Decision Making under Uncertainty
ASEN 6519-007
Maggie Jacoby **Homework 6** April 10, 2020

Conceptual Questions

Problem 1: Particle Filter on Tiger POMDP

Given
A 3-door version of the tiger POMDP, perform a belief update based on Particle Filter without Rejection (Algorithm 6.3). Use initial particle beliefs of $s_1 = 1$, $s_2 = 2$, and $s_3 = 3$, and intial noise terms as $v_1 = 0.01$, $v_2 = 0.32$, and $v_3 = 0.74$.

Find

(a) List the state particles in the updated belief approximation, $\hat{b}_1$.

(b) If the true belief were $\hat{b}_1$, what would the optimal action be? Given that $\hat{b}_1$ is only an approximation, why is this a poor choice? How would you change the belief update approach to avoid this?

Solution

(a) Given the starting beliefs and the specified noise terms, the updated states are:

$$s'_1 = 2,\ s'_2 = 2,\ s'_3 = 3$$

The observation at this step is o=1, which means $o \neq s'_i\ \forall i$, thus, $Z(o) = 0.1$ for all observations, so the weights in line 6 are equally distributed.
I randomly choose the new states based on the the prevalence of the states from line 6, such that:

`bp = [rand([2,2,3]), rand([2,2,3]), rand([2,2,3])]`,
which yielded $b'_1 = 3$, $b'_2 = 2$, $b'_3 = 3$.

Based on my method, my updated belief is: $\hat{b}_1 = (3,2,3)$.

(b) Based on this new belief, we are somewhat certain that the tiger is behind door 3, with a secondary belief that it is behind door 2. We would not think that it is behind door 1, and so would be likely to open door 1. Our observation **was** that the tiger was behind door 1, and so this would be a poor option, as we might get eaten. We got to this state with three low probability observations, and without a high likelihood of observations, and so I would change the problem to not allow an "open-door" action **without** an observation agreeing with the new state. I would do this by implementing a rule that unless you receive one majority weight in line 6, the next action must be listen.

Code for Question 1

```
using POMDPs
using QuickPOMDPs


"""
    Suppose that you decide to use a particle filter with 3 particles (| b | = 3)
    for approximate belief updates. The initial particle-based belief approximation
    is  b0  = {1,2,3}. Perform a single belief update (following Algorithm 6.3)
    with a = 4, o = 1, using G from (1) in Alg 6.3, line 5. Use the following
    outcomes for the random parts of the algorithm:

            Inline 4: s1 =1,s2 =2,s3 =3
            Inline 5,thevalues of v input to G are: v1 =0.01,v2 =0.32,v3 =0.74.
            In line 8, select the new states however you would like.
"""


tiger = QuickPOMDP(
    function (s, a, v=rand())
        ov = rand()
        v < 0.05 ? sp = mod(s,3)+1 : sp = s
        a == 4 ? r = 0.0 : a == s ? r = -100.0 : r = 10.0
        ov > 0.8 ? o = sp : o = rand(setdiff(Set([1,2,3]), sp))
        return (sp=sp, r=r, o=o)
    end,

    states = [1, 2, 3],
    actions = [1, 2, 3, 4],
    observations = [1, 2, 3]
);

function UpdateBelief(b=(1,2,3), a=4,o=1) # starting with b_0=(1,2,3)
    Gp(s, v) = v < 0.05 ? mod(s,3)+1 : s
    # w(si, o ) = o == si ? 0.8 : 0.1
    # bp = (0,0,0)
    s = (1, 2, 3)
    v = (0.01, 0.32, 0.74)
    sp = [Gp(s[i], v[i]) for i in 1:3]
    #wi = [w(si[i], 0) for i in 1:3]
    bp = [rand([2,2,3]), rand([2,2,3]), rand([2,2,3])]
    return bp
end

b1 = UpdateBelief();
```

CODE OUTPUT

```
    julia> b1
    3-element Array{Int64,1}:
     3
     2
     3
```

Exercises

Problem 2: SASOP vs QMDP on Tiger and Cancer POMDPs

Given
Cancer POMDP from HW 5 and standard Tiger problem.

Find
Solve both MDPs with QMDP and SARSOP and compare results.

Solution The values I received are in the table below. As expected, on the tiger problem QMDP and SARSOP achieve very similar scores. This is because, in the tiger problem, not doing anything is not costly. As I described above, if you are not very certain of the correct state, it is extremely costly to open to wrong door, and so listening (not doing anything) is often a good choice. Since QMDP assumes we will have full knowledge in the next step, it doesn't prematurely open a door, and more information can be gained by waiting. Thus, it doesn't matter what you believe, as long as you don't act.

However, for the cancer problem, waiting can be very costly. QMDP assumes perfect knowledge in the next step, but knowledge will not increase unless you take the test action. By not doing anything in the cancer problem you will not gain more information. Thus, waiting and not testing or treating can be very costly if the true state is cancer. So a false belief, coupled with inaction, will not result in a good outcome.

	Cancer	Tiger
SARSOP	78.0	19.6
QMDP	57.8	19.0

Code for Question 2

```
using POMDPs
using POMDPModels
using POMDPSimulators
using POMDPModelTools
using BeliefUpdaters
using POMDPPolicies
using QuickPOMDPs
using BeliefUpdaters
using SARSOP
using QMDP
include("HW5_soln.jl")


c = DiscreteExplicitPOMDP(S, A, O, T, Z, R, 0.99,
                            Deterministic(:healthy),
                            terminals=Set([:death]));
t = TigerPOMDP();

function evalSolver(m)
    results = Dict{String, Float64}()
    for (key, solver) in ["SARSOP"=>SARSOPSolver(), "QMDP"=>QMDPSolver()]
        policy = solve(solver, m)
        N = 10000
        rsum = 0.0
        for i in 1:N
            rsum += simulate(RolloutSimulator(max_steps=500), m, policy)
        end
        results[key] = rsum/N
    end;
    return results
end

results_c = evalSolver(c);
results_t = evalSolver(t);
```

CODE OUTPUT

```
julia> results_t
Dict{String,Float64} with 2 entries:
  "QMDP"   => 19.0395
  "SARSOP" => 19.5645

julia> results_c
Dict{String,Float64} with 2 entries:
  "QMDP"   => 57.8496
  "SARSOP" => 78.0485
```

Challenge Problem

Problem 3: Lasertag POMDP

Submitted to leaderboard

Code for Question 3

```
using POMDPs
using POMDPModels
using POMDPSimulators
using SARSOP
using QMDP
using BasicPOMCP
using DMUStudent
using DMUStudent.HW6

using ParticleFilters
using POMCPOW


# solver = POMCPOWSolver(criterion=MaxUCB(26.0), k_action=4.0, k_observation=4.0)

Q_planner = solve(QMDPSolver(), lasertag)
up = SIRParticleFilter(lasertag, 1000)



DMUStudent.submit(
    (Q_planner,up),
    "hw6",
    "maja8167@colorado.edu",
    nickname="mhsj"
)

ds = DisplaySimulator(max_steps=30, extra_final=false, extra_initial=true)
simulate(ds, lasertag, Q_planner)

DMUStudent.evaluate(
    (Q_planner,up),
    "hw6",
    n_episodes=100
)
```