Knows What It Knows: A Framework for Self-Aware Learning

Lihong Li **Michael L. Littman** **Thomas J. Walsh**

{lihong,mlittman,thomaswa}@cs.rutgers.edu

Rutgers Laboratory for Real-Life Reinforcement Learning (RL^3)

Department of Computer Science, Rutgers University, Piscataway, NJ, USA 08854

KWIK Overview

- ✓ **KWIK** = **K**nows **W**hat **I**t **K**nows
  - A framework for **self-aware** learning
- ✓ **Catalog** several basic KWIK learners
- ✓ **Combine** basic KWIK learners
  - to obtain more powerful KWIK learners
  - to unify and improve PAC-MDP RL

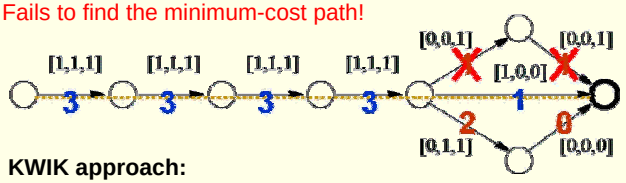
A Motivating Example

- Deterministic minimum-cost path finding
- Episodic task
- Edge cost = $x \cdot w^*$ where $w^*=[1,2,0]$
- Learner knows **x** of each edge, but not **w***
- Question: How to **find** the minimum-cost path?

"Traditional" approach:

Standard least-squares linear regression: $\hat{w} = [1,1,1]$.

Fails to find the minimum-cost path!

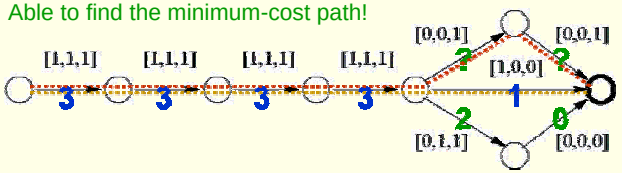


KWIK approach:

Reason about uncertainty in edge cost predictions.

Encourage agent to explore the unknown.

Able to find the minimum-cost path!

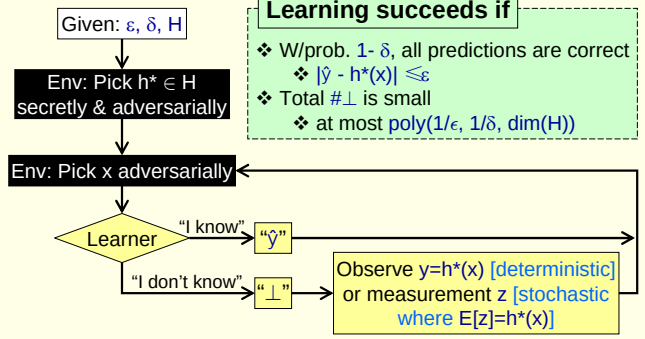


Relevant Scenarios

(where learner chooses samples/observations)

- **Selective sampling (active learning, in general)**: "only see a label if you buy it"
- **Bandit**: "only see the payoff if you choose the arm"
- **Reinforcement learning**: "only see transitions and rewards of states if you visit them"

KWIK Definition



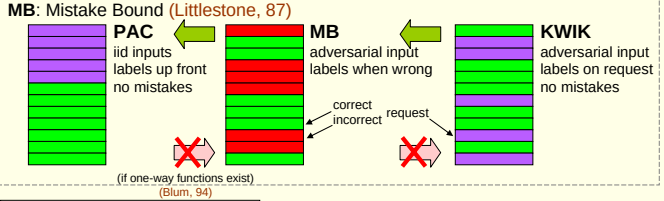
Learning succeeds if

- ❖ W/prob. 1- δ, all predictions are correct
  - ❖ $|\hat{y} - h^*(x)| \leq \varepsilon$
- ❖ Total #⊥ is small
  - ❖ at most poly(1/ε, 1/δ, dim(H))

Relation to Other Learning Models

PAC: Probably Approximately Correct (Valiant, 84)

MB: Mistake Bound (Littlestone, 87)



Example: Bar-Fight

- You own a bar frequented by n patrons…
- One is an **instigator (I)**. When he shows up, a fight breaks out, ***unless***
- Another patron, the **peacemaker (P)**, is also there.
- **Problem**: to predict, for a set of patrons, {**fight** or **no-fight**}

Alg: Memorization

- ◆ Memorize outcome for each subgroup of patrons
- ◆ Predict ⊥ if not seen before
- ◆ $\#\perp \leq 2^n$

Alg: Enumeration

- ◆ Enumerate all possible (I, P) combinations
- ◆ Say ⊥ when they disagree
- ◆ Eliminate inconsistent ones
- ◆ $\#\perp \leq n(n-1)$

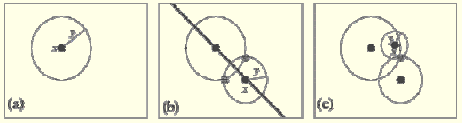
Example: Coin-Learning

- ◆ **Problem**: predict Pr(head) ∈ [0,1] for a coin flip
- ◆ But, observations are noisy: (**head** or **tail**)
- ◆ Predict ⊥ for the first $O(1/\varepsilon^2 \log(1/\delta))$ times (Hoeffding, 63)
- ◆ Use empirical estimate afterwards

Example: Distance-Learning

Problem: learn the distance to an unknown point (#⊥ ≤ 3)

Note: Can make accurate predictions before identifying h*.



Combining KWIK Learners by "Union"

Problem: KWIK-learn $H_1 \cup H_2$, where

$H_1 = \{f \mid f(x) = |x-c|, c \in \Re\}$

$H_2 = \{f \mid f(x) = mx+b, a,b \in \Re\}$

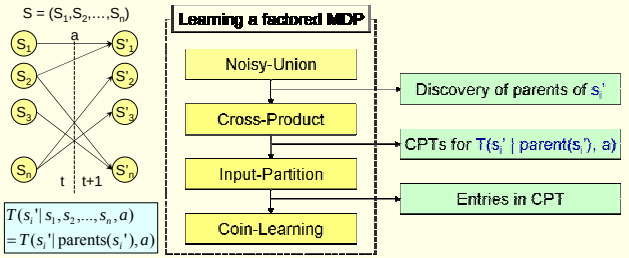
Note: Similar idea generalizes to the stochastic case.

Step	Input	L_1	L_2	Union	Observed Output
1	2	⊥	⊥	⊥	2
2	8	⊥	⊥	⊥	4
3	2	2	2	2	N/A
4	5	1	X	⊥	1
5	3	1	X	1	N/A

Finite MDP Learning by "Input-Partition"

- ◆ **Problem**: KWIK-learn transition probabilities of a Markov decision process ⟨ S, A, T, R ⟩
- ◆ T(s'|s,a): (unknown) transition probability (s,s' ∈ S, a ∈ A)
- ◆ Viewed as $|S|^2|A|$ instances of coin-learning
- ◆ Same insight as in prior PAC-MDP algorithms (Kearns & Singh, 02) (Brafman & Tenneholtz, 02)

Factored RL with Structure Learning



Note: Significantly improve on Strehl, Diuk, & Littman (07)

Open Challenges

- ➢ Systematic conversion from KWIK algorithms from deterministic cases to stochastic ones
- ➢ Unrealizable KWIK (where h* ∈ H is not guaranteed)
- ➢ Characterization of dim(H) for KWIK
- ➢ Relation between KWIK and active learning