On the Impact of Prior Knowledge on Autonomous Agents

Benjamin Rosman

Mobile Intelligent Autonomous Systems
Council for Scientific and Industrial Research
&
School of Computer Science and Applied Maths
University of the Witwatersrand
South Africa

8 September 2015
Long-Lived Agents

- Agents deployed in some environment over a long duration
  - Multiple tasks
  - Changing environment
- Continuously learn and adapt
  - Growing task, behaviour sets
- How to maintain knowledge?
  - Behaviour transfer
  - Generalisation
Transfer Learning

1. How can an agent **generalise from previous behaviours** to solve new tasks in the same environment quicker and with less risk?
   1. Accelerate policy learning
   2. Model of external agent behaviour

2. Given a set of previously learnt behaviours, what is the optimal way to **select the best one to be re-used** in a new environment or interaction?
What is reinforcement learning?

- How to learn behaviours under stochasticity and uncertainty?
  - Unsupervised?
  - Supervised?
  - Something else entirely...

Hardware and low level control by SRC
High level control & programming by ATR

CSIR
our future through science
Operating in an environment

- Rewards as a weak, delayed learning signal
  - Goal-directed learning
- Learn from repeated interaction
- Learn to **map situations to actions** so as to **maximise numerical reward** (which may be delayed)

```
<table>
<thead>
<tr>
<th>Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>state ( s_{t+1} )</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>reward ( r_t )</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>action ( a_t )</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Environment</td>
</tr>
</tbody>
</table>
```
Markov Decision Processes (MDPs)

- Model a decision problem
- $M = \langle S, A, T, R, \gamma \rangle$
- Observable
- Markov
- Policy $\pi$
Examples
Value functions

• Value of a state:
  – Expected return starting from that state and following a particular policy
  – $V^\pi(s) = E_\pi\{R_t|s_t = s\}$
    
    $= E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s\right\}$

• Value of an action in a state:
  – Expected return of starting in that state, taking that action, and then following a particular policy
  – $Q^\pi(s, a) = E_\pi\{R_t|s_t = s, a_t = a\}$
    
    $= E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s, a_t = a\right\}$
Why value functions?

• Optimal value functions:
  – $V^*(s) = \max_{\pi} V^\pi(s)$
  – $Q^*(s, a) = \max_{\pi} Q^\pi(s, a)$
  – These are the value functions given by the optimal policy $\pi^*$

• Any policy that is greedy w.r.t $V^*$ (or $Q^*$) is optimal
  – So, $\pi^*(s) = \arg\max_{a \in A} Q^*(s, a)$
Example

- Random policy:

- Optimal:

(a) gridworld

(b) \( V_\ast \)

(c) \( \pi_\ast \)
• RL learning is **trial-and-error learning** to find a good policy from experience
• So as not to solve a large system of value function equations

\[
V^\pi(s) = \max_a E \left\{ r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_t = s, a_t = a \right\}
\]

\[
= \max_a \sum_{s'} P^a_{ss'} \left[ R^a_{ss'} + \gamma V^\pi(s') \right].
\]

– Which aren’t even known!

• Exploration vs exploitation
• Model free vs model based algorithms
Q-Learning

- Initialise $Q(s, a)$ arbitrarily
- Repeat (for each episode):
  - Initialise $s$
  - Repeat (for each step of episode):
    - Choose $a$ from $s$ using $\epsilon$-greedy policy from $Q$
      
      \[
      a \leftarrow \begin{cases} 
      \arg\max_a Q(s, a) & \text{w. p. } \epsilon \\
      \text{random} & \text{w. p. } 1 - \epsilon 
      \end{cases}
      \]
    - Take action $a$, observe $r, s'$
    - Update $Q$
      
      \[
      Q(s, a) \leftarrow Q(s, a) + \alpha \left[ r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]
      \]
    - $s \leftarrow s'$
  - Until $s$ is terminal
Chapter 1:
Safe Behaviour Generalisation
(Action Priors)
Learning Domain Knowledge

• Agent performing multiple tasks in the same environment
  – Improve over time, across tasks
• Lifelong learning: what to learn over an agent’s lifetime?
  – Task independent regularities (structure) in the domain
  – Structure: general “common sense” behaviours
An Intuition

- Although many actions may be possible in some context, **only a small number are typically useful**
The Benefits of Multiple Tasks

- Set of RL policies
  - Each is specific knowledge
- Together: tease out the general domain knowledge

\[ \pi_i \]
Learning Domain Knowledge

• Tasks are drawn from a domain
  – Differ in goal: reward R
  – (In general: states S, transitions T)

• Learn model of behavioural invariances across domain
  – Task independent
  – From optimal policies

• Model: Context based distributions over action set
  – Action “usefulness” = reasonable behaviour choices
  – Condition on state (observations \( \varphi(s) \))
An Illustration

(a) Goal 1

(b) Goal 1
A Model of Domain Knowledge

• Action priors $\theta_{\varphi(s)}(A)$ [Rosman and Ramamoorthy, 2012, 2015]
  – Dirichlet distribution over $A$
  – Conditioned on $\varphi(s)$

• $\theta_{\varphi(s)}(A) \sim \text{Dir}(\alpha_{\varphi(s)}(A))$
  – Parameters $\alpha_{\varphi(s)}(A)$
A Model of Domain Knowledge

• Notion of “action usefulness”
• Formally:
  – For each policy $\pi$, define a weight $w(\pi)$ Measure of confidence/skill
  – Action utility under a policy:
    – $U^\pi_{\varphi(s)}(a) = \delta(\pi(\varphi(s), a), \max_{a' \in A} \pi(\varphi(s), a'))$
  – Action utility under a policy set:
    – $\alpha_{\varphi(s)}(a) = \sum_{\pi \in \Pi} w(\pi) U^\pi_{\varphi(s)}(a) + \alpha^0_{\varphi(s)}(a)$

Utility = 1 iff action optimal

Weighted sum of action utilities

Hyperprior
A Model of Domain Knowledge

- **Online update:**
  - Counts for each $\varphi(s)$
  - For each policy $\pi$, define a weight $w(\pi)$  
    Measure of confidence/skill

\[
\alpha_{\varphi(0)}^0(a) \leftarrow \alpha_{\varphi(a)}, \quad \forall \varphi, a
\]

Initialise to some hyperprior

\[
\begin{align*}
\alpha_{\varphi(s)}^{t+1}(a) & \leftarrow \begin{cases} 
\alpha_{\varphi(s)}^t(a) + w(\pi^t), & \pi^t(s, a) = \max_{a'} \pi^t(s, a') \\
\alpha_{\varphi(s)}^t(a), & \text{otherwise}
\end{cases}
\end{align*}
\]

Given a new policy $\pi^t$, update counts of optimal actions
Example Priors
How to Use? Guided Exploration

- **Action selection:**
  \[ \theta_{\varphi(s)}(A) \sim Dir(\alpha_{\varphi(s)}) \]
  \[ a \sim \theta_{\varphi(s)}(A) \]

- **Exploration in Q-learning (a twist on \( \epsilon \)-greedy):**
  \[
  a \leftarrow \begin{cases} 
  \arg \max_a Q(s, a), & \text{w.p. } 1 - \epsilon \\
  a \in A, & \text{w.p. } \epsilon \theta_{\varphi(s)}(a)
  \end{cases}
  \]

  Let action prior bias exploration

  Note: standard Q-learning uses uniform priors
Example: The Factory Domain

- The factory domain
  - Extended navigation domain
  - Task: procure and assemble a list of items
  - Assembly/procurement points, express route
  - Actions: $N, E, S, W, Procure, Assemble$
The Effect of Priors

- Uniform priors
- State action priors
- Hand coded expert priors
- Hand coded incorrect priors
Learning Across Multiple Tasks

- Assemble 1 item
- Assemble 4 items
The Factory Domain 2.0

- The extended factory domain
  - Each instance different
    - Assembly, procurement regions
    - Semi-random structure
    - States are not useful for transfer!
Results: Effect of Different Features

• Different feature sets in action prior

\( \phi_1 \): current cell, item status
\( \phi_2 \): cells N, S, E, W
\( \phi_3 \): \( \phi_1 \cup \phi_2 \)
\( \phi_4 \): \( \phi_3 \cup \{NE, NW, SE, SW\} \)

• Trade-off:
  – Under- vs over-representation
  – Feature learning

[Rosman 2014]
An Application: Autonomous Caregivers

Goal
Autonomous agent: enable novice agents to safely learn in a self-directed manner

[Rosman, Hayes, Scassellati, 2015]
Caregivers Perform Risk Mitigation

Approach
Adapt the environment to promote safety

– Without sacrificing quality of learning experience
– Assist through indirect communication and manipulation
Model of Novice Behaviour

- Novice modelled as an MDP
  - Environment states $s \in S$
  - Actions $a \in A$
  - Environment dynamics $T(s, a, s')$
  - Rewards $R(s, a)$
  - Policy $\pi(s, a)$
Caregivers Are A Shaping Mechanism!

- Corrective signals are provided by caregivers
  - Informed by an internal model of reasonable behaviour to assess how risk prone a novice agent is
  - Provided selectively (when necessary)
  - Provided with foresight (before harm is inevitable)
A Model of Safety

- **Goal:** Determine if novice is behaving safely
  - Estimate policy similarities between novice and expert
- **Current trajectory:** \( \tau = s^{t+1}, a^t, s^t, a^{t-1}, s^{t-1}, ... \)
- **Safe behaviour:** expert policies \( \theta_s(A) \)

\[
P(safe | \tau) = \frac{P(\tau|safe)P(safe)}{P(\tau)}
\]

\[
P(\tau | safe) = \prod_{k=1}^{t} \theta_{sk}(a^k)
\]

Prior prob. of behaving safely

Normalisation factor

Safe (reasonable) transition probs.
A Model of Danger

• **Goal: Estimate potential future dangers**
  – Expected environmental harm of likely future actions
• **Evaluate expectation for each potential source of harm** $o$:

$$P(\text{collision} \mid \tau) \times d_o$$

$$= (1 - P(\text{safe} \mid \tau)) \times P(\text{reach}_o \mid \tau) \times d_o$$

- Prob. of behaving safely
- Extrinsic damage caused by collision with $o$
- Prob. of reaching $o$ proportional to distance to $o$ from current position
Toybox World

- Exploring to reach toy boxes [ICDL 2015]
- Hazards:
  - Major damage: candles, stairs
  - Minor damage: tables
- Novice agent:
  - $\varepsilon$-greedy
  - ‘Play’ for 200 time steps
- Caregiver agent:
  - Trained on 1,000 expert steps
  - Moves 3x faster than novice
- Interventions:
  - Move candle between tables
  - Block stairwell
Results: Reducing Harm

<table>
<thead>
<tr>
<th>Agent</th>
<th>No caregiver</th>
<th>With caregiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>random motion</td>
<td>0.7346</td>
<td>0.2291</td>
</tr>
<tr>
<td>75% random</td>
<td>0.5295</td>
<td>0.1438</td>
</tr>
<tr>
<td>50% random</td>
<td>0.2846</td>
<td>0.1106</td>
</tr>
<tr>
<td>25% random</td>
<td>0.1887</td>
<td>0.0466</td>
</tr>
<tr>
<td>5% random</td>
<td>0.0565</td>
<td>0.0072</td>
</tr>
</tbody>
</table>
Results: Exploration Time

<table>
<thead>
<tr>
<th>Agent</th>
<th>No caregiver</th>
<th>With caregiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>random motion</td>
<td>35.50</td>
<td>97.05</td>
</tr>
<tr>
<td>75% random</td>
<td>41.05</td>
<td>164.20</td>
</tr>
<tr>
<td>50% random</td>
<td>89.50</td>
<td>171.50</td>
</tr>
<tr>
<td>25% random</td>
<td>107.70</td>
<td>200.00</td>
</tr>
<tr>
<td>5% random</td>
<td>195.75</td>
<td>200.00</td>
</tr>
</tbody>
</table>
Results: Environment Coverage

<table>
<thead>
<tr>
<th>Agent</th>
<th>No caregiver</th>
<th>With caregiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>random motion</td>
<td>14.6000</td>
<td>24.4500</td>
</tr>
<tr>
<td>75% random</td>
<td>16.8500</td>
<td>36.4500</td>
</tr>
<tr>
<td>50% random</td>
<td>29.7500</td>
<td>38.4500</td>
</tr>
<tr>
<td>25% random</td>
<td>34.1000</td>
<td>38.7000</td>
</tr>
<tr>
<td>5% random</td>
<td>34.7000</td>
<td>33.2000</td>
</tr>
</tbody>
</table>
Conclusion

• Action priors
  – Behavioural domain invariances
  – Task independent
  – “Common sense” knowledge
• Improve learning speed
  – Use as exploration bias in RL
• Identify safe/normal behaviour
• General paradigm for multi-task decision making agents
  – If learning multiple tasks in the same domain, learn from previous tasks!
Chapter 2: Efficient Skill Selection (Bayesian Policy Reuse)
Responding Online to New Situations

• Engaged in a task
  – Not enough time to learn a policy
• Previous experience of tasks
  – Choose the best policy in a sequence of interactions
  – Based on some latent variable
The Policy Reuse Problem

• Given:
  – Exposure to previous task instances
  – A policy library trained on those tasks
• Experience a new task
• Goal:
  – Select policies for new task to minimise total regret
• Assume: limited task duration
  – Cannot learn from scratch
Insight

(unknown embedding)
Bayesian Policy Reuse Overview
Ingredient 1: Performance

- Performance $U$:
  - Returns achieved by a policy on a task
- Performance models:
  - $P(U|\tau, \pi)$
  - Maintain for each experienced task and policy
- Use to estimate performance of a policy on an unknown task
Ingredient 2: Signals

- Signals $\sigma$: information correlated with task performance, provided during task execution
  - E.g. rewards, (partial) states
- Signal/observation models:
  - $P(\sigma|\tau, \pi)$
  - Maintain for each task and policy
- Use as feedback signal for identifying task
Belief Models

- Maintain belief over set of task instances $\tau$
- Update
  - Based on signals after playing a policy
  - Over ALL known tasks!
  - Notion of task similarity

Signal model

$$\beta^t(\tau) = \frac{P(\sigma^t | \tau, \pi^t) \beta^{t-1}(\tau)}{\sum_{\tau' \in \mathcal{T}} P(\sigma^t | \tau', \pi^t) \beta^{t-1}(\tau')}$$
Bayesian Policy Reuse

1. Select policy
2. Apply policy
3. Observe signal
4. Update belief

Algorithm 1 Bayesian Policy Reuse (BPR)

Require: Problem space $\mathcal{X}$, Policy library $\Pi$, observation space $\Sigma$, prior over the problem space $P(\mathcal{X})$, observation model $P(\Sigma|\mathcal{X}, \Pi)$, performance model $P(U|\mathcal{X}, \Pi)$, number of episodes $K$.

1: Initialise beliefs: $\beta^0(\mathcal{X}) \leftarrow P(\mathcal{X})$.
2: for episodes $t = 1 \ldots K$ do
3: Select a policy $\pi^t \in \Pi$ using the current belief $\beta^{t-1}$ and the performance model $P(U|\mathcal{X}, \pi^t)$.
4: Apply $\pi^t$ on the task instance.
5: Obtain an observation signal $\sigma^t$ from the environment.
6: Update the belief $\beta^t(\mathcal{X}) \propto P(\sigma^t|\mathcal{X}, \pi^t) \beta^{t-1}(\mathcal{X})$.
7: end for
Policy Selection

• Selection heuristics (based on Bayesian optimisation):

• Probability of Improvement (PI):

\[ \hat{\pi} = \arg \max_{\pi \in \Pi} \sum_{\tau \in \mathcal{T}} \beta(\tau) P(U^+ | \tau, \pi) \]

• Expected Improvement (EI):

\[ \hat{\pi} = \arg \max_{\pi \in \Pi} \int_{U}^{U_{\max}} \sum_{\tau \in \mathcal{T}} \beta(\tau) P(U^+ | \tau, \pi) dU^+ \]
## Illustrative Example – The Golf Range

### Ground truth:

<table>
<thead>
<tr>
<th>Club</th>
<th>Average Yardage</th>
<th>Standard Deviation of Yardage</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi_1 = 3$-wood</td>
<td>215</td>
<td>8.0</td>
</tr>
<tr>
<td>$\pi_2 = 3$-iron</td>
<td>180</td>
<td>7.2</td>
</tr>
<tr>
<td>$\pi_3 = 6$-iron</td>
<td>150</td>
<td>6.0</td>
</tr>
<tr>
<td>$\pi_4 = 9$-iron</td>
<td>115</td>
<td>4.4</td>
</tr>
</tbody>
</table>

![Diagram showing under shooting and over shooting](image-url)
Illustrative Example – Signal Models

- club #1: 215 yds
- club #2: 180 yds
- club #3: 150 yds
- club #4: 115 yds
Results on New Task

<table>
<thead>
<tr>
<th>Shot</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Club</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$\beta$ entropy</td>
<td>1.3863</td>
<td>0.2237</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

![Graph](image1.png)

![Graph](image2.png)
Surveillance Domain

- Watching for intruders, from hills
  - Connected visibility
- 68 tasks
Rapid Identification

Bayesian optimisation

Bandits (with cheating)
Library Size-Episodes-Regret Trade-off
Non-stationarity and Adversity

- Changing opponents:
  - Keep all beliefs non-zero
- New strategies:
  - Unlikely reward sequence
  - Enable learning

[Hernandez-Leal, Taylor, Rosman, *submitted*]
Multi-agents: Tracking Changes
Summary

• Bayesian Policy Reuse: general framework for rapid policy selection
  – Maintain beliefs over tasks
  – Update with observation models
  – Select according to performance models
• Interact efficiently with unknown tasks and agents
Future Work

• Extensions:
  – Continuous action/task sets
    • Distributions over parameter space
  – Different decision making paradigms
    • Classical planning
    • POMDPs
    • MCTS
Future Work

- Structure in task space?
  - Non-parametric:
    - Clustering MDPs
  - Parametric:
    - Hidden parameter MDPs
  - Compositionality and hierarchy of behaviours

[Mahmud, Hawasly, Rosman, Ramamoorthy, under review]
Thank you!

And thanks to all these great people:

Dr Subramanian Ramamoorthy (U. of Edinburgh)
Dr Majd Hawasly (U. of Edinburgh)
Dr Hassan Mahmud (U. of Edinburgh)
Bradley Hayes (Yale University)
Pablo Hernandez-Leal (INAOE)
Prof George Konidaris (Duke University)
Prof Brian Scassellati (Yale University)
Prof Matt Taylor (Washington State University)

Benjamin Rosman (brosman@csir.co.za)
Action Priors: Feature Entropy

Effect of removing a feature:

**Feature Set $\phi_1$**
- Current: 0.0156
- Items: 0.008

**Feature Set $\phi_2$**
- Up: 0.12
- Left: 0.1
- Right: 0.1
- Down: 0.1

**Feature Set $\phi_3$**
- U: 0.12
- L: 0.1
- C: 0.1
- R: 0.1
- D: 0.1
- It: 0.005

**Feature Set $\phi_4$**
- UL: 0.02
- U: 0.015
- UR: 0.015
- L: 0.01
- C: 0.01
- R: 0.015
- DL: 0.015
- D: 0.015
- DR: 0.015
- It: 0.005
Adaptive Feature Sets

- Features selected as a function of number of tasks
- Initial features: 10 (values: $4^9 \times 3$)
- Final features: 4 (values: $4^4$)
Results: Online Feature Selection

- Effect of priors: episodes 1 and convergence