Approximate Inference for Planning in Stochastic Relational Worlds

ICML 2009
Stochastic Relational Worlds

Simulator example
The Problem

- **Goal**: control an autonomous agent in an unknown environment for varying goals

- **Model-based approach**: learn a world model $P(s' \mid a, s)$ and use this model to plan actions

- **Requirements for world models**:
  - Noise
  - Stochastic action effects
  - Generalize to new situations
  - Learned from experience

- **Requirements for planning**:
  - Fast
  - Robust
  - Varying goals

We employ noisy probabilistic relational rules.

Novel planning approach
Background: Representation

- Symbolic relational representation
- States

\[
\begin{align*}
on(o_1, o_2) \\
on(o_2, \text{table}) \\
on(o_3, \text{table}) \\
inhand(o_4) \\
size(o_3) = \text{big}
\end{align*}
\]

- Actions

\[
\begin{align*}
grab(o_4) \\
puton(o_1)
\end{align*}
\]
Background: Relational rules

Noisy indeterministic deictic rules (Pasula, Zettlemoyer and Kaelbling, 2007)

- **Action**: $\text{grab}(X)$
  - **Context**: $\text{on}(X, Y)$, $\text{block}(Y)$, $\text{table}(Z)$

- **Outcomes**:
  - $0.7$ : $\text{inhand}(X)$, $\neg \text{on}(X, Y)$
  - $0.2$ : $\text{on}(X, Z)$, $\neg \text{on}(X, Y)$
  - $0.1$ : noise

- **Relational factorized deictic reference**

- **Indeterminism**

- **No efficient planning method**

- **Noise outcome**

- **Effective learning algorithm**
Background: SST Planning

- Existing method for planning with NID rules: sparse sampling trees (SST) planning (Kearns et al., 2002)
  - Near optimal, but highly inefficient.
  - Planning horizon $d$
  - Branching factor $b$

Leaves at horizon $d$: $(ba)^d$

$a = 10$, $d = 4$, $b = 4 \rightarrow \sim 2.5 Mio.$
Our planning approach

- **PRADA**: probabilistic relational action-sampling in dynamic Bayesian networks planning algorithm
- Plan in relational worlds by means of inference
- We sample action sequences and infer posteriors over hidden state variables.

1. Convert NID rules to **dynamic Bayesian networks** (DBNs)

2. Approximate **inference** algorithm to predict effects of action sequences

3. **Informed sampling** strategy for action sequences
For rule-set $\Gamma$ and set of objects $O$, ground all rules:

\[
\begin{align*}
\text{grab}(X) &: \quad \text{on}(X, Y), \ \text{block}(Y), \ \text{table}(Z) \\
&\quad \quad \quad \rightarrow \quad \begin{cases} 
0.7 & : \ \text{inhand}(X), \ \neg \text{on}(X, Y) \\
0.2 & : \ \text{on}(X, Z), \ \neg \text{on}(X, Y) \\
0.1 & : \ \text{noise}
\end{cases}
\end{align*}
\]

$O = \{o_1, o_2, o_3, \ldots \}$

\[
\begin{align*}
\Gamma(O)_1 &: \quad X \rightarrow o_1, \ Y \rightarrow o_2, \ \text{Z} \rightarrow o_3 \\
\text{grab}(o_1) &: \quad \text{on}(o_1, o_2), \ \text{block}(o_2), \ \text{table}(o_3) \\
&\quad \quad \quad \rightarrow \quad \begin{cases} 
0.7 & : \ \text{inhand}(o_1), \ \neg \text{on}(o_1, o_2) \\
0.2 & : \ \text{on}(o_1, o_3), \ \neg \text{on}(o_1, o_2) \\
0.1 & : \ \text{noise}
\end{cases}
\end{align*}
\]

\[
\begin{align*}
\Gamma(O)_2 &: \quad X \rightarrow o_1, \ Y \rightarrow o_3, \ \text{Z} \rightarrow o_2 \\
\text{grab}(o_1) &: \quad \text{on}(o_1, o_3), \ \text{block}(o_3), \ \text{table}(o_2) \\
&\quad \quad \quad \rightarrow \quad \begin{cases} 
0.7 & : \ \text{inhand}(o_1), \ \neg \text{on}(o_1, o_3) \\
0.2 & : \ \text{on}(o_1, o_2), \ \neg \text{on}(o_1, o_3) \\
0.1 & : \ \text{noise}
\end{cases}
\end{align*}
\]

\[
\text{etc.}
\]
Convert NID rules to DBNs

DBN model for $K$ ground rules

- Predecessor state
- Successor state
- Contexts
- Outcome
- Rule
- Action
- Reward

$S_1$, $S_N$ are states.
$O$ is the action.
$R$ is the reward.
$\Phi_1, \Phi_K$ are rules.
$S'$, $S'_N$ are successor states.
$U$, $U'$ are reward states.
Approximate Inference

- **Exact inference** is intractable in our graphical model.

- Idea of the **factored frontier** algorithm (Murphy & Weiss, 2001): approximate belief with a product of marginals

\[
P(s^t \mid a^{0:t-1}) \approx \prod_i P(s^t_i \mid a^{0:t-1})
\]

- Based on this approximation, we derive a **filter method** to **propagate action effects** forward:

\[
P(s^t \mid a^{0:t-1}) \times a^t \rightarrow P(s^{t+1} \mid a^{0:t})
\]
Approximate Inference

Let $\alpha(s_i^t) := P(s_i^t \mid a^{0:t-1})$ and $\alpha(s^t) := P(s^t \mid a^{0:t-1}) \approx \prod_{i=1}^{N} \alpha(s_i^t)$. 

We calculate:

\[
\alpha(s_i^{t+1}) = \sum_{r^t} P(s_i^{t+1} \mid r^t, a^{0:t-1}) P(r^t \mid a^{0:t})
\]

\[
P(s_i^{t+1} \mid r^t, a^{0:t-1}) \approx \sum_{s_i^t} P(s_i^{t+1} \mid r^t, s_i^t) \alpha(s_i^t)
\]

\[
P(R^t = r \mid a^{0:t}) = I(r \in \Gamma(a^t)) P(\Phi^t_r = 1 \mid a^{0:t-1})
\cdot P(\bigwedge_{r' \in \Gamma(a^t) \setminus \{r\}} \Phi^t_{r'} = 0 \mid \Phi^t_r = 1, a^{0:t-1})
\]

\[
P(U^t = 1 \mid a^{0:t-1}) \approx \prod_{i \in \pi(U^t)} \alpha(S_i^t = \tau_i)
\]
Informed action sequence sampling

- **Informed sampling strategy:** sample “sensible” action sequences $a^{0:T-1}$ with high probability

\[ P_{sample}^t(a) \propto \sum_{r \in \Gamma(a)} P(\phi^t_r = 1, \bigwedge_{r' \in \Gamma(a) \setminus \{r\}} \phi^t_{r'} = 0 | a^{0:t-1}) \]

- Compute **posteriors over rewards** by means of approximate inference

\[ Q(a^{0:T-1}, s^0) := \sum_{t=1}^{T} \gamma^t P(U^t = 1 | a^{0:t-1}, s^0) \]

- Choose first action of best action sequence $a^*$

- **An extension:** Adaptive PRADA

  - Can $a^*$ be further improved by deleting some actions?
Results

- 3 experiments with different planning goals
- Learn rule-sets in a world of 6 blocks
- Test worlds with different blocks and block numbers. → Generalization from training world to test worlds.

For 10 objects:
- Number of states $N_S > 2^{160}$
- Number of actions $N_A = 21$
- For planning horizon $d = 4$, number of possible action sequences: $N_A^d = 21^4 = 194481$
Results – Three specific blocks

- Build tower with three *specific* blocks.
- Can be achieved with four actions.

*PRADA* has high performance with small planning time!

*SST* either performs badly (small $b$) or is extremely slow (large $b$).
Results – Reverse tower

<table>
<thead>
<tr>
<th>Obj.</th>
<th>Planner</th>
<th>Suc.</th>
<th>Trial time (s)</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>5+1</td>
<td>SST (b=1)</td>
<td>0.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5+1</td>
<td>SST (b=2)</td>
<td>0.0</td>
<td>&gt; 1 day</td>
<td>-</td>
</tr>
<tr>
<td>5+1</td>
<td>PRADA</td>
<td>0.84</td>
<td>79.9±26.5</td>
<td>12.6±2.9</td>
</tr>
<tr>
<td>5+1</td>
<td>A-PRADA</td>
<td>0.78</td>
<td>66.3±15.6</td>
<td>10.6±1.4</td>
</tr>
<tr>
<td>6+1</td>
<td>PRADA</td>
<td>0.42</td>
<td>184.9±51.9</td>
<td>14.6±2.5</td>
</tr>
<tr>
<td>6+1</td>
<td>A-PRADA</td>
<td>0.49</td>
<td>190.4±49.8</td>
<td>12.8±1.7</td>
</tr>
<tr>
<td>7+1</td>
<td>PRADA</td>
<td>0.47</td>
<td>415.9±186.3</td>
<td>18.1±5.1</td>
</tr>
<tr>
<td>7+1</td>
<td>A-PRADA</td>
<td>0.56</td>
<td>331.6±118.3</td>
<td>14.8±1.8</td>
</tr>
</tbody>
</table>
Conclusions

Efficient planning method for probabilistic relational rules based on approximate inference.

Intelligent agent can now
- learn dynamics of complex stochastic world
- and quickly derive appropriate actions for varying goals generalizing to similar, but different worlds.
Thank you for your attention!

More information:
http://cs.tu-berlin.de/~lang/
References

