)

- Reward estimation uses entire session τ
  - Can lead to reward sparsity
  - May be of high variance due to different trajectory lengths

- -> Estimate state-action pairs instead
  - $J_\pi(\theta) = \mathbb{E}_{s,a\sim\pi}[f_\omega(s,a) - \log \pi_\theta(s,a)]$
  - $J_f(\omega) = \mathbb{E}_{s,a\sim D}[f_\omega(s,a)] - \mathbb{E}_{s,a\sim\pi}[f_\omega(s,a)]$

Guided Dialog Policy Learning (GDPL)

hhu
Heinrich Heine
Universität Düsseldorf

- Big jump in performance compared to baseline methods
- Efficient dialogues, number of turns similar to human demonstrations
- Human-human performance computed on test set

Method	Agenda			
	Turns	Inform	Match	Success
GP-MBCM	2.99	19.04	44.29	28.9
ACER	10.49	77.98	62.83	50.8
PPO	9.83	83.34	69.09	59.1
ALDM	12.47	81.20	62.60	61.2
GDPL-sess	**7.49**	88.39	77.56	76.4
GDPL-discr	7.86	93.21	80.43	80.5
GDPL	7.64	**94.97**	**83.90**	**86.5**
Human	*7.37*	*66.89*	*95.29*	*75.0*

Summary

- Learns reward estimator and optimizes policy simulatenously
  - By using adversarial inverse reinforcement learning
- Reward estimator evaluates state-action pair in every turn
  - Provides dense reward signal -> alleviates credit assignment problem
  - Better „guides“ the dialog policy learning
- Achieves state-of-the-art performance
- Requires pre-training

Intrinsic Reward Learning

Intrinsic reward and reward shaping

Heinrich Heine
Universität Düsseldorf

- Extrinsic reward: Defines the task and captures designer‘s preference of behaviour
  - Reward signal emitted by the environment

- Intrinsic reward: Serves as helpful signal to improve learning dynamics of the agent

- Reward shaping: Modifies the original reward function to make RL methods converge faster
  - For instance „New reward = extrinsic reward + intrinsic reward“

What can learned intrinsic rewards capture?

- What can learned intrinsic rewards capture? (Zheng et al., 2019)
- There is a difference between knowledge in rewards and policies
- Meta-learns an intrinsic reward function to help policies during learning

What can learned intrinsic rewards capture?

- Policies, value-functions, state representations, models of the environment
  - Are loci of knowledge as being structures where knowledge can be deposited and reused

What can learned intrinsic rewards capture?

- Policies, value-functions, state representations, models of the environment
  - Are loci of knowledge as being structures where knowledge can be deposited and reused

- Claim: Reward function is also a good locus of knowledge
  - Reward is usually treated as given and immutable

What can learned intrinsic rewards capture?

- Knowledge in rewards: „What“ the agent should strive to do
  - More indirect, thus slower to make an impact on behaviour

What can learned intrinsic rewards capture?

- Knowledge in rewards: „What“ the agent should strive to do
  - More indirect, thus slower to make an impact on behaviour
- Knowledge in policy: „How“ an agent should behave
  - Can directly have an impact on behaviour

What can learned intrinsic rewards capture?

- Knowledge in rewards: „What“ the agent should strive to do
  - More indirect, thus slower to make an impact on behaviour
- Knowledge in policy: „How“ an agent should behave
  - Can directly have an impact on behaviour
- Measure of usefulness of the intrinsic reward: Lifetime return
  - Lifetime return: Cumulative extrinsic reward obtained by the agent over ist entire lifetime

What can learned intrinsic rewards capture?

- Knowledge in rewards: „What“ the agent should strive to do
  - More indirect, thus slower to make an impact on behaviour
- Knowledge in policy: „How“ an agent should behave
  - Can directly have an impact on behaviour
- Measure of usefulness of the intrinsic reward: Lifetime return
  - Lifetime return: Cumulative extrinsic reward obtained by the agent over ist entire lifetime
  - Lifetime return: $G^{life} = \sum_{t=0}^{T-1} \gamma^t \, r_{t+1}$
  - T is the number of steps in the lifetime
  - r_t denotes extrinsic reward

Optimal Reward Problem (Singh et al. 2010)

- Intrinsic reward: A reward function $r_\eta(\tau_{t+1})$ parameterised by η
  - $\tau_t = (s_o, a_0, r_1, d_1, s_1, \ldots, r_t, d_t, s_t)$ is a lifetime history

- The reward function is **non-stationary**
  - Reward function can adapt to learning progress of agent
  - Useful as agent goes through different learning phases

- Goal: Learn parameters η that optimises lifetime return
  - Using lifetime return instead of episodic return allows exploration across multiple episodes

Learning intrinsic reward

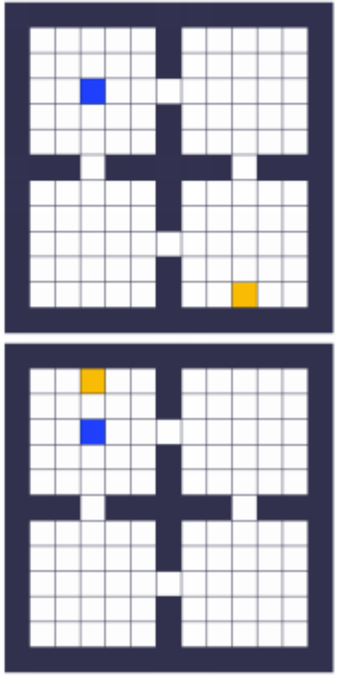
- Intrinsic reward function $r_\eta(s)$ modelled by an RNN which obtains whole history as input
  - History as input crucial: Balance exploration and exploitation
  - For instance by capturing how frequently a state is visited -> exploration bonus

- $r_\eta(s)$ is meta-learned with objective function
  - $J(\eta) = \mathbb{E}_{\theta_0 \sim \Theta, \mathcal{T} \sim p(\mathcal{T})}[\mathbb{E}_{\tau \sim p_\eta(\tau|\theta_0)}[G^{life}]]$

- Policy parameters updated using solely intrinsic rewards
  - By policy gradient

Experiment: Explore uncertain states

- Empty rooms environment
  - Agent starts in the centre of top-left room
  - Only one cell is rewarding, the „goal cell“
  - Goal cell is invisible to the agent
  - Goal sampled uniformly at the beginning of the lifetime
  - Episode terminates when goal has been reached



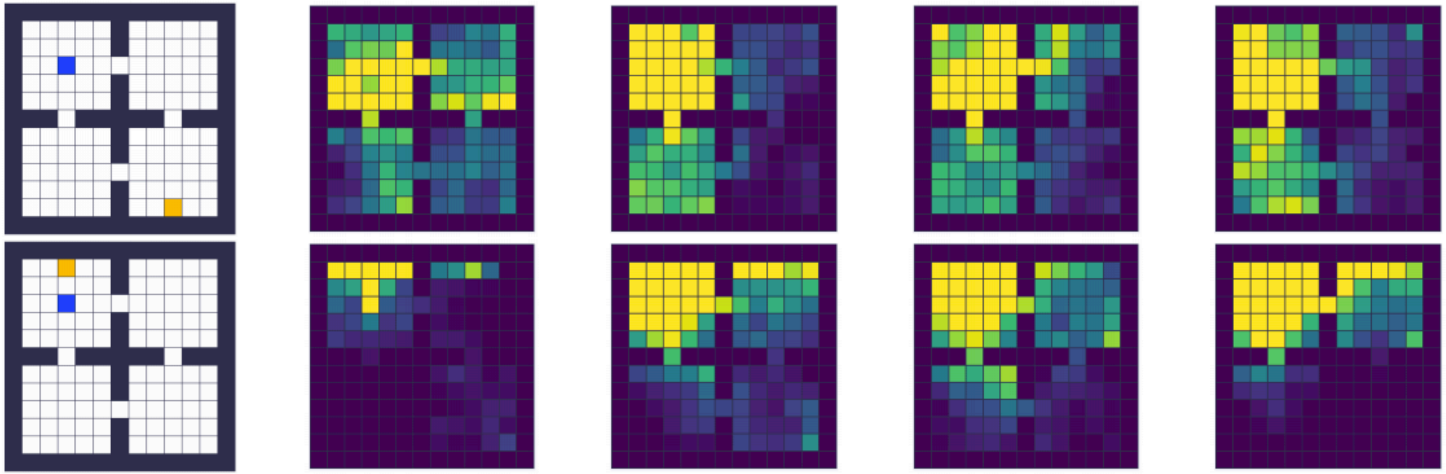
  - Blue and yellow squares represent agent and goal, respectively

Explore uncertain states

- Empty rooms environment
  - Agent needs to explore all cells until goal is found, then exploit knowledge

Explore uncertain states

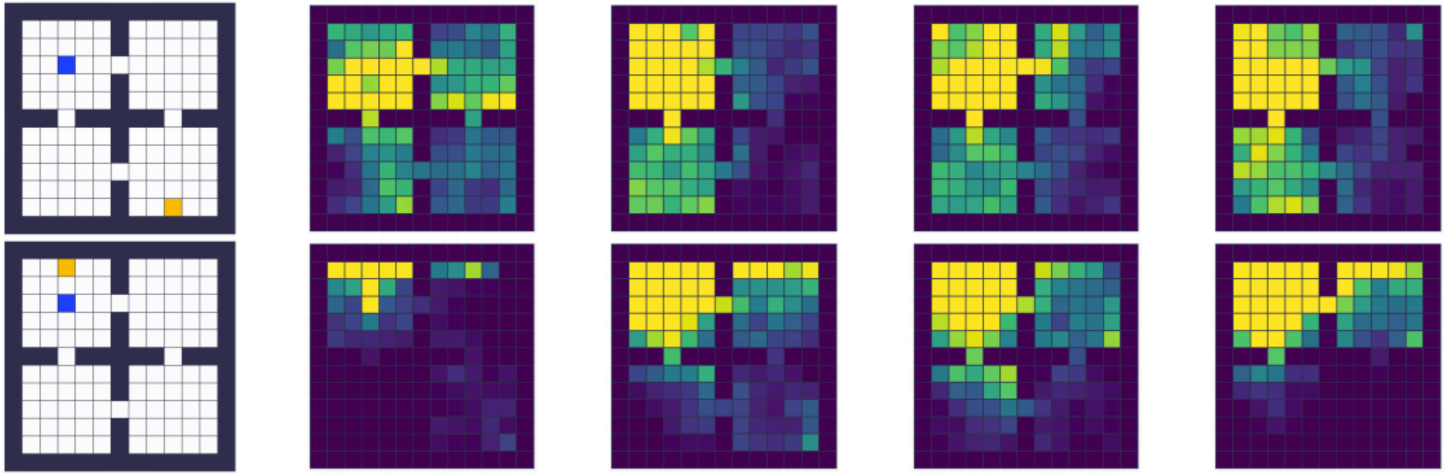
- Empty rooms environment
  - Agent needs to explore all cells until goal is found, then exploit knowledge



(a) Room instance (b) Intrinsic (ours) (c) Extrinsic (d) Count-based (e) ICM

Top: Agent is encouraged to explore Bottom: Agent should exploit knowledge of goal location

Explore uncertain states



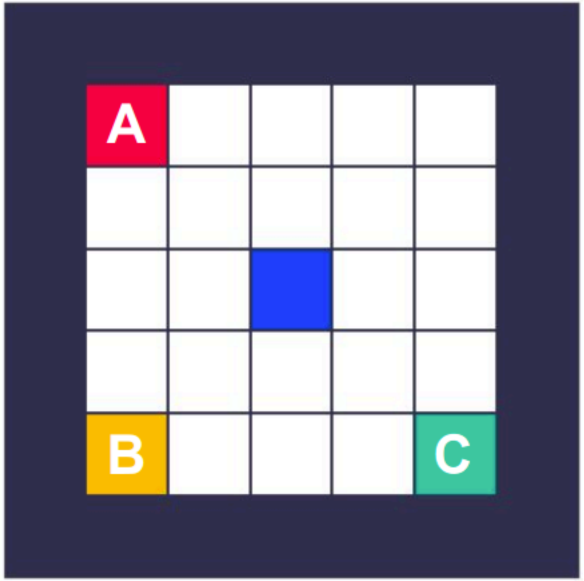
(a) Room instance (b) Intrinsic (ours) (c) Extrinsic (d) Count-based (e) ICM

- Exploration-focused models (d) and (e) do not adjust after the goal has been found
  - They are stationary and do not incorporate the lifetime history

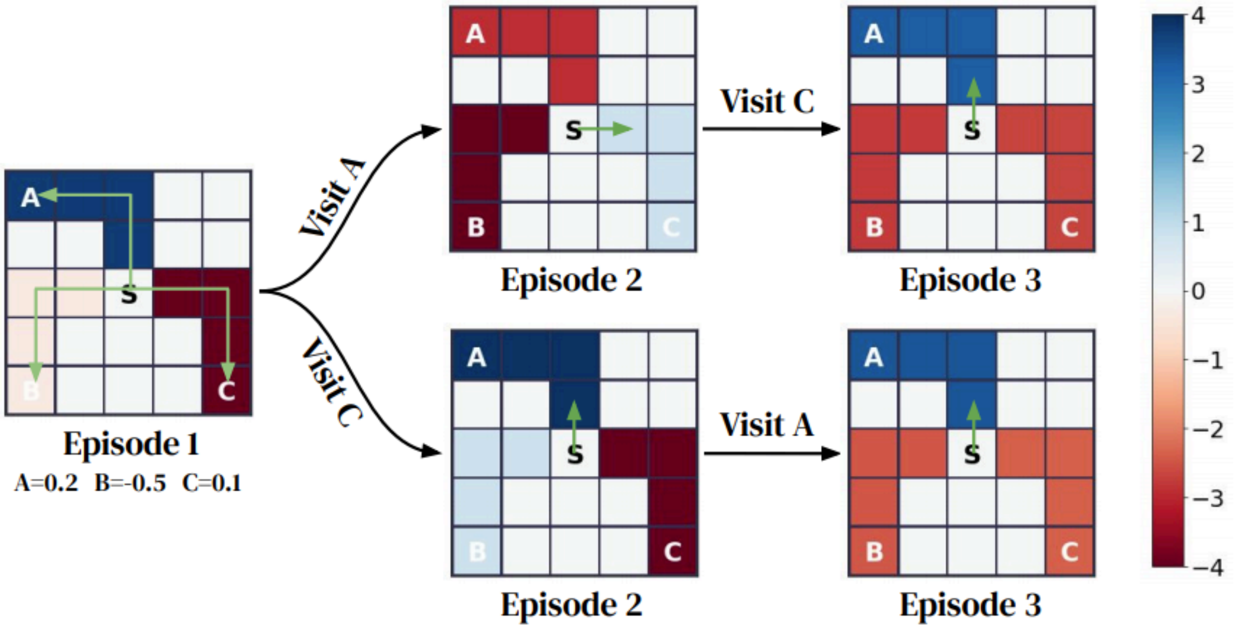
Explore uncertain, avoid harmful objects

- Random ABC environment
  - Rewards for objects A, B and C uniformly sampled
  - From [-1, 1], [-0.5, 0] and [0, 0.5] respectively
  - Then held fixed within the lifetime

- Should learn that
  - B should be avoided
  - A and C have uncertain rewards -> visit them
  - Once determined whether A or C is better -> exploit



Explore uncertain, avoid harmful objects



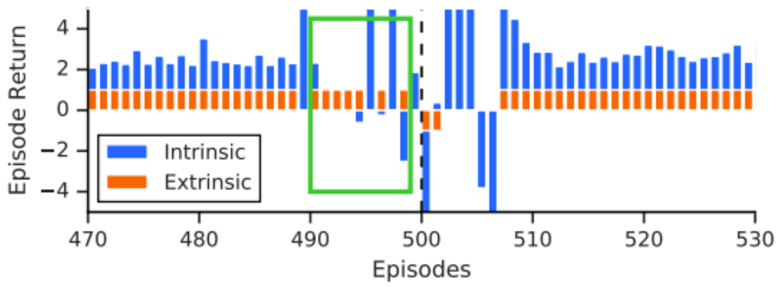
- Random ABC environment

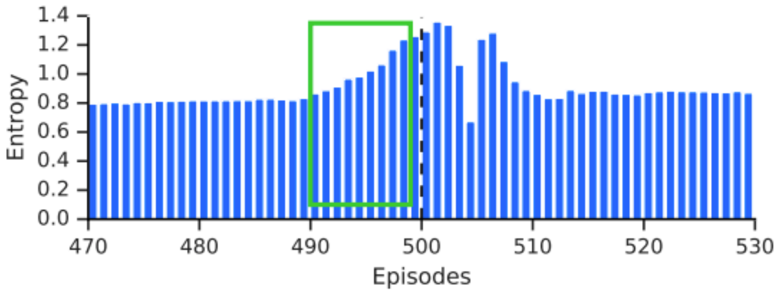
Dealing with non-stationarity

- Non-stationary ABC environment
  - Reward for A either 1 or -1
  - Reward for B is -0.5
  - Reward for C is the negative value of the reward for A
  - Reward of A and C swapped after 250 episodes
  - Lifetime lasts 1000 episodes

Dealing with non-stationarity

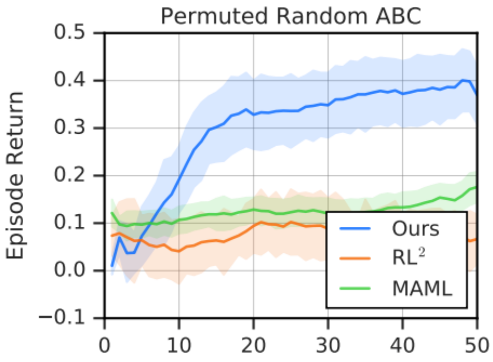
Heinrich Heine
Universität Düsseldorf





- Task changes at 500th episode
- Intrinsic reward gives a negative reward even before the task changes
- Makes policy less deterministic (entropy increases)
- Higher entropy -> agent can quickly adapt to changes

Generalisation via Rewards



- Generalisation to unseen action spaces:
  - Permute actions left/right and up/down
  - Intrinsic reward still useful because it only assigns reward to agent‘s state changes
  - Reward captures „what to do“, making it possible to generalize to new actions
- Meta-learning algorithms were not able to generalise to the permuted environment
  - Transferred policies are highly biased towards the original action space
  - Highlights the difference between „what to do“ and „how to do“ knowledge captured by policies

Summary

Heinrich Heine
Universität Düsseldorf

- Proposes a method for learning intrinsic rewards to tackle optimal reward problem
- Learned reward function is non-stationary
  - Encourages explorative and exploitative behaviour across multiple episodes
- Experiments highlight difference between „what do do“ and “how to do“ knowledge
- Computationally very expansive since you need to run a lot of lifetimes

Learning in interaction with real users

Gap between simulation and real-world

- Reinforcement Learning algorithms are usually trained in simulation
- Gap between simulation and real-world determines how good algorithm perform in the real world
- Necessary to adapt policy to real-world environment
- In dialogs: Learn on-line in interaction with real users

Learning in interaction with real users

Heinrich Heine Universität Düsseldorf

- On-line active reward learning for policy optimisation (Su et al., 2016)
- Learns policy from scratch in interaction with real users
- Using Gaussian processes

On-line active reward learning

- Task success can be determined from
  - Subjective user ratings (Subj)
  - Objective measure (Obj)

- Obj: Often impractical as user‘s goal normally not available
  - Inflexible and often fail if user does not strictly follow the task
  - Results in mismatch between Obj and Subj rating

- Subj: Frequently inaccurate responses
  - Results in unstable learning

On-line active reward learning

- Use Gaussian process prediction model for inferring task success

On-line active reward learning

- Use Gaussian process prediction model for inferring task success

- Goal: Compute probability $p(y|d, \mathcal{D})$ that task was successful
  - Given current dialog representation $\boldsymbol{d}$ and previously classified dialogues $\boldsymbol{\mathcal{D}}$

On-line active reward learning

Heinrich Heine
Universität Düsseldorf

- Use Gaussian process prediction model for inferring task success
- Goal: Compute probability $p(y|d, \mathcal{D})$ that task was successful
  - Given current dialog representation $\boldsymbol{d}$ and previously classified dialogues $\boldsymbol{\mathcal{D}}$
- Model $p(y|d, \mathcal{D}) = \phi(f(d|D))$, where
  - $f\colon \mathbb{R}^{n_d} \rightarrow \mathbb{R}$ is a latent function, modelled by a Gaussian process
  - ϕ denotes the cumulative density function of the standard Gaussian (sigmoid also possible)
  - Use expectation propagation algorithm to make posterior tractable

On-line active reward learning

- Why Gaussian process?

- Neural networks require large amounts of training data
  - Not suitable for training from scratch with real users

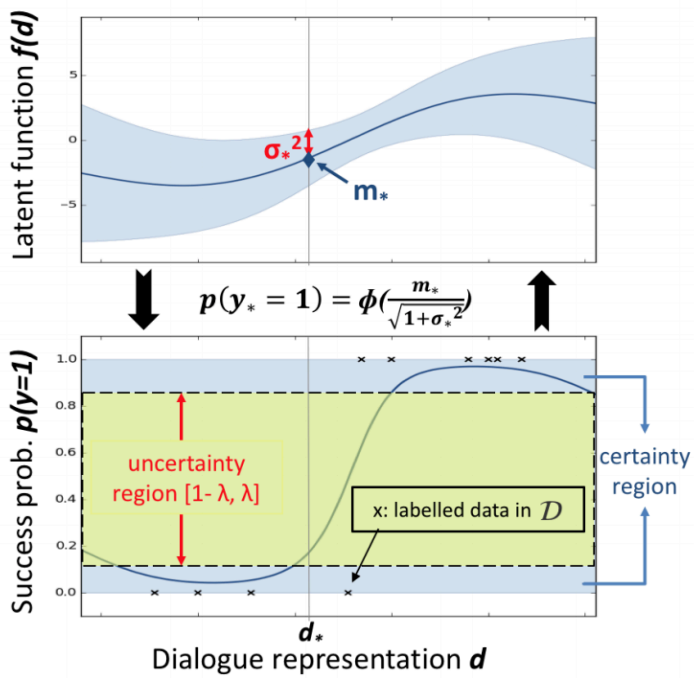
- Gaussian processes learn really quick
  - By incorporating prior knowlege in form of the kernel function

- Gaussian processes come naturally equipped with a measure of uncertainty

On-line active reward learning

- Use threshold intervall $[1-\lambda, \lambda], \lambda \in (0.5, 1]$
  - To decide whether the dialogue should be labelled

- m^*, σ_*^2 denote the posterior mean and variance of $f(d_*)$, respecitvely

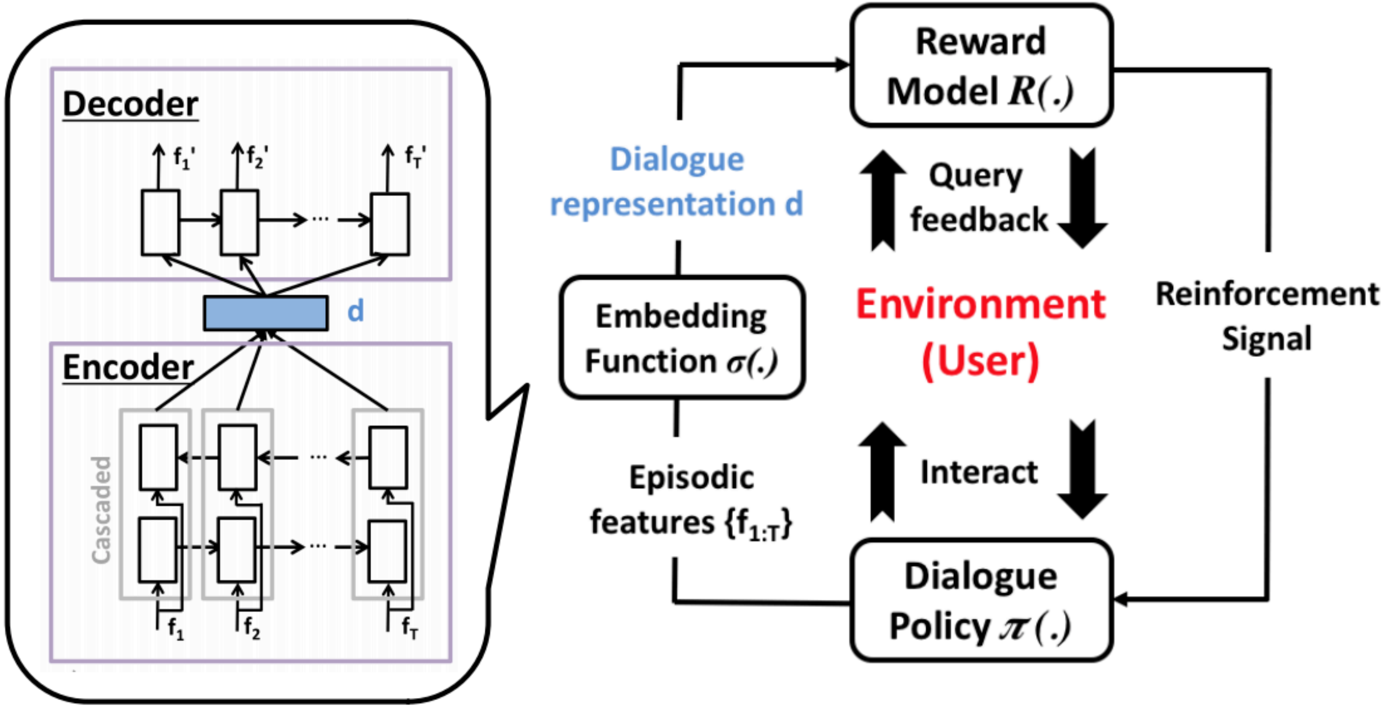


On-line active reward learning

- Dialogue representation *d* computed using a bidirectional LSTM autoencoder
  - Trained with a dialogue corpus comprising user dialogues in the cambridge restaurant domain

- Policy is modelled by a Q-network
  - Q-values estimated by a Gaussian process (GP-SARSA)

System framework



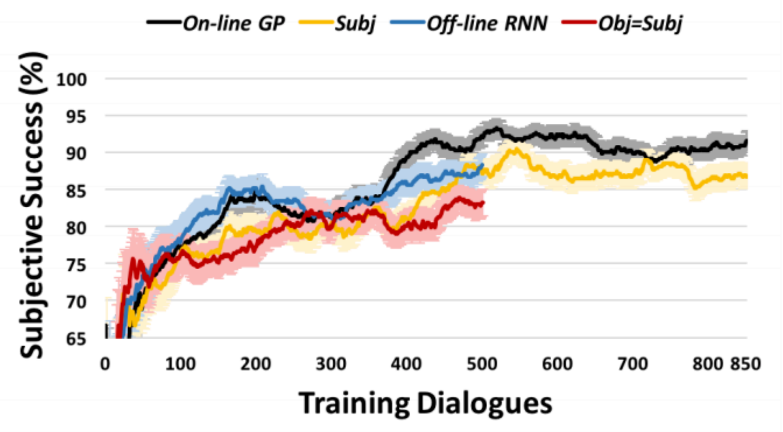
Experimental Results

- Compare on-line GP to
  - Obj=Subj: dialogue is only used if subjective and objective success rating coincide
  - Sub: Use subjective rating of the user
  - Off-line RNN: Train an RNN on 1K simulated dialogues off-line as success estimator

Experimental Results

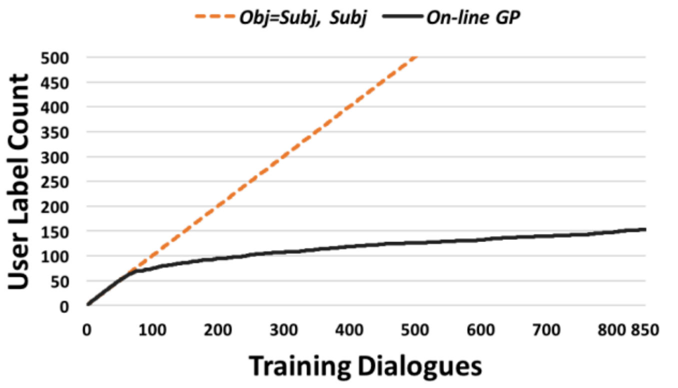
- Compare on-line GP to
  - Obj=Subj: dialogue is only used if subjective and objective success rating coincide
  - Sub: Use subjective rating of the user
  - Off-line RNN: Train an RNN on 1K simulated dialogues off-line as success estimator

- Success is calculated using the moving average



Experimental Results

- Compare on-line GP to
  - Obj=Subj: dialogue is only used if subjective and objective success rating coincide
  - Sub: Use subjective rating of the user
  - Off-line RNN: Train an RNN on 1K simulated dialogues off-line as success estimator
- Only requires ~150 user queries



Summary

Heinrich Heine
Universität Düsseldorf

- Goal: Learn policy from scratch in interaction with real users

- Approach:
  - Use Gaussian process prediction model to infer task success
  - Only query user feedback if uncertainty is within a given threshold
  - Learn dialgoue representation using an RNN autoencoder

- Results:
  - GP reward leads to best performance
  - Only requires a fraction of queries compared to all training dialogues

Thank you

References

- Ziebart, Brian D., et al. "Maximum entropy inverse reinforcement learning." *Aaai*. Vol. 8. 2008.
- Finn, Chelsea, et al. "A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models." *arXiv preprint arXiv:1611.03852* (2016).
- Takanobu, Ryuichi, Hanlin Zhu, and Minlie Huang. "Guided Dialog Policy Learning: Reward Estimation for Multi-Domain Task-Oriented Dialog." *arXiv preprint arXiv:1908.10719* (2019).
- Zheng, Zeyu, et al. "What Can Learned Intrinsic Rewards Capture?." *arXiv preprint arXiv:1912.05500* (2019).
- Su, Pei-Hao, et al. "On-line active reward learning for policy optimisation in spoken dialogue systems." *arXiv preprint arXiv:1605.07669* (2016).
- Singh, Satinder, et al. "Intrinsically motivated reinforcement learning: An evolutionary perspective." *IEEE Transactions on Autonomous Mental Development* 2.2 (2010): 70-82.