Representation Learning for Generalized Planning - SAT Encodings

April 19, 2021

Learning of Generalized Features & Policy in Adversarial FOND domains

Encoding is parametrized by

- Pool F of description logic features f, each with given feature complexity $\mathcal{K}(f)$.
- Training set consisting of a sample of transitions from a number of instances of the same domain. At the moment we're assuming the sample is complete, and we have full information on whether each state in the sample is a goal state, an unsolvable state, or otherwise (see below for definitions). We also have access to the minimum distance to a goal $V^*(s)$ for each state s.
- A parameter δ which is a "slack" value to determine the maximum deviation from the optimal $V^*(s)$ what we will allow in our policy. This will be made clearer in the encoding below.

Terminology

A state is called *reachable* if there is a path to it from s_0, and it is called *solvable* if there is a path from it to a goal state. A state is *alive* if it is solvable, reachable and not a goal state [1].

We use T to denote the set of all transitions (s, s') in the training sample such that s is alive.

Improvements

Non-distinguishability of transitions as an equivalence relation. Any fixed, given pool of features F implicitly defines an equivalence relation where to transitions are equivalent iff they cannot be distinguished *by any feature in* F. If two transitions cannot be distinguished by any feature, then clearly either the policy computed by the SAT solver considers all of them as "good", or as "bad". We'll exploit this by using one single SAT variable to denote whether *all transitions in a given equivalence class* are good or bad. When exploiting this notion of equivalence (which is implemented as an optional feature of the CNF generator), then every mention below to SAT variable $Good(s, s')$ needs to be read as $Good(s_\star, s'_\star)$, where $(s_\star, s'_\star)$ is the *representative* transition of the equivalence class to which (s, s') belongs.

"Bad" transitions. We use BAD to denote the set of transitions that have been determined at preprocessing as necessarily *not* good. At the moment, this set contains all transitions that go from an alive to an unsolvable state and, if using equivalence relations, all those other transitions in whose equivalence class there is some other "bad" transition.

Feature dominance. Work In Progress.[1]

[1] This is implemented, but needs to be adapted to the equivalence classes improvement.

Variables

Main Variables

In any non deterministic domain the agent can go a pull of states S' by executing an action a in the state s, and then the nature or the adversary (doing anything) turns s' into s''. Then all the transitions holds in the form $s \rightarrow a \rightarrow s'$, where $s' \epsilon\, S'$.

- $Good(s, a)$ for s alive, s' solvable for all possible s' result of executing a in s.
- $Bad(s)$ for s alive.
- $V(s, d)$ for s alive, and $d \in [0, D]$, where $D = \max_s \delta \cdot V^*(s)$, with intended denotation $V(s) = d$. Note that for states s that are a goal, we know $V(s) = 0$, and for states s that are unsolvable, we know that $V(s) \neq d$ for all d. Thus, we can restrict SAT variables $V(s, d)$ to those states s that are alive.
- $Select(f)$, for each feature f in the feature pool.

Hard Constraints

C1. The policy has to be complete with respect to all possible actions applicable in s:

$$\bigvee_{a \text{ s.t. } (s,a,S') \in T} Good(s, a), \text{ for } s \text{ alive.} \tag{1}$$

C2. V is always descending along Good actions:

$$Good(s, a) \wedge V(s', d) \rightarrow \bigvee_{d<k\leq D} V(s, k), \text{ for } s, a \text{ alive, } (s, a) \notin \text{BAD}, d \in [1, D). \tag{2}$$

$$V(s', D) \rightarrow \neg Good(s, a), \text{ for } s, a \text{ alive, } (s, a) \notin \text{BAD}. \tag{2'}$$

C3. All descending transitions must be considered Good:

$$V(s, d) \wedge V(s', d') \rightarrow Good(s, a), \text{ for } s, s' \text{ alive, } (s, a, s') \notin \text{BAD}, 1 \leq d' < d \leq D, \tag{3}$$

$$Good(s, a), \text{ for } s \text{ alive, } s' \text{ goal.} \tag{3'}$$

C3-4. Variables $V(s, d)$ define a function that is total over the set of alive states, and such that $V(s)$ is within lower bound $V^*(s)$ and upper bound $\delta \cdot V^*(s)$:

$$\bigvee_{V^*(s)\leq d\leq \delta\cdot V^*(s)} V(s, d), \text{ for } s \text{ alive.} \tag{4}$$

$$\neg V(s, d) \vee \neg V(s, d') \text{ for } s \text{ alive, } 1 \leq d < d' \leq D. \tag{5}$$

C5-6. Good pairs (s, a) can be distinguished from bad pairs (s, a). Let (s, a) and (s, a') of two different equivalence classes such that $(s, a, S') \notin \text{BAD}$ (which implies that any $s' \in S'$ is solvable). Then,
where $Dist(s, s', t, t')$ is shorthand for $\bigvee_{f\in D1\&2(s,s',t,t')} Select(f)$.

C7 (Optional). Goals are distinguishable from non-goals.

$$\bigvee_{f\in D1(s,s')} Select(f), \text{ for } s \text{ goal, } s \text{ not a goal} \tag{6}$$

C8 (Optional). All selected features need to have some Good transition that takes them to 0:

$$Selected(f) \rightarrow \bigvee_{(s,s')\in Z(f)} Good(s, s'), \text{ for } f \text{ in pool} \tag{7}$$

where $Z(f)$ is the set transitions starting in an alive state that change the denotation of f from something larger than 0 to 0.

Soft Constraints

We simply post a constraint $\neg Select(f)$ for each feature f in the pool, with weight equal to its complexity $\mathcal{K}(f)$.

Empirical Results

References

[1] Guillem Francès, Augusto B. Corrêa, Cedric Geissmann, and Florian Pommerening. Generalized potential heuristics for classical planning. In *Proc. IJCAI 2019*, pages 5554–5561, 2019.