Simultaneous Learning of Structure and Value in Relational Reinforcement Learning

Scott Sanner
University of Toronto
Overview

1. Relational RL: Advantages and challenges

2. Background and related work

3. An approach to structure and value RRL (SVRRL):
   - Domain assumptions/restrictions
   - Efficiently learning value
   - Efficiently learning structure

4. Experimental results

5. Conclusions and future work
RRL: Advantages and Challenges

- RRL is a natural representation/learning paradigm:
  - Describe world as objects and relations between them
  - Compact descriptions: absence-as-negation, quantification

- But, benefits are not without drawbacks:
  - Very large state spaces: Combinatorial explosion of ground relations as domain size increases
  - Need robust learning for sparse data: Restrict hypothesis space initially, relax in presence of more data
  - Must focus on good approximations: Optimal/exact inference extremely difficult
RRL: Addressing these Challenges

- General solution difficult, focus on restricted setting:
  - Finite-horizon, undiscounted domains (assuming MDP setting)
  - Single terminal reward of success/failure
  - Applies to goal-oriented tasks (e.g. planning, games w/ stationary opp.)

- Value function $\equiv$ probability of success

- Allows us to address previous RRL challenges:
  - Very large state spaces: Repr. value function as naive Bayes net
  - Robust learning: Leverage Bayes net parameter & structure learning
  - Good approximations: Use max-likelihood (ML) and MDL principles
Background and Related Work

- **Model-free relational RL:**
  - (Dzeroski et al, 1998): Logical regression trees for RRL (top-down)
  - (Walker et al, 2004): Sample & weight relational features (bottom-up)
  - (Croonenborghs et al, 2004): SVRRL can be viewed as instance of general QLARC framework (bottom-up)

- **Bayes net structure learning:**
  - (Friedman and Goldschmidt, 1996): Tree-augmented naive Bayes (TAN) for classification; SVRRL leverages similar approach
  - (Friedman et al, 1999): Probabilistic relational model (PRM) learning; full approach too computationally intensive for SVRRL
Notational Preliminaries

- \( \{R_1, \ldots, R_i\} \): Set of relations used to describe a state
- \( \{A_1, \ldots, A_j\} \): Set of relation attribute types
  - Example: \( R_1(A_1, A_2) \), \( A_1 = \{a, b\} \), \( A_2 = \{1, 2\} \)
  - 4 ground atoms: \( \{R_1(a, 1), R_1(a, 2), R_1(b, 1), R_1(b, 2)\} \)
  - \( 2^4 \) possible truth assignments = 16 states

- \( F = \{F_1, \ldots, F_n\} \): Ground rel. atoms (boolean features)
- \( f = \{f_1, \ldots, f_p, \overline{f}_{p+1}, \ldots, \overline{f}_n\} \): Feature truth assignment
  - Order true/positive features first, false/negative features last
  - Represent state \( f \) as \( \{f_1, \ldots, f_p\} \), assume absence-as-negation
  - Space efficient because typically \( p \ll n \)
Theoretic Preliminaries

- **Under a fixed policy** $\pi$, MDP reduces to a Markov chain:

- Only non-zero reward $r$ is initial transition to success

- Value function is prob. of reaching success in $\infty$ limit:

$$V_\pi(s) = E_\pi[\sum_{t=0}^{\infty} r^t | S^{t=0} = s] = P(S^{t=\infty} = success | S^{t=0} = s)$$
Overall Learning Framework

- For each trial/time-step, record state & final outcome:

<table>
<thead>
<tr>
<th>Time-step 1</th>
<th>Time-step 2</th>
<th>Time-step k-1</th>
<th>Time-step k</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack(1), Expose(3,5)</td>
<td>Block(2,7), Expose(9,10)</td>
<td>Block(19,1), Expose(9,1)</td>
<td>Expose(24,1)</td>
<td>Failure</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trial 1</th>
<th>Trial m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block(2,3), Block(7,9)</td>
<td>Block(3,4), Attack(11,5)</td>
</tr>
<tr>
<td>Block(23,1)</td>
<td>Expose(8,1), Expose(9,1)</td>
</tr>
</tbody>
</table>

- Computational and representational issues aside:
  - Let $W$ be a boolean variable denoting eventual win/success
  - Optimal value function under a fixed policy is $P(W|F_1,\ldots,F_n)$
  - Learning = direct estimate of $P(W|F_1,\ldots,F_n)$ from trial data
Unfortunately, \( P(W|F_1, \ldots, F_n) \) is intractably large... so approximate it with a naive Bayesian network, e.g.

ML cond. prob. table (CPT) params just observed freq.

Then value of a state can be easily calculated:

\[
\hat{P}(w|f) = \frac{\hat{P}(f|w)\hat{P}(w)}{\hat{P}(f)} = \frac{\hat{P}(w) \prod_{i=1}^{p} \hat{P}(f_i|w) \prod_{i=p+1}^{n} \hat{P}(\bar{f}_i|w)}{\sum_{o \in \{w, \bar{w}\}} \hat{P}(o) \prod_{i=1}^{p} \hat{P}(f_i|o) \prod_{i=p+1}^{n} \hat{P}(\bar{f}_i|o)}
\]
Efficient Policy Evaluation

- Still many ground atoms, need to eval policy efficiently:
  - Focus on policy evaluation via after-state analysis
  - Pol. execution is just choice of best state from possible set
  - Only need relative comp., use log winning odds \( \log \left( \frac{P(w|f)}{P(\bar{w}|f)} \right) \)

\[
\log \frac{P(w|f)}{P(\bar{w}|f)} = \log \frac{P(w)}{P(\bar{w})} + \sum_{i=1}^{p} \log \frac{P(f_i|w)}{P(f_i|\bar{w})} + \sum_{i=p+1}^{n} \log \frac{P(\bar{f}_i|w)}{P(\bar{f}_i|\bar{w})}
\]

Let \( C = \log \frac{P(w)}{P(\bar{w})} + \sum_{i=1}^{n} \log \frac{P(\bar{f}_i|w)}{P(f_i|\bar{w})} \)

\[
\log \frac{P(w|f)}{P(\bar{w}|f)} = C + \sum_{i=1}^{p} \left( \log \frac{P(f_i|w)}{P(f_i|\bar{w})} - \log \frac{P(\bar{f}_i|w)}{P(\bar{f}_i|\bar{w})} \right)
\]

- Find best after-state by only looking at positive features!
Structure Learning Overview

- **Feature attribute augmentation (FAA) learning:**
  - Each CPT is a conditional probability, e.g. \( P(E(5, 3, 0) | W) \)
  - Could approximate CPT probability using attribute estimates with don’t cares “.”:
    \[
    P(E(5, .., .) | W) \cdot P(E(., 3, ..) | W) \cdot P(E(., .., 0) | W)
    \]
  - Need to determine which joint attribute estimates are most informative (ML)

- **Feature conjunction (FC) learning:**
  - Can combine nodes to come up with joint probability estimates
  - Need to determine which joint nodes are most informative (ML)
Greedy Optimal Structure Learning

- **Given two independent features** $F_a$ and $F_b$:
  - Want to determine increase in log-likelihood if features considered jointly:
    \[ \Delta l^*(\theta|D) = C + M \cdot I(F_a, F_b|W) \] (see paper for derivation)
  - In brief, change in log-likelihood due to join given by mutual conditional entropy $I(\cdot)$ times # of data samples $M$ ($C$ is a common constant)
  - Choose FAA or FC joins to maximize log-likelihood (greedy optimal)

- **Caveat: Statistical noise leads to structure overlearning**
  - Solution: Use MDL score: $MDL(B|D) = \frac{1}{2} \log(M|B|) - l^*(\theta|D)$
  - Balances log-likelihood score vs. # parameters $B$ in Bayes net

- **Why is this relational RL?**
  - FAA learning applies to all ground relations sharing learned attributes
  - Non-parametric CPT learning exploits rel. structure via similarity of attribute dimensions and sparseness of relation sampling (esp. for FC)
Empirical Results

- Evaluated FAA-SVRRL on Backgammon (est. $10^{18}$ states)
- Learning efficiency data:
  - Trains 5000 games of self-play in < 10 min on 1 GhZ PIII, 128 Mb
  - Use non-parametric CPT learning: 240 instances, < 10Kb RAM
  - FAA-SVRRL learns faster than static version starting with full structure
- Asymptotic performance evaluation data:

<table>
<thead>
<tr>
<th>Player</th>
<th>Winning PCT</th>
<th># Training Games</th>
</tr>
</thead>
<tbody>
<tr>
<td>TD-GAMMON 1-Ply, estimated</td>
<td>66.0 % ± ???</td>
<td>1,500,000</td>
</tr>
<tr>
<td>FAA-SVRRL</td>
<td>51.2 % ± 0.02</td>
<td>5,000</td>
</tr>
<tr>
<td>PUBEVAL (LINEAR REGRESSION)</td>
<td>50.0 % ± 0.00</td>
<td>UNKNOWN</td>
</tr>
<tr>
<td>HC-GAMMON (GENETIC PROG)</td>
<td>40.0 % ± 3.46</td>
<td>100,000</td>
</tr>
</tbody>
</table>
Conclusions and Future Work

● Conclusions:
  – FAA-SVRRL is efficient structure and value RRL algorithm
  – Achieves commendable performance in Backgammon

● Future work:
  – Full implementation and evaluation of FC-SVRRL
  – Experiment with domains other than Backgammon
  – Use other learning frameworks: Prob/ML vs. Winnow/COLT
  – Can we efficiently learn more complex tree-augmented naive Bayes (TAN) or PRM-style structure?