Representing belief space for mobile manipulation

Leslie Pack Kaelbling
Tomas Lozano-Perez
MIT CSAIL
Knowledge representation in complex domains

1 Where’s the fork?
Real robot meets real world: uncertainty

Current-state uncertainty
- What is inside the tupperware?
- Is the dishwasher clean?
- What is the exact pose of the pot?
- What’s the friction of a wet dish?

Predictive uncertainty
- What will happen when the robot lifts the cookie sheet?
- What is the error in the motor control?
- When will the inhabitants come home?

Reason explicitly about uncertainty: Know what you know!
Pay attention to M&M paying attention
POMDP: Optimal solution is complex...

Policy is a total function
Solution to MDP in B space
Hierarchical planning

- maintain left expansion of plan tree
- each level uses a higher-fidelity model
- recursively plan to achieve those preconditions
- execute primitives
Replanning

• Job of action selection module: select next action
• Plan is a justification for the first action
• Use a simplified model for planning
• After action is executed, update belief, and replan

• Major simplification: deterministic dynamics
  • Assume most likely observation
  • Or, select desired observation with cost –log P(o)
Logical representation of beliefs and dynamics

Factoring:
- describe state in terms of state variables or propositions
- can induce factoring in belief state representation

Lifting:
- aspects of the dynamics do not depend on which particular objects are involved; just their properties

Infinite / indefinite domains
- we may not be able to enumerate all the individuals in advance: objects, regions of space, ...

Short names for big sets of belief states:
“"I think I’m in San Francisco.""
Representation for forward search: primitive

Joint commands

Geometric, continuous belief state representation

Has(tomas, beer)
Cannot represent all geometric and continuous aspects of the current belief state symbolically.
Representation for backward search: goals

Logical fluents only need to be:
• **tested** in a detailed continuous representation of the belief
• **regressed** backward through actions
Pre-image back-chaining

Weakest precondition of goal set under each action sequence

Test whether start belief is in a pre-image

Represent goal and pre-images as conjunctions of logical expressions

Belief-space pre-image of goal set $G$ under action $a$:

$$ R(G, a) = \{ b \mid \text{SE}(b, a, o^*(b, a)) \in G \} $$
Making pre-image back-chaining work

Success of the approach hinges on:

- heuristic for 'suggesting' values of free variables in operators that are likely to connect well with the initial state
- finding domain vocabulary in which good approximations of pre-images have compact representations
Fluents describe **sets** of belief states

Simple example:
- three possible states
- belief space is 3-simplex

\[ \text{KLoc}(o, l, p) \equiv \text{Pr}(\text{Loc}(o) = l) > p \]

\[ \text{KVLoc}(o, p) \equiv \exists l. \text{Pr}(\text{Loc}(o) = l) > p \]
Fluents for beliefs on continuous spaces

\[ KVLoc(O, \epsilon, \delta) \equiv Pr(\|\text{Loc}(O) - \mu(\text{Loc}(O))\| > \delta) < 1 - \epsilon \]

For a one-D Gaussian, probability near mean:

\[ PNM(X, \delta) = \Phi \left( \frac{\delta}{\sigma} \right) - \Phi \left( -\frac{\delta}{\sigma} \right) = \text{erf} \left( \frac{\delta}{\sqrt{2}\sigma} \right) \]

Set of distributions:

\[ PNM(X, 0.05) > 0.95 \]
PNM Regression

Regression of $PNM(X, \delta) > \theta_{t+1}$ through observation action is $PNM(X, \delta) > \theta_t$

$$
\theta_t = \text{erf} \left( \sqrt{\text{erf}^{-1}(\theta_{t+1})^2 - \frac{\delta^2}{2\sigma_o^2}} \right)
$$

Set of distributions in pre-image of obs with stdev 0.4: $PNM(X, 0.05) > PNM\text{Regress}(0.05, 0.4, 0.95)$
Probability near desired value

\[ K_{\text{Loc}}(O, L, \epsilon, \delta) \equiv \Pr(|\text{Loc}(O) - L| > \delta) < \epsilon \]

For a one-D Gaussian, probability near value 2:

Set of distributions:
\[ \text{PNV}(X, 2, 0.5) > 0.95 \]
PNV Regression: Observation

\[ K_{\text{Loc}}(O, L, \epsilon, \delta) \equiv \Pr(|\text{Loc}(O) - L| > \delta) < \epsilon \]

Set of distributions such that:
- after observation with sigma 0.4
- \( \text{PNV}(2, 0.5) > 0.95 \)
State estimation in the world

- Joint tangent-space Gaussian distribution on \((X,Y,Z,\Theta)\) poses of robot base and all objects
- Updated based on transitions and observations using EKF
- Assumes objects are recognizable in point cloud data
Fluents for mobile manipulation

- KVCondPose(Obj, Ref, Eps, Delta)
- KCondPose(Obj, Pose, Ref, Eps, Delta)
- PoseMeanNear(Obj, Pose, Delta)
- KCondRobotConf(Conf, Ref, Eps, Delta)
- RobotConfNear(Conf, Delta)
- KContents(Region)
- NotKNotClear(Region)
- KClearX(Region, Exceptions)

Conditional distribution: e.g., dist on object given robot pose
not KContents(Region) or KClear(Region)
Kcontents(Region) and nothing overlaps
Representing known space

Geometric reasoning about visibility
Planning operator in belief space: Pick

Pick(O, ObjPose):
pre: KVCondPose(O, ‘table’, bigEps, planDelta)

exists: ObjPose ∈ \{modeObjPose\} U generateParking(O)
        P ∈ generatePickPaths(ObjPose)

pre: KClearX(sweptVol(P), O)
       K Holding(None)
       KCondPose(O, ObjLoc, ‘table’, eps, graspDelta)
       KCondRobotConf(O, baseConf(P), smallEps, graspDelta)

result: K Holding(O)

Need to know O pose roughly before further planning

Suggest approximate robot path quickly

Object is close to where we first saw it

Detailed prim execution depends on detailed beliefs: pose, shape, mass, etc

Robot base is in position and well localized wrt O
Planning operator in belief space: Look

Look(0):
pre: PoseMeanNear(0, OPose, Delta)
exists: (Rconf, ViewCone) ∈ generateViewPose(0, OPose)
pre: NotKNotClear(ViewCone)
   KCondPose(0, OPose, RefObj, PNRegress(eps, obsVar, Delta), Delta)
   KCondRobotConf(0, RobotConf, smallEps, lookDelta)
result: KCondPose(0, OPose, RefObj, Eps, Delta)
Example problem

Object not in initial model

Goal: Believe blue object is here

Substantial uncertainty about poses of known objects

Contents of most of space not known
Execution: observation and motion
Execution: belief and planning
Next steps

- Structured world model: rooms, containers, ...
- Representing and reasoning about what space is known to be free
- Integrating more sensory modalities
- Domains with more non-geometric properties
- Integrating other actions: compliant motions, pushing, ...
Tiny domain

State: object is in L0, L1, or L2

Actions:
  * Look in L0, L1, or L2
    * FalsePos = 0.1, FalseNeg = 0.2
  * Move obj from Start to Goal
    * Fails w.p. 1.0 if obj not in Start
    * Fails w.p. 0.2 otherwise

Observations:
  * After look action, see the object, or not see it

Goal:
  * Believe, with probability > 0.95, that object is in L0
Planning operators in belief space

Move(0, TargetLoc):
  \textbf{result}: KLoc(0, TargetLoc, Prob)
  \textbf{exists}: StartLoc
  \textbf{pre}: KLoc(0, StartLoc, Prob / (1-MoveFailProb))
  \textbf{cost}: -\log(Prob / (1-MoveFailProb))

Look(Loc):
  \textbf{result}: KLoc(0, Loc, Prob)
  \textbf{pre}: KLoc(0, Loc, lookRegressProb(Prob))
  \textbf{cost}: -\log(lookRegressProbProb(Prob))

\begin{equation}
\text{lookRegressProbProb(p)} = \frac{\text{falsePos} \times p}{\text{falsePos} \times p + (1-\text{falseNeg}) \times (1-p)}
\end{equation}

Same operators for any number of objects and locations!
Planning, replanning, and execution

Goal: $K_{Loc}(o, L_0, 0.95)$
Belief: $(0.2, 0.3, 0.5)$
Plan: Look($L_0$), Look($L_0$)
Planning, replanning, and execution

Goal: \( \text{KLoc}(o, L0, 0.95) \)
Action: Look(L0)
Observation: No see
Belief: (0.09, 0.26, 0.65)
Planning, replanning, and execution

Goal: $K_{Loc}(o, L_0, 0.95)$
Plan: Look(L2), Move(L2, L0), Look(L0)
Belief: (0.09, 0.26, 0.65)
Planning, replanning, and execution

Goal: $K_{Loc}(o, L0, 0.95)$
Action: Look(L2)
Observation: No see
Belief: (0.18, 0.53, 0.29)
Planning, replanning, and execution

Goal: \( \text{KLoc}(o, L0, 0.95) \)

Plan: Look(L1), Move(L1, L0), Look(L0)

Belief: (0.18, 0.53, 0.29)
Planning, replanning, and execution

Goal: $K_{LOC}(o, L0, 0.95)$
Action: Look(L1)
Observation: See object
Belief: (0.04, 0.9, 0.06)
Planning, replanning, and execution

Goal: $\text{KLoc}(o, L0, 0.95)$
Action: $\text{Move}(L1, L0)$
Observation: --
Belief: $(0.76, 0.18, 0.06)$
Planning, replanning, and execution

Goal: $\text{KLoc}(o, L0, 0.95)$
Action: Look(L0)
Observation: see object
Belief: (0.96, 0.03, 0.01)
PNV Regression: Transition

\[ KL_{\text{oc}}(O, L, \epsilon, \delta) \equiv \Pr(|\text{Loc}(O) - L| > \delta) < \epsilon \]

Set of distributions such that:
- after transition with mean 0.5, sigma 0.1
- PNV(2, 0.5) > 0.95
PNV Regression: Transition

\[ K_{Loc}(O, L, \epsilon, \delta) \equiv \Pr(\{|Loc(O) - L| > \delta\}) < \epsilon \]

Set of distributions such that:
- after transition with mean 0.5, sigma 0.1
- then observation with sigma 0.4
- \( \text{PNV}(2, 0.5) > 0.95 \)