Reward Inference for Reinforcement Learning: A Semi-Supervised Approach

Alex Albors Juez

Supervised by Matthew Thorpe

WARWICK
THE UNIVERSITY OF WARWICK

Reinforcement Learning Setting

In Reinforcement Learning, an intelligent agent interacts with a dynamic environment seeking to maximize is cumulative rewards.

At at a time t, the agent observes an action given by a state variable s_t, takes an action a_t and receives a reward r_t. Obtaining rewards for every action taken in the environment often requires effort. Rewards are usually scarce and the agent must learn how to act from sparse reward signals.

Goal: Optimally select which actions to query a reward for (Active Learning), as well as how to leverage the unrewarded actions to infer rewards without having to query the model (Semi-Supervised Learning).

Introduction to (Laplacian) Semi-Supervised Learning

Graph-based Methods: SSL methods use labeled and unlabeled data in learning tasks, taking advantage of the topological and geometric structure of the unlabeled data to improve algorithm performance. Graph-based SSL methods employ a weighted graph to express relationships within the data, with weights representing a notion of similarity between the vertices. A common choice is $w_{ij} = \frac{1}{\epsilon^d} \exp\left(-\frac{\|x_i - x_j\|_2^2}{\epsilon}\right)$.

Some notation: We denote $\mathcal{X} = \{x_1, \ldots, x_m\} \subset \mathbb{R}^d$ to be the given data, with labels $y_1, \ldots, y_m \in \{\mathbf{e}_1, \ldots \mathbf{e}_k\}$ where $n << m$. Denote $\Omega_n = \{x_1, \ldots, x_n\}$ to be the labeled set. Let $g(x_i) = y_i$ be the function mapping Ω_n to its corresponding labels.

Laplace Learning: Laplace learning [1] seeks a solution $u : \mathcal{X} \to \mathbb{R}^k$ that minimizes a graph-Dirichlet energy $\mathcal{E}(u)$ subject to the conditions $u(x_i) = y_i$ for $1 \leq i \leq m$. The minimizer satisfies the (discrete) Laplace equation $\mathcal{L}u(x_i) = 0$ for $i > n$ where $\mathcal{L}$ is the graph Laplacian operator.

$$\mathcal{E}(u) = \frac{1}{2}\sum_{i,j=1}^{m} w_{ij}\left(u(x_i) - u(x_j)\right)^2, \quad \mathcal{L}u(x_i) = \sum_{j=1}^{m} w_{ij}\left(u(x_i) - u(x_j)\right).$$

Can be solved efficiently by preconditioned conjugate gradient methods.
Probabilistic interpretation of Laplace learning: Let $X_0, X_1, \ldots$ be a random walk on the graph nodes with transition probabilities given by

$$\mathbb{P}(X_{k+1} = x_j | X_k = x_i) = \frac{w_{ij}}{\sum_l w_{il}}. \tag{1}$$

If $\tau = \inf\{t : X_t \in \Omega_n\}$, the solution satisfies $u(x_i) = \mathbb{E}\left[g(X_\tau)|X_0 = x_i\right]$. Looking at u component-wise, $u(x_i) = (p_{i1}, p_{i2}, \ldots, p_{ik})$ where $p_{ij} = \mathbb{P}(g(X_\tau) = e_j | X_0 = i)$. This shows $u(x_i)$ is a probability distribution for each $x_i \in \mathcal{X}$!

Framework for Active Reinforcement Learning

Setting: The dataset $\mathcal{X}$ is state-action pairs $(s, a) \in \mathcal{S} \times \mathcal{A}$. With a subset being labeled with the corresponding rewards r. Active learning seeks to learn the labels $\mathcal{X}$ by making as few queries as possible. In Laplace learning, can quantify certainty of $u(x_i)$ prediction with acquisition function $\mathcal{A}(u(x)) = \|u(x)\|_2$, introduced in [2].

1. **While** True **do**:
2. $u \leftarrow \arg\min_u \mathcal{E}(u)$. - Laplace learning
3. $x \leftarrow \arg\min_x \mathcal{A}(u(x))$. - Select most uncertain prediction
4. $\Omega \leftarrow \Omega + \{x, r\}$. - Update labeled set Ω.

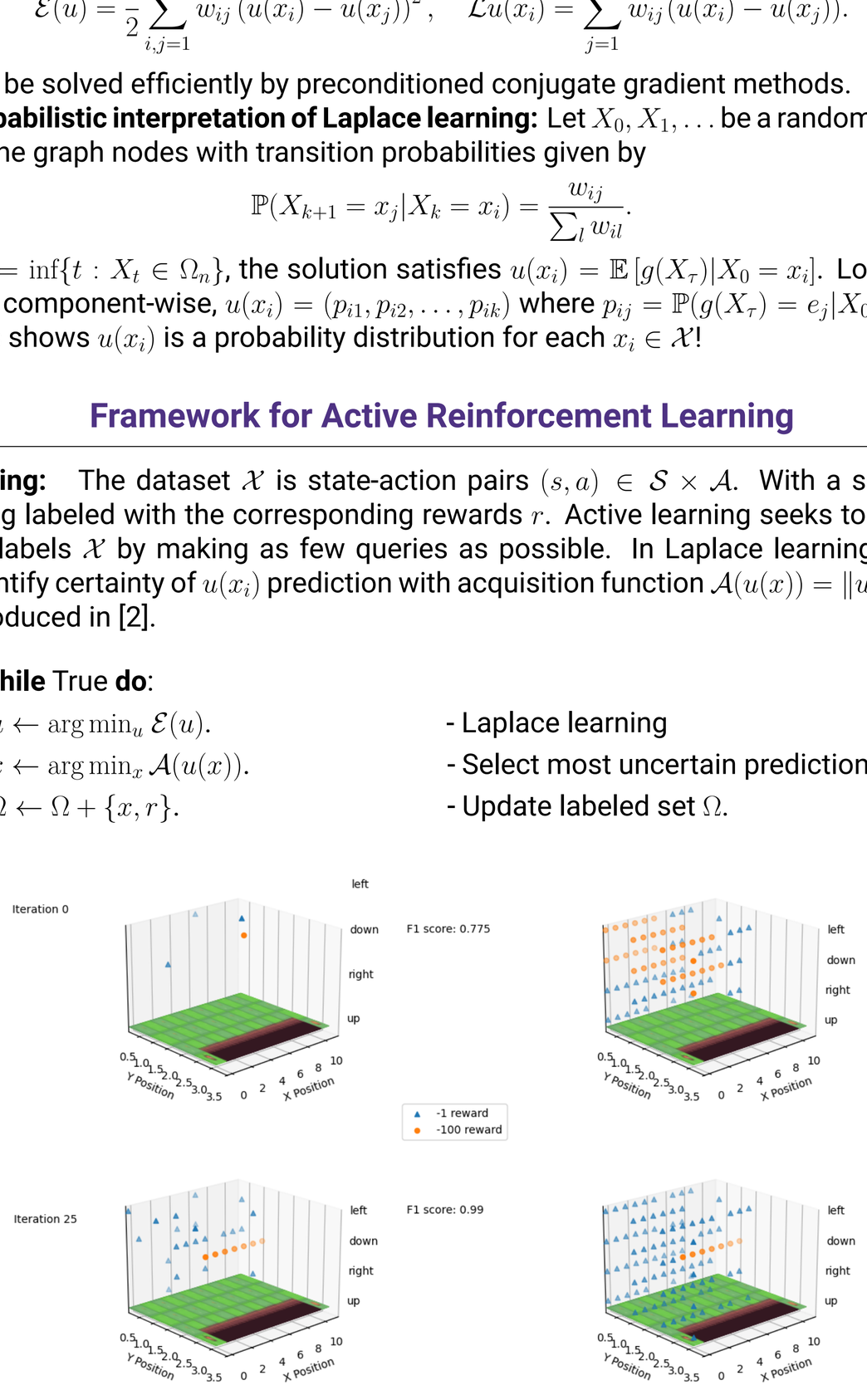
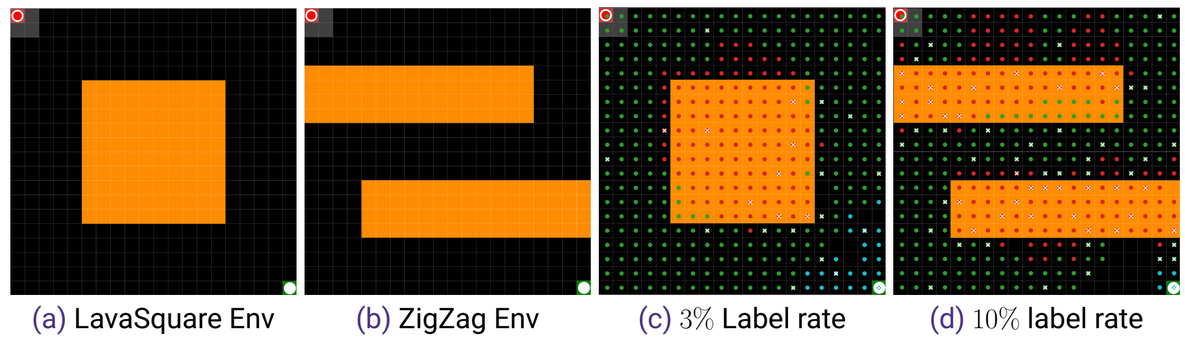


Figure 1. Reward landscape after 25 active learning iterations

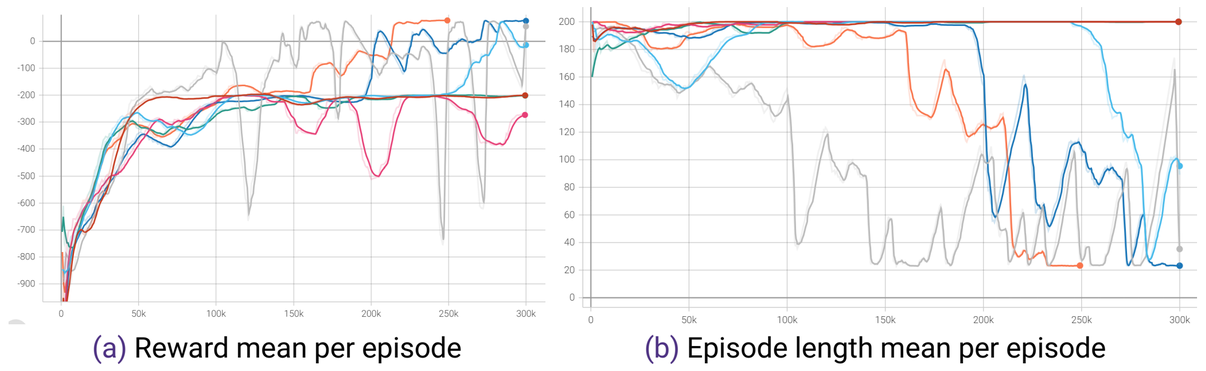
Inference on ActiveGrid Reward Landscape

For simplicity, we illustrate the results for the classic DQN algorithm [3], with minor improvements such as Double Q-networks, but the method may work for any off-policy algorithm. We design a grid environment suitable for SSL that allows the agent to step over obstacles to better explore the environment. The agent must travel from the red square to the green square, avoiding the lava obstacles. Stepping over the incurs a penalty of $r = -20$. To ensure the agent seeks to reach the goal, stepping on black blocks incur a reward of -1, and the goal of $r = 100$.



(a) LavaSquare Env (b) ZigZag Env (c) 3% Label rate (d) 10% label rate

Figure 2. Environments (a, b) and active learning estimations on them (c, d), respectively



(a) Reward mean per episode (b) Episode length mean per episode

Figure 3. Sample learning runs on LavaSquare environment with grid size 20

The light gray curve represents Active Reinforcement Learning (ARL-DQN), where the agent is allowed to query the oracle with a 0.05% probability. For comparison, all others are standard agents. The orange agent see rewards with a given probability p, detailed below.

Plot Color	Method	p	Steps needed
Orange	DQN	1	150k
Blue	DQN	0.1	120k
Light Gray	ARL-DQN	0.10	150k
Pink	DQN	0.005	N/A
Light Blue	DQN	0.01	200k

Table 1. Legend for Figure 3

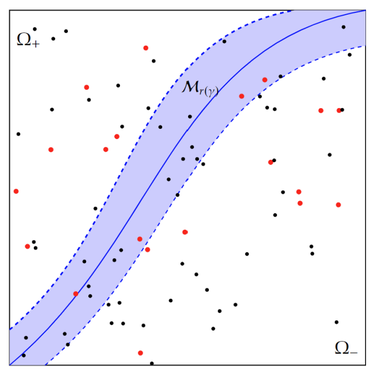


Figure 4. Theoretical Setting

Theoretical Guarantees

Laplace learning can also be formulated in the scalar-valued setting $u : \mathcal{X} \to \mathbb{R}$. Under binary label assumptions $y \in \{-1, 1\}$, where points in $\mathcal{X}$ are labeled (so that $x \in \Omega_n$) with label rate $\beta \in (0, 1)$, and with true, underlying labels given by the regions $\mathcal{X} = \mathcal{X}_+ \cup \mathcal{X}_-$, separated by a boundary $\mathcal{M}$, then we may ensure that $u(x) > \gamma$ for some $\gamma \in (0, 1)$ as long as $\text{dist}(x, \mathcal{M}) > r$ there exists an $r > 0$ of the order

$$r \in \mathcal{O}\left(\frac{\epsilon}{\sqrt{\beta}}\sqrt{\log d}\sqrt{\log\left(\frac{4}{1-\gamma}\right)}\right).$$

See Figure 4. The proof uses martingale techniques through the Random walk interpretation 1, and is largely inspired by the (fantastic) paper [4].

References

[1] X. Zhu, Z. Ghahramani, and J. Lafferty, *Semi-supervised learning using gaussian fields and harmonic functions*, 2003.

[2] K. Miller and J. Calder, *Poisson reweighted laplacian uncertainty sampling for graph-based active learning*, 2022. arXiv: 2210.15786 [stat.ML].

[3] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, *Playing atari with deep reinforcement learning*, 2013. [Online]. Available: https://arxiv.org/abs/1312.5602.

[4] J. Calder, D. Slepčev, and M. Thorpe, "Rates of convergence for laplacian semi-supervised learning with low labeling rates," *Research in Mathematics Sciences*, vol. 10, 2023.

The Github repositories containing the Active Grid environment and the DQN adaptation for SSL can be found at ://github.com/alexalbors7