Temporal Logic Motion Planning and Control with Probabilistic Satisfaction Guarantees

Jatin, Vibhor, Sahil

Instructor: Indranil Saha

March 27, 2019

Motivation

- Till now, we have done motion planning under two fundamental assumptions:
  - purely deterministic or,
  - purely non-deterministic (with no information on the likelihoods of those transitions).
- However, noise in the actuators and sensors can invalidate both these assumptions.
- The probabilities of each transition can be found by using repeated simulation and experiments.
- We also assume that robot can find its current position precisely.
- Sample Problem: Find the maximum probability of reaching Destination by always avoiding Unsafe regions.

Markov Decision Process (MDP) $\mathcal{M}$ is a tuple (S,s_0,Act,T,Π,L,c) where,

- S is a finite set of states
- s_0 $\in$S is the start state
- Act is finite set of actions
- $T : S \times Act \times$ S $\rightarrow$ [0,1] is transition probability function such that for each s $\in$ S and α $\in$ Act, either T(s,α,.) is a probability distribution on S, or T(s,α,.) is the null function.
- Π is a finite set of atomic propositions
- L:S$\rightarrow$ 2^{Π} is the labelling function assigned to each s in S.
- Cost, c: $S \times Act \rightarrow \mathbb{R}^{\geq 0}$

MDP Example

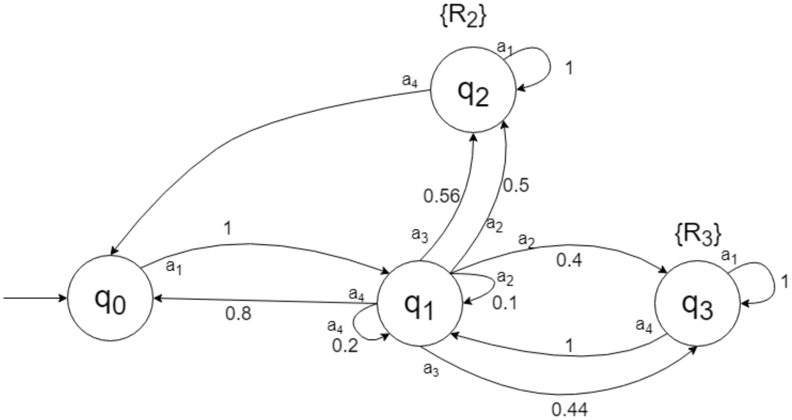

Figure: MDP

- Syntax of PCTL:
  - $\phi ::= true \mid \pi \mid \neg\phi \mid \phi \wedge \phi \mid P_{\bowtie p}[\psi] \mid \varepsilon_{\bowtie c}[\phi]$ state formulas
  - $\psi ::= X\phi \mid \phi\mathcal{U}^{\leq k}\phi \mid \phi\mathcal{U}\phi$ path formulas

  where $\bowtie \in \{\leq, <, \geq, >\}, p \in [0,1], c \in [0,\infty]$ and $k \in \mathbb{N}$
- Semantics of PCTL:
  For any state $s \in S$, the satisfaction relation $\models$ is defined inductively as follows:
  - s $\models$ true for all $s \in S$;
  - s $\models \pi \Leftrightarrow \pi \in L(q)$;
  - s $\models (\phi_1 \wedge \phi_2) \Leftrightarrow s \models \phi_1 \wedge s \models \phi_2$;
  - $s \models \neg\phi \Leftrightarrow s \not\models \phi$;
  - $s \models P_{\bowtie p}[\psi] \Leftrightarrow \exists\ \mu$ s.t. $p_{\mu}^{s} \bowtie p$;
  - $s \models \varepsilon_{\bowtie c}[\phi] \Leftrightarrow \exists\ \mu$ s.t. $e_{\mu}^{s} \bowtie c$;

- p_μ^s is the probability of all the infinite paths that start from s and satisfy ψ under policy μ and $e_\mu^s(\phi)$ denotes the total expected cost of reaching a state that satisfy ϕ from s under μ. Moreover, for any path $\omega \in$ Path:
  - $\omega \models X\phi \Leftrightarrow w(1) \Leftrightarrow \phi$
  - $\omega \models \phi_1 \mathcal{U}^{\leq k} \phi_2 \Leftrightarrow \exists i \leq k, \omega(i) \models \phi_2 \wedge w(j) \models \phi_1 \forall j < i;$
  - $\omega \models \phi_1 \mathcal{U} \phi_2 \Leftrightarrow \exists k \geq 0, \omega \models \phi_1 \mathcal{U}^{\leq k} \phi_2$

- A control policy is a function $A : Path^{fin} \rightarrow Act$. For every finite path, a policy specifies the next action to be applied.
  Under a policy A, an MDP becomes a Markov Chain denoted by D_A.
- There are two kinds of control policies - *History Dependent* and *Stationary*.

Optimal control synthesis for Simple PCTL

Next operator

- **Next Optimal-** $\phi = P_{max=?}[X\phi_1]$:
  - First find the set Sat(ϕ_1), the set of states that satisfy ϕ_1.
  - Let $\sigma_a^{s_i}(s_j)$ denotes the transition probability of making a transition from s_i to s_j on action a i.e. $\sigma_a^{s_i}(s_j)$ = T(s_i,a,s_j).
  - Probability that ϕ_1 is satisfied in next state on action a is(current state is s_i)

$$\sigma_a^{s_i}(Sat(\phi_1)) = \sum_{s_j \in Sat(\phi_1)} \sigma_a^{s_i}(s_j)$$

  - Then the optimal probability $(x^*_{s_i})$ of satisfying ϕ at state s_i can be find out using:

$$x^*_{s_i} = \max_{a \in A(s_i)} \sum_{s_j \in Sat(\phi_1)} \sigma_a^{s_i}(s_j)$$

  - The corresponding optimal policy is:

$$\mu^*(s_i) = arg \max_{a \in A(s_i)} \sum_{s_j \in Sat(\phi_1)} \sigma_a^{s_i}(s_j)$$

Example for Next Optimal

- Consider the MDP given in Fig 1, take $\phi_1 = P_{max=?}[X(\neg R_3)]$
- Now, $Sat(\phi_1) = \{q_0, q_1, q_2\}$.
- We can see that
  $\sigma_{a_2}^{q_1}(Sat(\phi_1)) = 0.6$,
  $\sigma_{a_3}^{q_1}(Sat(\phi_1)) = 0.56$,
  $\sigma_{a_4}^{q_1}(Sat(\phi_1)) = 1$.
- Thus, $x^*_{q_1} = 1$ and $\mu^*_{q_1} = a_4$.

- **Next all-** $P_{\bowtie p}[X\phi_1]$: where $\bowtie \in \{\leq, <, \geq, >\}$
  - Aim to find all the policies that satisfy the formula.
  - Find the probabilities of satisfying $X\phi_1$ for each state–action pair, using the *Next Optimal Algorithm*.
  - Eliminate all the policies that are not in the range of $\bowtie p$.

- Bounded Until Optimal - $\phi = P_{max=?}[\phi_1 \mathcal{U}^{\leq K} \phi_2]$:
  - First group the MDP states into three subsets.
    - $S^{yes} = Sat(\phi_2)$
    - $S^{no} = S \setminus (Sat(\phi_1) \cup Sat(\phi_2))$
    - $S^{?} = S \setminus (S^{yes} \cup S^{no})$
  - Example: $P_{max=?}[true\ \mathcal{U}^{\leq 2} R_3]$
    - $S^{yes} = \text{Sat}(R_3) = q_3$
    - $S^{no} = \phi$
    - $S^{?} = \{q_0, q_1, q_2\}$
  - $x_{s_i}^j$ denotes the maximum probability of satisfying the formula $(\phi_1 \mathcal{U}^{\leq j} \phi_2)$ from state s_i.
  - Hence, $x_{s_i}^j = 1 \ \forall\ s_i \in S^{yes}, \forall j$ and $x_{s_i}^j = 0 \ \forall\ s_i \in S^{no}, \forall j$
  - The probabilities for the remaining states $s_i \in S^?$ are defined recursively.

Finding maximum probabilities for $s_i \in S^?$:

- $x_{s_i}^0 = 0 \; \forall s_i \in S^?$
- $\forall \; k > 0, s_i \in S^?$
  - $x_{s_i}^k = \max_{a \in A(s_i)} \left(\sum_{s_j \in S^?} \sigma_a^{s_i}(s_j) x_{s_j}^{k-1} + \sum_{s_j \in S^{yes}} \sigma_a^{s_i}(s_j) \right)$
  - $\mu_{s_i}^{*^k} = arg\,\max_{a \in A(s_i)} \left(\sum_{s_j \in S^?} \sigma_a^{s_i}(s_j) x_{s_j}^{k-1} + \sum_{s_j \in S^{yes}} \sigma_a^{s_i}(s_j) \right)$
- We can solve the minimization problem similarly.

Example

$P_{max=?}[true\ \mathcal{U}^{\leq 2} R_3]$:

- $S^{yes} = \text{Sat}(R_3) = q_3$
- $S^{no} = \phi$
- $S^{?} = \{q_0, q_1, q_2\}$

k	x_0^k	x_1^k	x_2^k
0	0	0	0
1	0	0.44	0
2	0.44	0.444	0

- In above table, for k=2,
  $x_1^k = \max(0.44 \text{ (corr. to } a_3), 0.444 \text{ (corr. to } a_2))$.

- **Bounded Until Stationary-** $P_{\bowtie p}[\phi_1 \mathcal{U}^{\leq k} \phi_2]$: where $\bowtie \in \{\leq, <, \geq, >\}$
  - Aim to find all the policies that satisfy the formula.
  - Find the probabilities of satisfying $(\phi_1 \mathcal{U}^{\leq k} \phi_2)$ for each state–action pair, using the *Bounded Until Optimal Algorithm* with the exception that the optimal actions determined at each iteration are fixed for the remaining iterations.
  - Eliminate all the policies that are not in the range of $\bowtie p$.

- Unbounded Until Operator - $\phi = P_{max=?}[\phi_1 \mathcal{U} \phi_2]$:
  - First group the MDP states into three subsets: $S^{yes} = Sat(\phi_2)$, S^{no} using Algorithm 2, $S^{?} = S \setminus (S^{yes} \cup S^{no})$

Algorithm 1 Find $S^{no} = \{s \in S \mid p_q^{max}(\phi_1 \mathcal{U} \phi_2) = 0\}$

R $=$ Sat(ϕ_2)
done $=$ **false**
while done $=$ **false do**
  $R' =$ R $\cup$ $\{s \in Sat(\phi_1) \mid \exists a \in Act(s), \exists q' \in$ R s.t. $\sigma_a^q(q') >0$ $\}$
  if $R' =$ R **then**
    done $=$ true
  end if
  R $= R'$
end while
return Q$\backslash R$

Optimal control synthesis for Simple PCTL
continued..

- Consider the example: $P_{max=?}(\neg R_3 \mathcal{U} R_2)$
- $S^{yes} = Sat(\phi_2) = \{q_2\}, Sat(\phi_1) = \{q_0, q_1, q_2\}$
- Finding S^{no}: Initially $R = \{q_2\}$. After first iteration $R = \{q_1, q_2\}$. After second iteration $R = \{q_0, q_1, q_2\}$. Hence $S^{no} = \{q_3\}$.
- To compute the optimal probabilities for the states in $S^?$, we will use linear programming problem which is in fact maximum reachability problem.

- minimize

$$\sum_{s_i \in S^?} x_{s_i}$$

subject to:

$$x_{s_i} \geq \sum_{s_j \in S^?} \sigma_a^{s_i}(s_j).x_{s_j} + \sum_{s_j \in S^{yes}} \sigma_a^{s_i}(s_j)$$

for all $s_i \in S^?$ and $(a, \sigma_a) \in Steps(s_i)$

- Linear Equations:

$$x_0 \geq x_1$$
$$x_1 \geq 0.1x_1 + 0.5$$
$$x_1 \geq 0.2x_1 + 0.8x_0$$
$$x_1 \geq 0.56$$

- Solution to this will be $x_0 = x_1 = 0.56$.

Complex PCTL Formulas

- The formulas which consists of more than one P or ε operator.
- The problem can be solved if all the inner P operators are of the form $P_{\bowtie}[\phi]$ rather than of the form $P_{max=?}[\psi]$
- The general nested formula are of the form:
  1. $\phi = P_{max=?}[X\phi_R]$
  2. $\phi = P_{max=?}[\phi_L \mathcal{U}^{\leq k} \phi_R]$
  3. $\phi = P_{max=?}[\phi_L \mathcal{U} \phi_R]$

Complex PCTL Specification

1. $\phi = P_{max=?}[X\phi_R]$:
   - Find Optimal control Policy for ϕ_R and use the *Next Optimal Algorithm* on this.
2. $\phi = P_{max=?}[\phi_L \mathcal{U}^{\leq k} \phi_R]$
3. $\phi = P_{max=?}[\phi_L \mathcal{U} \phi_R]$:
   - States satisfying ϕ_R are directly marked.
   - Find Stationery Control Policy using the algorithms described above for ϕ_L.
   - Remove all the other actions from MDP which are not part of the above policy.
   - Now this problem can be solved using the above algortihms for simple PCTL formulas.

Extensions

- Dynamic Environment: [2]
  - Finding a control strategy that maximizes a PCTL specification in a dynamic environment composed with doors (which can open/close in sync with robot transition) under different levels of assumptions:
    - With prior knowledge on probability of door being open/closed and perfect sensing of the state of door.
    - With only perfect sensing of the state of door.
    - With error in sensing the state of door (Most Realistic Case).
- Continuous-time Dynamic Environment with time-bounded specification: [3]
  - Problem Statement being same as above, but here doors need not make transition in sync with robot motion.
  - Change in state of door is assumed to follow poisson distribution.
  - The time for which robot stays in a given region is modeled as exponential distribution.

Extensions

continued . . .

- Presence of other agents: [5]
  - To generate optimal control policy for a robot in presence of independent uncontrollable agents in a graph like environment when mission specification is given as an sc-LTL formula.
  - For example, consider a setting where a car(robot) is required to go from one location to another without colliding with the pedestrians(agents) crossing the road can be formulated as the above problem.
- Uncertain Environment: [4]
  - To generate a control strategy for a robot in presence of non-determinism not only in the robot motion, but also in the robot's observation of properties in the environment.
  - For example, we can model a task where a robot is operating in an indoor environment, and is required to pick-up and deliver items among some rooms. Non-determinism occurs in observations because items may or may not be available when a robot visits a room.

References

[1] Morteza Lahijanian, Sean B. Andersson, Calin Belta. "Temporal Logic Motion Planning and Control With Probabilistic Satisfaction Guarantees" (Presented in the presentation)

[2] I. Medina Ayala, Sean B. Andersson, Calin Belta. "Temporal logic control in dynamic environments with probabilistic satisfaction guarantees"

[3] I. Medina Ayala, Sean B. Andersson, Calin Belta. "Probabilistic control from time-bounded temporal logic specifications in dynamic environments"

[4] Xu Chu Ding, Stephen L. Smith, Calin Belta, Daniela Rus "LTL Control in Uncertain Environments with Probabilistic Satisfaction Guarantees"

[5] Alphan Ulusoy, Tichakorn Wongpiromsarn, Calin Belta. "Incremental control synthesis in probabilistic environments with Temporal Logic constraints"

Thank You

Questions?