CSE 513T Theory of Artificial Intelligence & Machine Learning	Nov 29th, 2016
Lecture 25	
Lecturer: Brendan Juba	*Scribe: Shi Shu*

1 Today's Topics

1. Algorithm for integrated learning and planning
2. Analysis of part 1 of the proof ("soundness")

REMINDER: Reports for final projects due 12/8.

2 Recall: Key definition from last time

We say that a sequence of actions $a_1, ...a_t$ is $\mu - typical$ ($\mu \in [0,1]$) for reference policy Π_0 and POMDP(Partially observed Markov Decision Processes), if the probability that Π_0 takes the precise sequence of actions in the POMDP is $\geq \mu$.
Recall our task:
We are given an example history of a reference policy Π_0 in a factored $n - attribute$ POMDP(in which the observations are produced by a masking process.) We are given a goal DNF G and parameters μ(typicality), γ(accuracy), ε(error tolerence), and δ(confidence), all ≥ 0.
If there exist a rank-k T-horizon decision tree policy and an implicit KB CNF ψ such that on each branch of the policy:

1. Only $\mu - typical$ sequence of action appear and
2. There is a space-S($= 2$) tree like resolution proof from ψ that G is true on the branch(using the literal on the branch as additional premises).

Then we will find with probability $1-\delta$ a rank-k T-horizon decision tree policy that achieves G with probability $1-\alpha(n,T)(\varepsilon+\gamma)$ for some polynomial function $\alpha(_,)$ in polynomial time in $n, T, 1/\mu, 1/\gamma, log(1/\delta), |A|$ (exponential in S and k).
Example: Consider Random walk Π_0, $log(1/\mu)$ length actions sequences are typical.

3 Algorithm for integrated learning and planning

3.1 The Basic Structure

Follows the recursive strategy used to find a small-space tree like resolution proof.
Search for the smaller rank branch first.
Two main difference:

1. For us, a good branch is one on which G is achieved.
2. We must search over actions.

3.2 The algorithm: Find-DT

Algorithm 1 Find-DT

function FIND-DT($G, KB, k, t, \Pi, H, \mu\varepsilon, \delta, \gamma$)
  if Π takes more actions than the horizon of H **then**
    Return FAIL
  else
    Put $H^t \Leftarrow histories(^{(i)}_{t}, a^{(j)}_{(t-1)}{}^{(j)}_{t-1}...a^{(j)(j)}_{i\ i}) \Leftarrow H$
    with weight $w^{(j)} = \Pi_{i=1}^{(t-1)} w_i^{(j)}$ for histories
    with weight $w^{(j)} \leq \frac{4}{\mu}$
  if weighted reasoning with example on H^t for the query $\neg\Pi \vee G$ w/ KB returns $\hat{p} \geq 1-\varepsilon$ **then**
    Return Π
  if $k > 0$ for each literal l such that neither $l^{(t)}$ nor $\neg l^{(t)}$ appear on Π **then**
    Put $\Pi_l \Leftarrow \Pi \Rightarrow l \Rightarrow$[1]
    Put $\Pi_1 \Leftarrow$ Find-DT$(G, KB, k-1, \Pi_l, H, ...)$
    if $\Pi_1 \neq FAIL$ **then**
      $\Pi_0 \Leftarrow$ Find-DT$(G, KB, k, \Pi_{\neg l}, H, ...)$
      if $\Pi_0 \neq FAIL$ **then**
        Return policy [$l \Rightarrow \Pi_1$ if $l = 1$, $l \Rightarrow \Pi_0$ if $l = 0$]
      else
        Return FAIL
    for all action $a \in A$ such that $h^{(j)} \in H, \alpha_{a,t}^{(j)} = 1 \geq \frac{\mu}{2} m$ **do**
      Put $\Pi_\alpha \Leftarrow \Pi \Rightarrow \alpha \Rightarrow$
      Put $H_\alpha \Leftarrow h^{(j)} \Leftarrow H, \alpha_{a,t}^{(j)} = 1$
      Put $\Pi'_\alpha \Leftarrow$ Find-DT$(G, KB, k, t+1, \Pi_\alpha, H_\alpha, ...)$
      if $\Pi'_\alpha \neq FAIL$ **then**
        Return policy $\alpha \Rightarrow \Pi'_\alpha$
    Return FAIL

We initially call Find-DT with the empty branch, $t = 1$, and a sample of $m(n, T, |A|, \mu, \varepsilon, \delta, \gamma)$, histories H from Π_0.

4 Theorem for Find-DT

4.1 Theorem for Find-DT

Theorem: Given $m \geq \frac{8}{\mu^2\gamma}(2nTln3 + ln(\frac{4T|A|}{\mu}) + ln(\frac{3}{\delta}))$ example of Π_0 the algorithm runs in time $O(m|KB|(4nT)^{2k+2})$ and either returns FAIL or a rank-k T-horizon decision tree policy such that for each branch only $\mu - typical$ actions w.r.t Pi_0 are taken and such that for each branch, there is a $(1-\varepsilon+\gamma)$-testable CNF ψ such that there is a space-2 proof of a G from $\psi \wedge KB$ and the literals asserted on the branch, then with probability $1-\delta$ the algorithm returns a policy that achieves G with probability $1 - O((nT)^k(\varepsilon+\gamma))$.

4.2 Part 1 "Soundness" Proof

("When the algorithm returns a policy, it is a good one!")
Proof(Part 1):

Observe that there are at most $\frac{1}{\mu}$ $\mu - typical$ sequence of any fixed length.
Therefore there are at most $\frac{T}{\mu}$ $\mu - typical$ sequences.
There are at most $|A|T\frac{4}{\mu}$ sequences of actions are either $\frac{\mu}{4} - typical$ or a minimal prefix of a sequence that is not $\frac{\mu}{4} - typical$.
Note that: Since m is at least $\frac{8}{\mu^2\gamma}(2nTln3 + ln(\frac{4T|A|}{\mu}) + ln(\frac{3}{\delta}))$, Hoeffding inequality guarantees that the fraction of times that a sequence of actions appears in the sample is within an additive $\frac{\mu}{4}$ of its probability of being taken with probability $1 - \frac{\delta}{3|A|3^{2nT\frac{4T}{\mu}}}$.
So by a union bound over the sequences, $\geq \frac{3}{4}\mu - typical$ sequences appear at least $\frac{\mu}{2}m$ times and no $< \frac{\mu}{4} - typical$ sequences appears $\frac{\mu}{2}m$ times.
And by union bound over 3^{2nT} subsets of literals that may appear on a branch. By **HW4P3**, we find that our queries all estimates the probability that the branch covers the examples and G is falsified to within an additive γ with probability $1 - \frac{\delta}{3}$.
Claim 1: Find-DT only returns rank-k decision tree (or FAIL).
Proof of Claim 1: Proof by induction.
For $k = 0$, by induction on numbers of steps remaining.
It can only return an action followed by another rank-0 decision tree policy, which is another rank-0 policy.
By induction on k, and the number of literals remaining, we find that the algorithm either returns a policy that branches on a literal and (by $|H|$) then chooses either a rank$\leq k-1$ or a rank$\leq k$ policy and hence has rank$\leq k$ itself or returns a policy that takes an action and (by $|H|$) follows a rank$\leq k$ policy, which is also a rank$\leq k$ policy.
■
(**Recall** that rank-k decision trees on n' attributes have at most $n'^k + 1$ leaves. From **HW2**.)
Thus overall we have $O((nT)^k)$ leaves. So the probability that the policy would FAIL is $\leq O((nT)^k(\varepsilon + \gamma))$.
■

4.3 Part 2 "Completeness" Proof ideas

It doesn't hurt to include an additional test for a literal – each of the branches of the original(unknown) target decision tree policy will still pass the reasoning with examples query since the probability of $\neg\Pi \wedge G$ only measures as we add literals to Π. This is why we can make a single rank-k recursive call when some rank-$k-1$ call succeeds.
For actions, the fact that the recursive calls partition the histories allow us to bound the run time.
The actual proof to be continued...