Discovering Subgoals from Trajectories Using Empowerment

Luiz A. Thomasini[1], **Arturo de Souza**[1], **Gabriel de Oliveira Ramos**[1]

[1]Graduate Program in Applied Computing
Universidade do Vale do Rio dos Sinos, São Leopoldo – RS – Brasil

{luizalfredo, arturossouza}@edu.unisinos.br, {gdoramos}@unisinos.br

Abstract. *Reinforcement learning algorithms struggle in complex environments with sparse rewards due to inefficient exploration. Identifying subgoals to guide exploration and structure learning is a promise and a challenge. This work presents a novel method to discover subgoals from trajectories by leveraging empowerment, an information-theoretic measure of an agent's potential to influence its environment. The main hypothesis is that states with high empowerment correspond to strategic locations, such as bottlenecks, which serve as subgoals. We evaluate our method in grid-world navigation tasks, demonstrating that it successfully identifies important states without requiring reward signals or successful trajectories, and improves agent performance.*

1. Introduction

Reinforcement Learning (RL) [Sutton 2018] provides a powerful framework for developing intelligent agents that learn through trial and error and apply decision making over time to achieve a goal. However, in environments with sparse or delayed rewards, random exploration is often inefficient, leading to poor sample efficiency. A promising direction is to identify intermediate subgoals that can break down a task. If an agent can identify strategic locations within its environment, it can use those to learn specialized skills that guide its exploration to learn more effectively [Sutton et al. 1999]. Those are key to developing hierarchical RL architectures.

This paper proposes a method that uses empowerment as a utility function to identify states of intrinsic interest from an agent's experience. Our main hypothesis is that states with high empowerment, locations from which an agent can reach a large set of future states, are strategic and make good subgoals. For example, a doorway in a building has high empowerment because passing through it unlocks access to an entirely new set of rooms. Unlike prior methods that require successful trajectories or graph-based analysis [McGovern and Barto 2001, Şimşek and Barto 2004, Şimşek et al. 2005], our empowerment-based approach has advantages in both that it discovers subgoals purely from the environmental structure without task-specific success patterns, and that it allows for scaling to continuous and stochastic environments.

Our aim is to develop an approach that is task independent, does not require successful trajectories, and discovers subgoals based purely on the learned structure of the environment. We present a simple, three-step process to achieve our objective:

1. **Build a Model**: Use RL trajectories to construct a transition model of the world.
2. **Compute Empowerment**: Use the model to compute empowerment of states.

3. **Select Subgoals**: Rank states by their scores and select the top-k as subgoals.

The main contribution of this work is integrating empowerment computation to the subgoal discovery problem, which, to the best of our knowledge, has not yet been reported by the scientific community. Our experiments in classic grid-world domains show that our method effectively identifies intuitive subgoals, such as doorways, validating empowerment as a potential metric for subgoal discovery.

2. Background

We formalize the reinforcement learning setting as a *Markov Decision Process* (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ is the set of possible states, $\mathcal{A}$ is the set of possible actions, $P : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ is the transition function, where $P(s'|s, a)$ gives the probability of transitioning to state s' when taking action a in state s, $R : \mathcal{S} \times \mathcal{A} \rightarrow R$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. A trajectory T in the MDP is composed by tuples of steps $< s_t, a_t, s_{t+1} >$, and formalized as $T = \{s_0, a_0, s_1, a_1, s_2, ...s_t, a_t, s_{t+1}\}$. Given an arbitrary number of experiences n, define $\mathcal{T}$ as the set of trajectories produced by the agent.

Empowerment [Klyubin et al. 2005] measures the perceived amount of influence or control that the agent has over the environment. Analogous to a communication channel from information theory [Cover 1999], where discrete signals are described by a conditional probability distribution $p(y|x)$, we define the h-step empowerment by $\mathcal{E}_h = \max_{p(a^h)} I(A^h; S_n) = \max_{p(a^h)} \sum_{a^h, s_n} p(s_n|a^h)p(a^h) \log_2 \frac{p(s_n|a^h)}{\sum_{a^n} p(s_n|a^h)p(a^h)}$, the channel capacity of the agent's actuation channel, measured in bits.

3. Method

This work proposes *Empowerment Subgoal Discovery*, a solution to the subgoal discovery problem in reinforcement learning by leveraging empowerment. We present a three-step progression that: (1) build a transition model of the environment; (2) compute empowerment of every state; (3) select the top-k empowerment states as subgoals.

3.1. Building the Transition Model

Given a set of n trajectories $\mathcal{T}$ containing tuples $< s_t, a_t, s_{t+1} >$ from agent-environment steps, we build a transition model $M : S \times A \times S \rightarrow \psi$. The entire transition probability distribution of a state-action pair is defined by Ψ, therefore ρ describes the probability of ending in state s_{t+1} when action a is selected in state s. The learned model works both for deterministic and stochastic environments, but currently is limited to discrete state spaces.

3.2. Computing Empowerment

We compute empowerment of each state in set $\mathcal{S}$ from the model. The time horizon h is the length of a sequence of actions. The set of action sequences $\mathcal{A}^h$ is generated as the product of every action in $\mathcal{A}$ of length h. In addition, we are executing each action sequence in $\mathcal{A}^h$ against the model and storing the state reached to form the set of distinct states $\mathcal{S}'(s)$. We denote $T^h(s, a^h)$ the state reached by executing the action sequence $a^h = (a_1, \ldots, a_n)$ starting from state s. This allows us to compute the h-step empowerment.

Computing empowerment method is shown in algorithm 1. Here we used a simplified application of the metric $\mathcal{E}_n(s) = \log_2 |\{T^n(s, a^n) : a^n \in \mathcal{A}^n\}|$ where empowerment

Algorithm 1 Compute Empowerment

Input: Model M, time horizon h
$\mathcal{S} \leftarrow \{s$: Unique states in $M\}$
$\mathcal{A} \leftarrow \{a$: Unique actions in $M\}$
Output: Empowerment scores $\mathcal{E}$
Initialize $\mathcal{A}^h$ as the product of every action sequence of length h from $\mathcal{A}$
Initialize $\mathcal{E}$ as zero for each state $s \in S$
for each state $s \in \mathcal{S}$ **do**
    $\mathcal{S}'(s) \leftarrow \emptyset$ ▷ Reachable states set
    for each action sequence $a^h \in \mathcal{A}^h$ **do**
        $\Psi_s \leftarrow M[(s, a^h)]$ ▷ Query model using action sequence
        $s' \leftarrow \text{sample}(P(S \mid \Psi_s))$
        $\mathcal{S}'(s) \leftarrow \mathcal{S}'(s) \cup \{s'\}$
    end for
    $\mathcal{E}(s) \leftarrow \log_2 |\mathcal{S}'(s)|$ ▷ Empowerment score
end for
return $\mathcal{E}$

is obtained by taking the binary logarithm from the size of $\mathcal{S}'(s)$ [Klyubin et al. 2005]. This simple approach is feasible for dealing with discrete grid environments.

3.3. Selecting top-*k* Subgoals

Finally, we rank all states according to their h-step empowerment scores and select the top-k states as subgoal candidates $\mathcal{G}_k = argtop(\mathcal{E}, k)$. In the experiment section, we randomly select a single subgoal from $\mathcal{G}_k$, although the framework naturally extends to working with multiple subgoals.

4. Experiments and Results

Our experiments validate the hypothesis that high-empowerment states serve as effective subgoals to improve agent performance. We consider the TwoRooms and FourRooms environments of discrete action as state spaces. The experiments consider the sparse reward setting for the main task, where every transition gives a reward of 0, except for transitioning to the terminal state, which gives a reward of +1.

We present the numerical results from the perspective of the accumulated reward over time. For these demonstrations, the Q-Learning algorithm was selected as the baseline and compared with the augmented action set using a *Reward-Respecting* [Sutton et al. 2023] skill formulation that reaches the subgoal discovered by our method. Figure 1 shows that our method outperforms the baseline in terms of accumulated rewards.

5. Conclusion and Future work

We presented a method for discovering subgoals from trajectories using the principle of empowerment. Our approach successfully identifies bottleneck states in these grid environments and improves agent performance. By framing subgoal discovery as a search for states that maximize an agent's control over its future, we provide a robust and interpretable foundation for hierarchical learning and exploration.

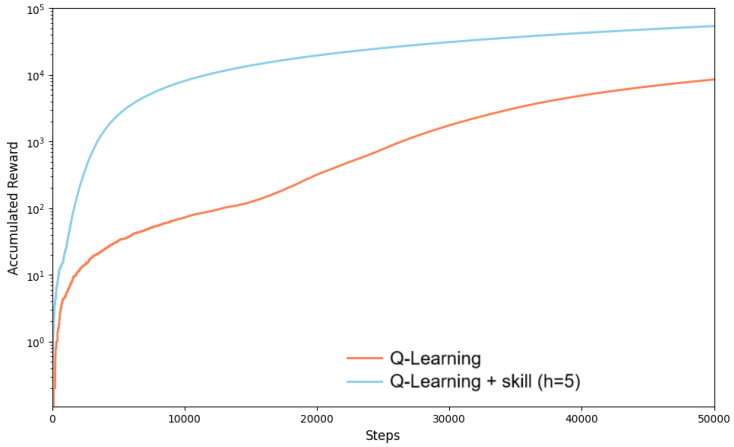

Figure 1. Accumulated reward over time for standard Q-learning (orange) and augmented Q-Learning with skill that reaches the subgoal (ours, blue).

The main challenge for future work is scalability. The exponential complexity of exact empowerment computation is a significant limitation. Promising directions include developing sampling-based or function approximation methods to estimate empowerment in large or continuous state-action spaces. Further research could also explore how to dynamically adjust the time horizon h or integrate this subgoal discovery mechanism into end-to-end deep reinforcement learning architectures.

Acknowledgments

This research was partially supported by CNPq (grants 443184/2023-2, 313845/2023-9, 445238/2024-0) and CAPES (Finance Code 001).

References

Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2005). All else being equal be empowered. In *European Conference on Artificial Life*, pages 744–753. Springer.

McGovern, A. and Barto, A. G. (2001). Automatic discovery of subgoals in reinforcement learning using diverse density. In *ICML*, volume 1, pages 361–368.

Şimşek, Ö. and Barto, A. G. (2004). Using relative novelty to identify useful temporal abstractions in reinforcement learning. In *Proc. of ICML*, page 95.

Şimşek, Ö., Wolfe, A. P., and Barto, A. G. (2005). Identifying useful subgoals in reinforcement learning by local graph partitioning. In *Proceedings of the 22nd international conference on Machine learning*, pages 816–823.

Sutton, R., Precup, D., and Singh, S. (1999). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211.

Sutton, R. S. (2018). Reinforcement learning: An introduction. *A Bradford Book*.

Sutton, R. S., Machado, M. C., Holland, G. Z., Szepesvari, D., Timbers, F., Tanner, B., and White, A. (2023). Reward-respecting subtasks for model-based reinforcement learning. *Artificial Intelligence*, 324:104001.