Research Presentation

Active Learning with V-Learning

Thorben Werner

March 31, 2022

Information Systems and Machine Learning Lab (ISMLL)
Institute for Computer Science
University of Hildesheim

Pool-Based Active Learning

$\mathcal{L} \in \mathbb{R}^{\lambda \times m}$ Labeled Set

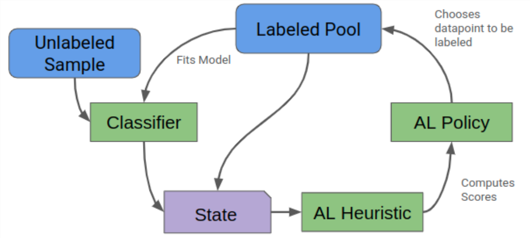
$\mathcal{U} \in \mathbb{R}^{\mu \times m}$ Unlabeled Set

$\phi_\theta := \mathbb{R}^m \rightarrow \mathbb{R}^c$ Classifier

$\mathcal{K} \in \mathbb{R}^{k \times m}$ Unlabeled Sample

$\psi := \mathbb{R}^{k \times m} \rightarrow \mathbb{R}^k$ Active Learning Heuristic

$\pi_\psi := \mathit{argmax}\ \psi(\mathcal{S})$ Active Learning Policy

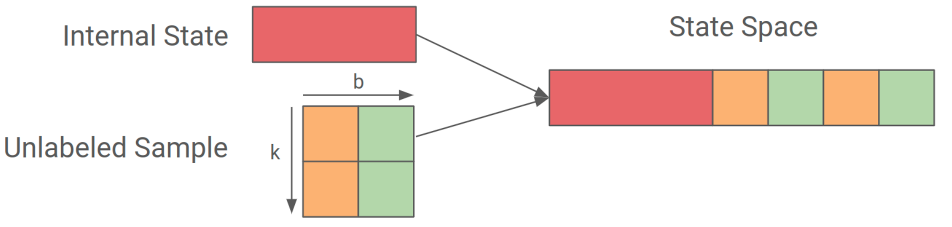


Active Learning Cycle

State Space:

1. Internal state of the environment: content of labeled pool $\mathcal{L}$, state of the classifier θ, remaining budget, current F1-Score, etc. $\rightarrow \mathbb{R}^{a}$
2. Information about the unlabeled sample $\mathcal{K}$: Output of the classifier $\phi_\theta(k_t)$, the datapoints themselves, other metrics, etc. $\rightarrow \mathbb{R}^{k \times b}$

Results in a flattened state space of $\mathcal{S} \in \mathbb{R}^{a+k \times b}$



Active Learning with Q-Learning

$\mathcal{A} \in [0, \ldots, k]$ Action Space

$\mathcal{R} := \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ Reward Function

$\tau := \{\mathcal{S}, \mathcal{A}, \mathcal{S}, \mathcal{R}, \mathbb{R}\}$ Transition

Problems:

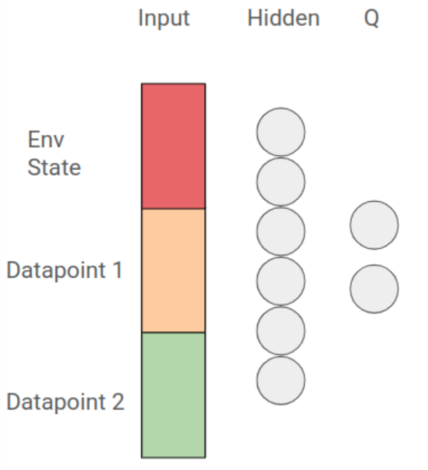
- Fixed Sample size
- Expensive Transitions
- Same actions in different places

Sample Size $k = 2$

The same datapoint can appear in multiple places in the input

Both output nodes essentially learn the same function since the datapoints are sampled randomly and independently.

A permutation invariant network can fix the problem and create a ranking of actions



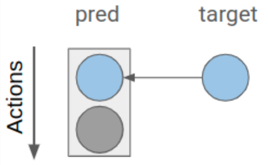
The Bellman Target

Q-Learning Target: What is the expected discounted improvement in F1-Score under the current policy when we choose a given datapoint?

The Bellman target does not rank actions, but tries to estimate their independent, global value under the current policy

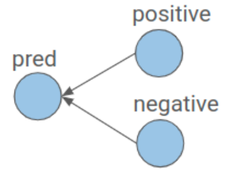
Reinforcement Learning

$$\mathcal{L}_Q := \hat{\pi}(s_t, a_t) - r_t + \gamma \max_a(\pi(s_{t+1}))$$



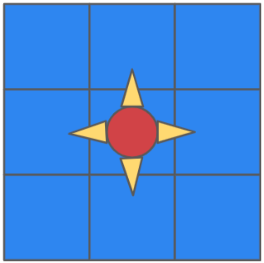
Ranking (Triplet Loss)

$$\mathcal{L}_{triplet} := max(0, m + d(z_a, z_p) - d(z_a, z_n))$$



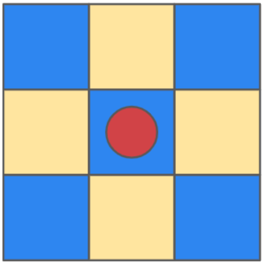
Q-Learning:

$Q(s_t, a) = \mathbb{R}^4$



V-Learning:

$V(s_{t+1}) = \mathbb{R}$



Nonetheless, both functions learn the value of a field in the maze under the current policy.

Active Learning with V-Learning

Adaptations

- We use the batch dimension for representing the sample size k
- No action space needed

$\mathcal{S} \in \mathbb{R}^{\sigma}$ State Space

$\mathcal{R} := \mathcal{S} \rightarrow \mathbb{R}$ Reward Function

$V_{\theta} := \mathcal{S} \rightarrow \mathbb{R}$ Agent Network

Example:

$$
\begin{aligned}
\text{State is generated} &: s \in \mathbb{R}^{b \times \sigma} \\
\text{Agents makes prediction} &: v = V_\theta(s) \\
\textbf{Policy selects a point} &: a = \operatorname*{argmax}_b v_b \\
\text{Action is applied} &: s' = \text{env}(a) \\
\text{Reward is observed} &: r = \mathcal{R}(s') \\
\text{Transition is stored} &: \tau = \{s_a, r, \bar{s}', \text{done}\} \\
\textbf{with} &: \bar{s}' = \frac{1}{b} \sum_{i=0}^{b} s'_i
\end{aligned}
$$

Storing Transitions in Memory

Q-Learning:

$$O(\tau) = s \times a \times s' \times r \times \mathbb{R}$$
$$O(\tau) = k^2 \times \sigma^2 + 3$$

V-Learning:

$$O(\tau) = s_a \times a \times \bar{s}' \times r \times \mathbb{R}$$
$$O(\tau) = 2 \times \sigma^2 + 3$$