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 \( a_1, \ldots, a_t \) is \( \mu \)-typical (\( \mu \in [0,1] \)) for reference policy \( \Pi_0 \) and POMDP (Partially observed Markov Decision Processes), if the probability that \( \Pi_0 \) 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 \( \Pi_0 \) 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 \( \mu \) (typicality), \( \gamma \) (accuracy), \( \varepsilon \) (error tolerance), and \( \delta \) (confidence), all \( \geq 0 \).
If there exist a rank-k T-horizon decision tree policy and an implicit KB CNF \( \psi \) 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 \( \psi \) 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 \( \alpha(\cdot) \) in polynomial time in \( n, T, 1/\mu, 1/\gamma, \log(1/\delta), |A| \) (exponential in \( S \) and \( k \)).

Example: Consider Random walk \( \Pi_0 \), \( \log(1/\mu) \) 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($G, KB, k, t, H, \mu, \delta, \gamma$)
    if $\Pi$ takes more actions than the horizon of $H$ then
        Return FAIL
    else
        Put $H^t \leftarrow histories_{i=t}^{(i)}, a_{(t-1)\ldots i}^{(j(j))} \subseteq H$
        with weight $w^{(j)} = \Pi_{i=1}^{(t-1)} w_i^{(j)}$ for histories
        with weight $w^{(j)} \leq \frac{n}{\mu}$
        if weighted reasoning with example on $H^t$ for the query $\neg \Pi \lor G$ w/ KB returns $\hat{p} \geq 1 - \varepsilon$ then
            Return $\Pi$
        if $k > 0$ for each literal $l$ such that neither $l^{(l(t)}$ nor $l^{(l(t)}$ appear on $\Pi$ then
            Put $\Pi_0 \leftarrow \Pi \Rightarrow l \Rightarrow 1$
            Put $\Pi_1 \leftarrow$ Find-DT($G, KB, k - 1, \Pi_1, H, ...$)
            if $\Pi_1 \neq FAIL$ then
                $\Pi_0 \leftarrow$ Find-DT($G, KB, k, \Pi, H, ...$)
                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 $h^{(j)} \in H, a_{\alpha}^{(j)} = 1 \geq \frac{\mu}{2} m$ do
                Put $\Pi_\alpha \leftarrow \Pi \Rightarrow a \Rightarrow$
                Put $H_\alpha \leftarrow h^{(j)} \subseteq H, a_{\alpha}^{(j)} = 1$
                Put $\Pi_\alpha' \leftarrow$ Find-DT($G, KB, k, t + 1, \Pi_\alpha, H, ...$)
                if $\Pi_\alpha' \neq FAIL$ then
                    Return policy $a \Rightarrow \Pi_\alpha'$
            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 $\Pi_0$.

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 $\Pi_0$ 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 $P_{\Pi_0}$ are taken and such that for each branch, there is a $(1 - \varepsilon + \gamma)$-testable CNF $\psi$ such that there is a space-2 proof of a G from $\psi \land KB$ 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 \( \frac{1}{n} \mu - \text{typical} \) sequence of any fixed length.

Therefore there are at most \( \frac{T}{n} \mu - \text{typical} \) sequences.

There are at most \(|A|T\frac{2}{n}\) sequences of actions are either \( \frac{\mu}{4} - \text{typical} \) or a minimal prefix of a sequence that is not \( \frac{\mu}{4} - \text{typical} \).

Note that: Since \( m \) is at least \( \frac{8}{\delta^2} (2nT\ln 3 + \ln(4T|A|) + \ln(\frac{3}{2})) \), 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 - \( \frac{\delta}{3|A|3^{2nT}} \).

So by a union bound over the sequences, \( \geq \frac{3}{4}\mu - \text{typical} \) sequences appear at least \( \frac{\mu}{2} \) times and no \( < \frac{\mu}{4} - \text{typical} \) sequences appear \( \frac{\mu}{2} \) 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 \( \gamma \) 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 \( k - 1 \) or a rank \( k \) policy and hence has rank \( k \) itself or returns a policy that takes an action and (by \(|H|\)) follows a rank \( k \) policy, which is also a rank \( k \) 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 \( \neg \Pi \land \neg G \) only measures as we add literals to \( \Pi \). 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...**