„Planning and Acting in Uncertain Environments using Probabilistic Inference“
by Vermao and Rao

Helmut Kreuzer,
CI Seminar D, SS2007
The paper

D. Verma, R.P.N. Rao
Dept. of CSE, Univ. of Washington, 2006

Problem:
  planning and selecting actions in noisy environments.

Solution:
  infer actions using MPE method
Overview

- POMDP and relations to MDP
- Graphical Models
- Probabilistic Inference in graphical models
- Three Methods for inferring Actions
- Control strategies
- The POMDP algorithm
- Results
- Conclusion
POMDP and relations to MDP

- **partially observable Markov decision process**
- states not fully observable
  - e.g. sensors are imperfect, objects occlude one another, etc.
- some aspects of the states are hidden
- no one-to-one mapping between observed and hidden feature
POMDP and relations to MDP

- tinny navigation POMDP[Littman]:

- 9 observations: the surrounding walls and the star

- “north in room c” = “south in room b”
POMDP and relations to MDP

- the POMDP model:
  - at discrete time step $t$
  - the agent is in State $S_t$
  - after performing Action $A_t$ the State stochastically changes to State $S_{t+1}$
  - doesn’t see the state, but receives a observation $O_t$
  - if $O_t = S_t \rightarrow$ the POMDP reduces to a MDP (fully observed)
POMDP and relations to MDP

- POMDP is characterized by the
  - transition function $P(S_{t+1}|S_t,A_t)$, the
  - observation function $P(O_t|S_t,A_{t-1})$, and the
  - reward function $E(R_t|S_t,A_{t-1})$

- Goal: find policy $\pi$ which maps the
  $h_{1:t} = ((O_1,A_1,R_1), \ldots, (O_t,A_t,R_t))$
  into a action $A_t$
Graphical Models

- used to make inferences for actions.

- the standard model for MDP

- the POMDP model with goal node $G_t$ and "finished" node $F_t$ used in the paper
Probabilistic Inference in graphical models

wanted:

- a action sequence $A_{1:T}$ that maximizes the probability of reaching the goal

→ using MDP:
  \[ T = \text{max. episode length} \]
  high reward when reaching the goal
→ but: problems in noisy environments
Probabilistic Inference in graphical models

- using probabilistic inference for action selection:
  - infer $A_t$, given state $S_t = s$ and goal state $S_{T+1} = g$
  - encode domain-specific knowledge
  - e.g. dependence between variables or prior probabilities over states
Inferring Actions

- consider three methods:
  - marginal distribution over actions
  - maximum a posteriori (MAP) sequence
  - most probable explanation (MPE)
Inferring Actions

- marginal distribution over actions:

$$\tilde{a}_t = \arg\max_{a_t} P(A_t = a_t \mid S_t = s, S_{T+1} = g)$$

- typical approach in graphical models
- but: sub-optimal, because
  - local
  - maximize the probability of success of all possible future actions
  - no assumption about potential optimal actions after $\tilde{a}_t$
Inferring Actions

- maximum a posteriori (MAP) sequence:

$$\hat{a}_{1:T} = \arg\max_{a_{1:T}} P(a_{1:T} \mid S_1 = s, S_{T+1} = g)$$

- computes the posterior over a action sequence
- conditioned on reaching the goal within a specified number of steps
Inferring Actions

- maximum a posteriori (MAP) sequence:
  - 1: joint probabilities over all states and actions:
    \[
    p(s_{2:N+1}, a_{1:N}) = \prod_{n=2}^{N} p(s_n \mid s_{n-1}, a_{n-1}) p(a_n \mid a_{n-1})
    \]
  - 2: posterior distribution over actions via Bayes’ rule:
    \[
    p(a_{1:N} \mid s_1 = i, s_{N+1} = g)
    \]
Inferring Actions

- maximum a posteriori (MAP) sequence:
  - describe an optimal policy, optimal plan
  - computing exact MAP sequence is NP-complete
  - approximation exists, [Attias] but no exact solution
Inferring Actions

- most probable explanation (MPE):

\[
\bar{a}_{1:T}, \bar{s}_{2:T} = \arg\max P(a_{1:T}, s_{2:T} \mid S_1 = s, S_{T+1} = g)
\]

- inspired by the observation, that humans often visualize a specific action sequence when they plan, instead of optimum over all possible outcomes
Inferring Actions

- most probable explanation (MPE):
  - can be solved efficiently using standard techniques:
    - junction tree algorithm
      - included in the BNT toolbox
      - 2 steps: build tree (clustering, cliques) message passing
      - \( O(|C|)^2 \) if tree is build and consistent
Control strategies

1: Plan and Execute

- execute the total computed plan $a_{1:T}$ (MPE)
- calculate a new one if goal is not reached
Control strategies

2: Greedy/Local strategy

- execute the local marginal action $\tilde{a}_t$ at each time step
Control strategies

3: Execute and Verify

- execute the MPE sequence and compare at each time step the new state with the predicted state.
- if they do not match, recompute.
Control strategies

- Plan-and-Execute is a poor strategy for noisy environments
  - need exponentially more time
  - closed-loop strategies are better
POMDP algorithm

- if the robot has no access to the true state
  - problem becomes a POMDP problem
- modified MPE algorithm
  - compute the plan $a_{1:T}$ based on the observation $o_{1:t}$

$$
\bar{a}_{t:T}, \bar{o}_{t+1:T+1}, \bar{s}_{1:T} = \\
\arg\max P(a_{t:T}, o_{t+1:T+1}, s_{1:T} \mid o_{1:t}, a_{1:t-1}, S_{T+1} = g)
$$
POMDP algorithm

- in unknown environments:
  - select random start and goal states
  - infer MPE plan and execute it
  - update $\mathcal{T}_{s'sa}$ (frequency counts)
  - update the policy only if the goal is reached
  - use Exploration/Exploitation and decay
POMDP algorithm

- if the environment model is known:
  - given: initial observation $o_1$, $g$, $T$
  - compute MPE plan
  - execute $a_t$ and observe $o_{t+1}$
  - if expected observation $\neq o_{t+1}$
    $\Rightarrow$ update MPE plan
Results

- evaluate Algorithm in Hallway environment [Littman]
  - 89 States, 17 observations, 5 actions
Results

- Percentage of times goal was reached starting random:

<table>
<thead>
<tr>
<th>Domain</th>
<th>Q-MDP</th>
<th>PBVI</th>
<th>HSVI</th>
<th>MPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hallway</td>
<td>47.4</td>
<td>96</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Hallway2</td>
<td>25.9</td>
<td>98</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

Comparison with other POMDP algorithms (T=251)
Conclusion

- new approach for planning and acting in uncertainty
- use MPE instead of MAP, therefore more complex models are possible
- beat or match the performance of advanced POMDP solvers
Future work

- real-time-performance
  - Matlab BNT Toolbox not optimized for large networks
- problem with large number of states
  - hierarchical extensions
- exploring algorithms for continuous states and action spaces
  - controlling a robotic arm
  - balance a humanoid robot
Thank you for your attention!