VDCBPI: an Approximate Scalable Algorithm for Large POMDPs

Pascal Poupart
University of Waterloo
ppoupart@cs.uwaterloo.ca

Craig Boutilier
University of Toronto
cebly@cs.toronto.edu

By combining Value-Directed Compression (handles large state spaces) with Bounded Policy Iteration (restricted policy search) we obtain an approximate scalable algorithm VDCBPI demonstrated on POMDPs of 33 million states.
POMDPs

• Natural framework for
  – Sequential decision making
  – Noisy sensors & imperfect actuators
  – Partially observable environments
  – Complex concurrent goals
  – Optimizing exploration/exploitation tradeoff

• Lots of applications: helicopter control [Ng], robot navigation [Roy & Gordon 02], preference elicitation [Boutilier 02], spoken dialogue systems [Peak 00], active gesture recognition [Darell 96], Bayesian reinforcement learning [Duff 00]
In a Nutshell

• Problem: lack of efficient algorithms

• Two sources of intractability
  – Complex policy spaces
  – Large state spaces

• Real-world problems often “structured”:
  – Small reachable belief region, small yet good policies
  – Conditional independence, context-specific independence, additive separability

• VDCBPI: exploit structure to mitigate both sources of intractability simultaneously
Graphical Representation

Solution: policy $\pi$ maximizes expected total rewards
Belief states

• State \( s \):
  – defined by all relevant features of the decision process

• Belief state \( b \):
  – probability distribution over states
  – summarizes all past actions and observations
    \[ b_t = <..., a_{t-3}, z_{t-2}, a_{t-2}, z_{t-1}, a_{t-1}, z_t> \]

• Belief update: (Bayes theorem)
  – let \( T^{a,z} \) be the update operator for belief states
    \[ b_t = T^{a_{t-1}, z_t}(b_{t-1}) \quad \text{or} \quad b_t = <b_{t-1}, a_{t-1}, z_t> \]
Policies

- Policy $\pi : \langle A, Z, A, Z, \ldots \rangle \rightarrow A$ or $\pi : B \rightarrow A$
  - Mapping from past experiences to actions
  - Mapping from belief states to actions

- Evaluate policy $\pi$ using value function $V^\pi : B \rightarrow \mathbb{R}$
  - Mapping from belief states to expected total return
  \[ V^\pi(b) = \sum_t \gamma^t R(b_t) \]

- Goal: find optimal policy $\pi^*$
  - Policy that maximizes expected total return
  \[ V^*(b) \geq V^\pi(b) \quad \forall \pi, b \]
Bounded Policy Iteration [NIPS-03]

- Finite state controllers (automata):
  - Simple, convenient representation of POMDP policies

- Proposal: Bounded Policy Iteration (BPI)
  - Grow size of controller slowly
    - Policy iteration [Hansen]: exponentially large controller
    - Converge to policy optimal at “tangent” belief states
      - Gradient ascent [Meuleau & al.]: local convergence
  - Efficient running time and memory usage
    - Branch & bound [Meuleau & al.]: poor running time
    - Stoch. local search [Braziunas & al.]: poor memory usage
Finite State Controller

- Nodes: actions
- Edges: observations
- Automaton: policy $\pi$
Bounded Policy Iteration

- Consider stochastic controllers
  - Node: $\Pr(a)$
  - Edges: $\Pr(n'|a,z)$

- Improve one node at a time

- No improvement possible $\rightarrow$ add new node
Node Improvement

- Uniform improvement

\[
\begin{align*}
\text{max } & \quad \varepsilon \\
\text{s.t. } & \quad V_n(s) + \varepsilon \leq \\
& \quad \sum_a \Pr(a) \left[ R^a(s) + \gamma \sum_{z,s'} \Pr(s'|s,a,z) \sum_{n'} \Pr(n'|a,z) V_{n'}(s') \right]
\end{align*}
\]

- Biased search
  - Occupancy distribution: \( o_n(s) \propto \Pr(s,n) \)

\[
\begin{align*}
\text{max } & \quad \sum_s o_n(s) \varepsilon(s) \\
\text{s.t. } & \quad V_n(s) + \varepsilon(s) \leq \\
& \quad \sum_a \Pr(a) \left[ R^a(s) + \gamma \sum_{z,s'} \Pr(s'|s,a,z) \sum_{n'} \Pr(n'|a,z) V_{n'}(s') \right]
\end{align*}
\]
Scalability

- Tag-avoid [Pineau & al.]
  - 870 states, 30 obs, 5 acts
## Benchmark comparison

<table>
<thead>
<tr>
<th>Problems</th>
<th>Algorithms</th>
<th>Exp. Reward</th>
<th>Sol. Size</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tiger-grid</strong></td>
<td><strong>BPI</strong></td>
<td>2.22</td>
<td>120</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>Perseus</td>
<td>2.34</td>
<td>134</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>PBVI</td>
<td>2.25</td>
<td>470</td>
<td>3448</td>
</tr>
<tr>
<td></td>
<td>PBUA</td>
<td>2.30</td>
<td>660</td>
<td>12116</td>
</tr>
<tr>
<td><strong>Hallway</strong></td>
<td><strong>BPI</strong></td>
<td>0.51</td>
<td>43</td>
<td>185</td>
</tr>
<tr>
<td></td>
<td>Perseus</td>
<td>0.51</td>
<td>55</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>PBVI</td>
<td>0.53</td>
<td>86</td>
<td>288</td>
</tr>
<tr>
<td></td>
<td>PBUA</td>
<td>0.53</td>
<td>300</td>
<td>450</td>
</tr>
<tr>
<td><strong>Hallway2</strong></td>
<td><strong>BPI</strong></td>
<td>0.32</td>
<td>60</td>
<td>790</td>
</tr>
<tr>
<td></td>
<td>Perseus</td>
<td>0.35</td>
<td>56</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>PBVI</td>
<td>0.34</td>
<td>95</td>
<td>360</td>
</tr>
<tr>
<td></td>
<td>PBUA</td>
<td>0.35</td>
<td>1840</td>
<td>27898</td>
</tr>
<tr>
<td><strong>Tag-avoid</strong></td>
<td><strong>BPI</strong></td>
<td>-6.65</td>
<td>17</td>
<td>250</td>
</tr>
<tr>
<td></td>
<td>Perseus</td>
<td>-6.17</td>
<td>280</td>
<td>1670</td>
</tr>
<tr>
<td></td>
<td>PBVI</td>
<td>-9.18</td>
<td>13340</td>
<td>180880</td>
</tr>
<tr>
<td></td>
<td>BBSLS</td>
<td>-8.20</td>
<td>30</td>
<td>100000</td>
</tr>
</tbody>
</table>
Value-Directed Compressions [NIPS-02]

• Recall: belief state $b$ is
  – summary of all information available
  – sufficient statistic for predicting future belief states

• For planning purposes
  – only care to distinguish value $V^\pi$ for each policy $\pi$
  – value $V^\pi$ is the sum of future rewards
  – retain only information relevant to prediction of future rewards in compressed belief state $\tilde{b}$
Sufficient Conditions

- Functional flow
  - Retain only information necessary to predict future rewards

- Sufficient conditions:

\[
R = \tilde{R} \circ f
\]

\[
f \circ T^{a,z} = \tilde{T}^{a,z} \circ f \quad \forall a,z
\]
Compressed POMDP

- Original POMDP: $< B, T^{a,z}, R >$
- Compressed POMDP: $< \tilde{B}, \tilde{T}^{a,z}, \tilde{R} >$

**Theorem:** the original and compressed POMDPs are equivalent.

If $R = \tilde{R} \circ f$ and $f \circ T^{a,z} = \tilde{T}^{a,z} \circ f \ \forall a,z$

Then $V^\pi = \tilde{V}^\pi \circ f \ \forall \pi$
Lossless Linear Compressions

- **Best lossless compression:**
  - Krylov Iteration
    \[ F = [R, T_{a_1,z_1}R, T_{a_1,z_2}R, T_{a_2,z_1}R, T_{a_2,z_2}R, (T_{a_1,z_1})^2R, \ldots] \]
  - Solve: \( R = FR \) and \( T^{a,z}F = FT^{a,z} \)

| Problems       | |A| | |O| | |S| | lossless |
|----------------|---|---|---|---|---|---|---|---|
| Coffee         | 2 | 3 | 32 | 7 |
| Spoken-dialog  | 16| 37|433 |31 |
| Cycle7         | 15| 2 |128 |128 |
| 3legs7         | 15| 2 |128 |128 |
Lossy Linear Compressions

- Good *lossy* compression:
  - Truncated Krylov Iteration
  - Solve: $F^T R = F^T F \tilde{R}$ and $F^T T_{a,z} F = F^T F \tilde{T}_{a,z}$

Cycle7

Spoken-dialog
Compressed Bounded Policy Iteration

1st: compress POMDP with VDC
   - Structure exploited: additive separability, conditional independence, context-specific independence
   - Advantage: VDC can be easily integrated with any solution algorithm

2nd: solve POMDP with BPI
   - Structure exploited: small reachable belief region, good policies of low complexity
   - Advantages: robustness to local optima, scalability, no belief state monitoring required
Integrating VDC with BPI

• F must be non-negative
  – Since $V = F\tilde{V}$, maximizing $\tilde{V}$ also maximizes $V$

• Normalize the columns of F
  – Let $F = [F_1 \ F_2]$
  – Suppose $V = F\tilde{V} = \tilde{v}_1 \ F_1 + \tilde{v}_2 \ F_2$
  – $\tilde{v}_1$ has impact $||F_1||$ and $\tilde{v}_2$ has impact $||F_2||$
Integrating VDC with BPI

• Evaluate policies iteratively
  – Eigenvalues of $T_{a,z}$ between -1 and 1
  – Eigenvalues of $\tilde{T}_{a,z}$ unconstrained
  – $\tilde{\mathcal{V}}(\tilde{b}) = \sum_n \gamma^n \tilde{R}(\tilde{T}_n(\tilde{b}))$

• When finding witness belief states
  – $B$ is a simplex
  – $\tilde{B}$ is convex hull of the rows of $F$
Synthetic Network Management

- 3legs25: 33,554,432 states, 51 actions, 2 obs.
Compressed Point-based Value Iteration

1\textsuperscript{st}: compress POMDP with VDC
   - Structure exploited: additive separability, conditional independence, context-specific independence
   - Advantage: VDC can be easily integrated with any solution algorithm

2\textsuperscript{nd}: solve POMDP with Perseus [Spaan& al.]
   - Structure exploited: small reachable belief region, good policies of low complexity
   - Advantages: very fast, easy to implement
Symbolic Point-based Value Iteration

• Use ADDs as a compact symbolic data structure
  – Structure exploited: additive separability, conditional independence
  – Advantage: policy-directed state abstraction

• Solve POMDP directly with Perseus [Spaan & al.]
  – Structure exploited: small reachable belief region, good policies of low complexity
  – Advantages: very fast, easy to implement
Synthetic Network Management

- 3legs25: 33,554,432 states, 51 actions, 2 obs.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Expected rewards</th>
<th>Solution size</th>
<th>Time compression</th>
<th>solution</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPI</td>
<td>fail</td>
<td>fail</td>
<td>fail</td>
<td>fail</td>
<td>fail</td>
</tr>
<tr>
<td>BPI+VDC</td>
<td>164.8</td>
<td>123</td>
<td>7097</td>
<td>6596</td>
<td>13693</td>
</tr>
<tr>
<td>Perseus</td>
<td>fail</td>
<td>fail</td>
<td>fail</td>
<td>fail</td>
<td>fail</td>
</tr>
<tr>
<td>Perseus+VDC</td>
<td>162.6</td>
<td>33</td>
<td>7097</td>
<td>211</td>
<td>7308</td>
</tr>
<tr>
<td>Perseus+ADD</td>
<td>fail</td>
<td>fail</td>
<td>fail</td>
<td>fail</td>
<td>fail</td>
</tr>
<tr>
<td>Heuristic</td>
<td>152.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DoNothing</td>
<td>147.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
**Assistive Technology**

- COACH: 33,454,080 states, 23 actions, 12 obs.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Expected rewards</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>compression</td>
</tr>
<tr>
<td>Perseus+ADD</td>
<td>39.1</td>
<td>0</td>
</tr>
<tr>
<td>MDPVI+ADD</td>
<td>37.6</td>
<td>0</td>
</tr>
<tr>
<td>CallCaregiver</td>
<td>25.0</td>
<td>0</td>
</tr>
<tr>
<td>BPI+VDC</td>
<td>18.1</td>
<td>75168</td>
</tr>
<tr>
<td>Perseus+VDC</td>
<td>17.8</td>
<td>75168</td>
</tr>
<tr>
<td>DoNothing</td>
<td>13.2</td>
<td>0</td>
</tr>
</tbody>
</table>
Conclusion

• Contributions
  – Improved BPI by tailoring search to reachable belief region
  – Lossy VDC by truncated Krylov iteration
  – Mitigate large state spaces and complex policy/value function simultaneously
  – Combine: VDC+BPI, VDC+Perseus, ADD+Perseus
    • Solve some POMDPs of 33 million states

• Future work
  – POMDPs with large action and observation spaces
  – Continuous POMDPs