Adapting Behaviour via Intrinsic Rewards to Learn Predictions

Martha White
Assistant Professor
University of Alberta
Joint work with
Cam Linke, Nadia Ady, Thomas Degris and Adam White
Motivation

• Imagine an RL agent is wandering around making many predictions about the world

  • What will happen if I pick up this object?
  
  • How many steps until I get to the door?

• How should it act, to make those predictions more accurate?

Would I bump if I drive-forward

If I navigate-to-lab, would try-to-pugin succeed?
This problem has been studied in many flavours
This problem has been studied in many flavours

- Active learning and optimal experimental design
  - e.g., batch of data, choose most useful subset to label

- Active perception and attention
  - e.g., what part of an image should the agent look at
This problem has been studied in many flavours

- Active learning and optimal experimental design
  - e.g., batch of data, choose most useful subset to label

- Active perception and attention
  - e.g., what part of an image should the agent look at

- How to adapt behaviour to learn **many parallel predictions online** has not been explicitly formalized
This talk is about problem formulation

- It is about **understanding how to formalize** active data gathering for **learning predictions** in parallel

- This talk is **not about**
  
  - a solution strategy for exploration

  - new algorithms

On arXiv next week:
Adapting Behaviour via Intrinsic Reward: A Survey and Empirical Study
Problem Setting
Problem Setting

- N targets, for which we have N prediction learners
- Online setting: one stream of (nonstationary) experience
- **Goal**: take actions to provide data that makes the N prediction learners as accurate as possible
Problem Setting

• N targets, for which we have N prediction learners

• Online setting: one stream of (nonstationary) experience

• **Goal**: take actions to provide data that makes the N prediction learners as accurate as possible

• **Issues**:
  
  • unknown world and what data is useful is not obvious
  
  • different samples are useful for different learners — the behavior needs to balance these needs
How can we formalize this problem?

- **Ideally**, maximize prediction accuracy across time
- **Hard** to specify as a continuous optimization problem
  - action selection indirectly affects prediction accuracy
Naturally formulated as an RL problem

• **Reward** the behavior for taking actions that produce data that is “**useful**” for the prediction learners

• The **actions** are still the actions in the underlying MDP

• The **states** should be the MDP state and the parameters of the N prediction learners
The Parallel Prediction Learning Problem

Environment

Many Learners

Behavior Agent

Intrinsic Reward computation

$A_t, O_{t+1}$

$A_1, A_2, A_N$

$A_t$

action

$R_{t+1}, O_{t+1}$
The Parallel Prediction Learning Problem

The key to the formulation is the reward definition
The Parallel Prediction Learning Problem

The key to the formulation is the reward definition

\[
\sum_{i=1}^{N} |\hat{y}_i - y_i|
\]
Issue with Error-based rewards

- Prediction Error includes Variance of Targets
  \[ \sum_{i=1}^{N} |\hat{y}_i - y_i| \]
- What we really want is error to true target
  \[ \sum_{i=1}^{N} |\hat{y}_i - \mathbb{E}[Y_i]| \]
- …But we do not have the true target
The first step is to understand existing intrinsic rewards
Only minor connection to Intrinsic Motivation in RL

- There the goal is to find an optimal policy
  - internal reward is added just to encourage exploration
  - accuracy of prediction is secondary, if considered at all
- Example: Count-based methods encourage systematically revisiting all of the space
- Example: Use Model Error to encourage exploration (though predictions from the model are not used)
The first step is to understand existing intrinsic rewards using an empirical study.
Let's start in the simplest setting

- There is **no context** or state

- Each prediction learner is estimating the **mean** of a **different target** (N independent learners)

- Each **action** only generates **data** for **one** prediction learner

  - there are N actions
Formalizing a Testbed

\[ \theta(t, i) \] distribution for \( i \)th target on timestep \( t \)
This setting still has the key properties of the problem

- Must balance needs of several learners
- Some learners might have harder to estimate targets
- The world is unknown and potentially non-stationary/partially observable
- An appropriate reward for behavior is not obvious
Target distributions

\[ \theta(t, i) \overset{\text{def}}{=} \mathcal{N}(\mu_{t,i}, \sigma_{t,i}^2) \]

for \( \mu_{t+1,i} \leftarrow \Gamma_{[-50,50]} (\mu_{t,i} + \mathcal{N}(0, \xi_{t,i}^2)) \)

\( \sigma_{t,i} \in \mathbb{R} \) for sampling noise

\( \xi_{t,i} \in \mathbb{R} \) for the rate of drift
Examples of targets

Figure 1: Our parallel multi-prediction learning formulation.

Table 1: The target distributions for each action in the Switch Drifter-Distractor problem. Phase 1 lasts 50,000 time steps, then targets are permuted and remain fixed for the remainder of the experiment (another 100,000 steps). The parameters for each target type—constant, distractor and drifter—are given on the right. These parameters define $\mathbf{w}(t, i)$ in Equation (3), where $\sigma^2$ is the sampling variance and $\xi^2$ is the drift variance.

<table>
<thead>
<tr>
<th>Target Type</th>
<th>$\sigma^2$</th>
<th>$\xi^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Distractor</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Drifter</td>
<td>0</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Table 2: Parameters defining $\mathbf{w}(t, i)$ for each prediction task in the Jumpy Eight-Action problem, where $\sigma^2$ is the sampling variance and $\xi^2$ is the drift variance for Equation (3). Prediction Task 6 is special, defined in Equation (4).
Example of good behavior

- High-variance target (1)
- Constant target (3)
- Drifting target (2)
- High-variance target (4)

Time Steps (in thousands)

Probability of selecting each action

Target values

Change in uncertainty

Stepsize change

Weight change

Variance of prediction

Unexpected demon error

Error derivative change

Bayesian surprise

Expected error

Absolute error reduction

Squared error
What intrinsic rewards give us this good behavior?
A survey of intrinsic rewards

- Violated Expectations (surprise)
  - Absolute Error, Squared Error, Expected Error (windowed average)

- Learning Progress (reduction in error)
  - Error Reduction, Error Derivative, Positive Error Part

- Amount of Learning (change in model)
  - Bayesian surprise, Weight Change, Variance of Prediction
<table>
<thead>
<tr>
<th>Reward name</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute Error*</td>
<td>$</td>
</tr>
<tr>
<td>(Schmidhuber, 1991b)</td>
<td></td>
</tr>
<tr>
<td>Squared Error*</td>
<td>$\delta_{t,i}^2$</td>
</tr>
<tr>
<td>(Gordon and Ahissar, 2011)</td>
<td></td>
</tr>
</tbody>
</table>
A survey of intrinsic rewards

- Violated Expectations (surprise)
  - Absolute Error, Squared Error, Expected Error (windowed average)

- **Learning Progress (reduction in error)**
  - Error Reduction, Error Derivative, Positive Error Part

- Amount of Learning (change in model)
  - Bayesian surprise, Weight Change, Variance of Prediction
Learning Progress

Error Reduction*
(Schmidhuber, 1991a)

Positive Error Part
(Mirolli and Baldassarre, 2013)

Error Derivative
(Oudeyer et al., 2007)

| \delta_{t-1,i} | - | \delta_{t,i} |

max(\delta_{t,i}, 0)

\frac{1}{\eta} \sum_{j=0}^{\eta} \delta_{t-(j-\tau-\eta),i}^2 - \frac{1}{\eta} \sum_{j=0}^{\eta} \delta_{t-(j-\eta),i}^2

\text{Variance of error} \ Var[\delta_{t,i}]

\text{Variance of Prediction} \ Var[\hat{\delta}_{t,i}]

\text{Uncertainty Reduction} \ Var[\hat{\delta}_{t-1,i}] - Var[\hat{\delta}_{t,i}]

\text{Uncertainty Change} \ | Var[\hat{\delta}_{t-1,i}] - Var[\hat{\delta}_{t,i}]|
A survey of intrinsic rewards

• Violated Expectations (surprise)
  • Absolute Error, Squared Error, Expected Error (windowed average)

• Learning Progress (reduction in error)
  • Error Reduction, Error Derivative, Positive Error Part

• Amount of Learning (change in model)
  • Bayesian surprise, Weight Change, Variance of Prediction
## Amount of Learning

### Weight Change*

\[ \|w_{t,i} - w_{t-1,i}\|_1 \]

### Bayesian Surprise*

(Itti and Baldi, 2006)

\[ \text{KL}(p_{w_t} \| p_{w_{t-1}}) \]

where \( p_{w_t}(\theta) = p_{w_{t-1}}(\theta|y_t) = \frac{p(y_t|\theta)p_{w_{t-1}}(\theta)}{p_{w_{t-1}}(y_t)} \)

### Table 3: Intrinsic rewards investigated in this work. Separate statistics are maintained for each learning task \( i \), and only updated when task \( i \) is selected by the agent. Starred rewards are included in the results. Non-starred rewards were tested but performed poorly.

<table>
<thead>
<tr>
<th>Reward Name</th>
<th>( R_{t,i} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute Error*</td>
<td>(</td>
</tr>
<tr>
<td>Squared Error*</td>
<td>( 2t,i )</td>
</tr>
<tr>
<td>Expected Error*</td>
<td>( t,i )</td>
</tr>
<tr>
<td>Unexpected Demon Error*</td>
<td>( x_t )</td>
</tr>
<tr>
<td>Error Reduction*</td>
<td>( t_1,i )</td>
</tr>
<tr>
<td>Positive Error Part</td>
<td>( \max(t,i,0) )</td>
</tr>
<tr>
<td>Error Derivative</td>
<td>( 1\cdot x_j^{0} = 0 )</td>
</tr>
<tr>
<td>Weight Change*</td>
<td>( k_{w_{t,i}} )</td>
</tr>
<tr>
<td>Step-size Change*</td>
<td>( |t_1,i| )</td>
</tr>
<tr>
<td>Bayesian Surprise*</td>
<td>( \log_2 \frac{c_{2t,i} + c_{2t-1,i}}{c_{2t,i} + (\hat{y}<em>{t,i} - \hat{y}</em>{t-1,i})^2} )</td>
</tr>
</tbody>
</table>

\( c_{2t,i} \) is an estimate of \( \text{Var}[y_{t,i}] \), using an exponential average variant of Welford’s algorithm, with \( c_{2t,i} = (1 - \epsilon)c_{2t-1,i} + \epsilon(y_{t,i} - \hat{y}_{t,i})^2 \) for \( 0 < \epsilon < 1 \).
Which intrinsic reward is “best”? (And feasible)

- Bayesian surprise with Bayesian prediction learners in expectation equals Information Gain
  - Information Gain = Mutual Information between targets and parameters
- Maximize expected Bayesian surprise = maximize Info Gain
- Bayesian surprise with certain Bayesian or MAP learners corresponds to Weight Change with a MAP learner
Which intrinsic reward is “best”? (And feasible)

- Bayesian surprise with Bayesian prediction learners in expectation equals Information Gain
  - Information Gain = Mutual Information between targets and parameters
- Maximize expected Bayesian surprise = maximize Info Gain
- Bayesian surprise with certain Bayesian or MAP learners corresponds to Weight Change with a MAP learner

**Potential Proposal:** Weight Change for an approximate MAP learner
Let’s give this a name

**Introspective prediction learners**

increase their rate of learning when progress is possible

and decrease when it is not
An undesirable situation

- Imagine an SGD learner with a fixed stepsize
  - a non-introspective learner

- Assume the target is drawn from a standard normal

- This learner chases noise
  - the weights will always change, because of stochasticity in targets

- Weight Change is not meaningful for such a learner
Can’t the intrinsic reward account for bad learners?

• If a prediction learner is chasing noise, then intrinsic reward could simply be set to zero to stop wasted effort

• If this can be recognized to modify intrinsic reward, then the prediction learner can recognize it too

• Separation of responsibilities:
  
  • Prediction learner modulates learning

  • Behavior trusts prediction learners to modulate learning, focuses on exploring and balancing needs across learners
Experiment

- Compared 15 intrinsic rewards in Drifter-Distractor

- Behavior is a stochastic policy over 4 actions, learned with a Gradient Bandit algorithm

- Prediction learner uses
  - fixed stepsizes (non-introspective)
  - adaptive stepsizes (introspective)

- 200 runs, ~14000 parameters swept

- Performed other experiments with sudden changes, more targets
Contrasting learners

**Constant** can be learned fast, should stop selecting

**Distractors** take longer (due to stochasticity), but also should stop
being selected

**Drifting** needs to be selected forever (non-stationary)
Violated Expectations and Learning Progress do Poorly

Non-introspective learners

Introspective learners

Constant Drifting Distractors
Key take-away

Intrinsic rewards based on the amount of learning can generate useful behaviour if each individual learner is introspective.
Scaling to the general RL setting

- The strategies scale to the general RL setting
  - Weight change can easily be calculated for parameterized functions
  - Stepsize adaptation methods can still provide introspective learners

- There are some important differences
  - Learners would no longer be independent, as data for one can provide information for others
  - We will need to use off-policy methods

- It remains to be seen whether the conclusions scale
Open Questions

- What other intrinsic rewards could we consider?
- Can we prove that maximizing expected intrinsic reward provides guarantees on prediction accuracy?
- What behaviour do these intrinsic rewards induce in MDPs?
- Is this an easier exploration problem, than maximizing (sparse) environment rewards?
- Do we need smarter exploration strategies in MDPs?
Open Questions

• What other intrinsic rewards could we consider?

• Can we prove that maximizing expected intrinsic reward provides guarantees on prediction accuracy?

• What behaviour do these intrinsic rewards induce in MDPs?

• Is this an easier exploration problem, than maximizing (sparse) environment rewards?

• Do we need smarter exploration strategies in MDPs?

Thank you!