Preference-based Policy Learning

Riad Akrour, Marc Schoenauer and Michèle Sebag
TAO
CNRS − INRIA − Université Paris-Sud
FirstName.Name@inria.fr

European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases 2011

Setting

- Output: A policy $\pi : \mathcal{S} \mapsto \mathcal{A}$ mapping state on action
- Input: A weak expert
  - ✗ Does not know how to solve the problem globally
  - ✗ Does not know what is good locally
  - ✔ Given two behaviors he is able to prefer one of them

RL : forcluded as no reward available

IRL: forcluded as insufficient expertise

Motivations

- Context: Swarm robotics
- Requirements on approach: run on-board
  - Using only internal robot sensors (no ground truth)
  - Avoid reality gap due to using simulators

State of art (1/2)

Reinforcement Learning *[Sutton & Barto 98]*

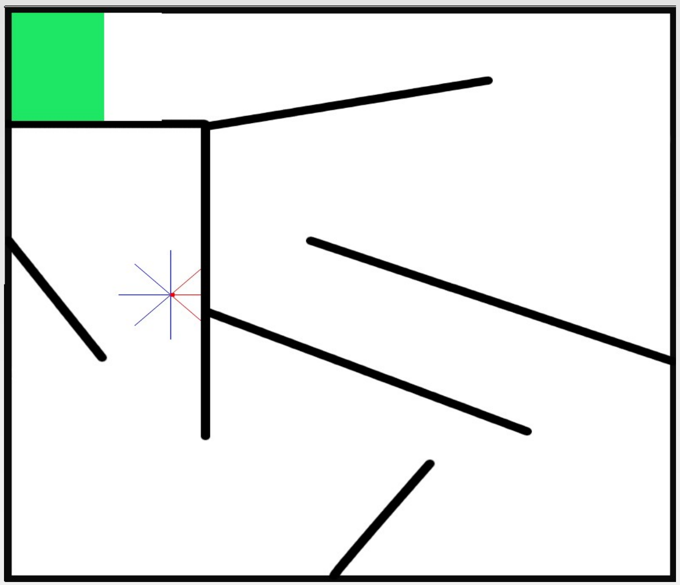
- Handcraft a reward function $\mathcal{R} : (\mathcal{S}, \mathcal{A}) \mapsto \mathbb{R}$
  - Maximize $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$

- Natural to define in some applications (episodic games: win or lose)

- Issues with high dimensional continuous state/action spaces (robot sensory-motor data)

Issues in RL

How to define the reward?



$(\mathcal{S}, \mathcal{A})$

```
-0.533333 -0.3 1 1 1 1 1 -0.3
-0.533333 -0.333333 1 1 1 1 1 -0.3
-0.533333 -0.333333 1 1 1 1 1 -0.266667
-0.533333 -0.366667 1 1 1 1 1 -0.233333
-0.533333 -0.4 1 1 1 1 1 -0.2
-0.566667 -0.4 1 1 1 1 1 -0.166667
-0.566667 -0.433333 1 1 1 1 1 -0.166667
-0.566667 -0.466667 1 1 1 1 1 -0.1
-0.566667 -0.5 0.833333 1 1 1 1 -0.0666667
-0.566667 -0.5 0.633333 1 1 1 1 -0.0666667
-0.6 -0.533333 0.433333 1 1 1 1 -0.0333333
-0.6 -0.533333 0.333333 1 1 1 1 0
-0.6 -0.533333 0.2 1 1 1 1 0.0333333
-0.6 -0.566667 0.2 1 1 1 1 0.0333333
```

$\xrightarrow{\mathcal{R}}$?

Hint: +1 at the green zone raises difficulties (partial observability)

State of art (2/2)

Apprenticeship Learning *[Abbeel & Ng 04]*

- Principle
  - An expert demonstrates some near-optimal trajectories
  - Used to get the underlying reward, then policy
- Many learning options (what, how)
  - But requires near-optimal trajectories
- Our case: Not even good-enough trajectories
  - Many degrees of freedom
  - Robot swarm

Issues in IRL

How to demonstrate an optimal policy to a swarm?

Liu & Winfield 2010

The dots on the floor are Epucks robots

Preference-based Policy Learning

- Iterate
  - Expert: expresses preferences over demonstrated policies
  - Robot: learns a *policy return estimate* (PRE) from Expert preferences
  - Robot: self-trains by optimizing **PRE + an exploration term**, to demonstrate a new policy

Outline

- Background
- **Preference-based Policy learning**
  - Learning the PRE
  - Exploration/Exploitation dilemma
  - Self-training
  - Overview of Algorithm
  - Experiments
- Discussion

Policy Return Estimate
WHAT, HOW

- A scoring function for guiding policy search (during self-training)

- Linear function $J_w(\mu) = \langle w, \mu \rangle$ learned by optimizing a standard convex problem *[Joachims 05]*:

$$(P) \left\{ \begin{array}{ll} \text{Minimize} & \frac{1}{2}\ ||\mathbf{w}||^2 + C \sum_{i,j=1,i>j}^{t} \xi_{i,j} \\ \text{subject to} & (\langle \mathbf{w}, \mu_i \rangle - \langle \mathbf{w}, \mu_j \rangle \geq 1 - \xi_{i,j}) \text{ and } (\xi_{i,j} \geq 0) \text{ for all } \mu_i \succ \mu_j \end{array} \right.$$

- Standard learning to rank, using archived expert preferences

Policy Return Estimate
The search space

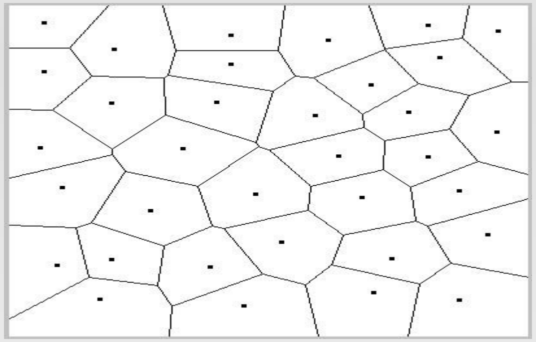
- Search space: policy space (parametric space)
  - But unlikely to learn good ranking functions on parametric space
  - Inconsistent in presence of noise

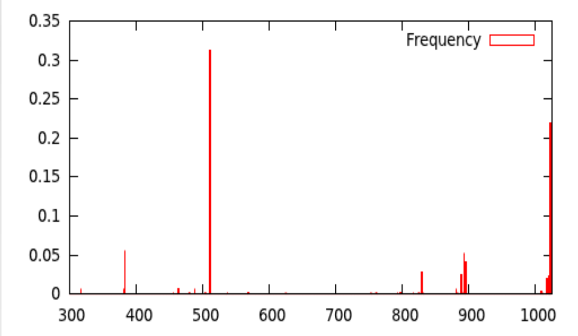
- Use behavioral representation μ

Behavioral representation

- Trajectory → quantized (ε-means) **S**ensory-**M**otor **S**tates

```
-0.533333 -0.3 1 1 1 1 1 -0.3
-0.533333 -0.333333 1 1 1 1 1 -0.3
-0.533333 -0.333333 1 1 1 1 1 -0.266667
-0.533333 -0.366667 1 1 1 1 1 -0.233333
-0.533333 -0.4 1 1 1 1 1 -0.2
-0.566667 -0.4 1 1 1 1 1 -0.166667
-0.566667 -0.433333 1 1 1 1 1 -0.166667
-0.566667 -0.466667 1 1 1 1 1 -0.1
-0.566667 -0.5 0.833333 1 1 1 1 -0.0666667
-0.566667 -0.5 0.633333 1 1 1 1 -0.0666667
-0.6 -0.533333 0.433333 1 1 1 1 -0.0333333
```





Policy: behavioral representation μ as a histogram of sms

- Linear PRE implies setting rewards on SMS

as $J_w(\mu) = \frac{1}{H} \sum_{t=0}^{H-1} R(s_t, a_t)$ given $R(s_t, a_t) = w_{cluster(s_t, a_t)}$

Exploration/Exploitation

- PRE defined over SMS of demonstrated policies
  - Need to enforce exploration

- Exploration term: min of normalized distance w.r.t already demonstrated policies
  - Given Π the archive of already demonstrated policies
  - Define $E(\mu) = \min_{\mu' \in \Pi} \Delta(\mu, \mu') = \min_{\mu' \in \Pi} \frac{||\mu - \mu'||^2}{||\mu||^2 ||\mu'||^2}$

Self-training

- Selected policy π_{t+1} maximizes $J_t(\mu) + \alpha_t E_t(\mu)$
- Gradient methods not applicable
  - Use Black-Box optimization algorithm
- α_t does the balance between $J(\mu)$ and $E(\mu)$
- As Expert ranks π_{t+1} , α_t is updated:
  - Increased if progress observed
  - Decreased otherwise

Preference-based Policy Learning
PPL Algorithm

Algorithm 1 Preference-based Policy Learning

$w_0 \leftarrow 0$
$\theta_0 \leftarrow random$
$\Pi_0 \leftarrow \pi_{\theta_0}$
for $t = 0 \rightarrow$ Expert satisfaction **do**
  $\theta_{t+1} = \arg\max_\theta J_t + \alpha_t E_t$ {call Black-Box optization algorithm}
  $\Pi_{t+1} \leftarrow \Pi_t \bigcup \pi_{\theta_{t+1}}$
  Expert updates the preference matrix
  $w_{t+1} \leftarrow$ Solution of (P_{t+1}) {Use a quad. solver. Ex. SVM^{light}}
  if $\exists \pi' \in \Pi_t, \pi' \succ \pi_{\theta_{t+1}}$ **then**
    $\alpha_{t+1} \leftarrow \alpha_t * decrease\_factor$
  else
    $\alpha_{t+1} \leftarrow \alpha_t * increase\_factor$
  end if
end for
return θ_t

Outline

- Background
- Preference-based Policy learning
  - Learning the PRE
  - Exploration/Exploitation dilemma
  - Self-training
  - Algorithm
  - **Experiments**
- Discussion

Experimental goal and setting

Setting: One and two robots

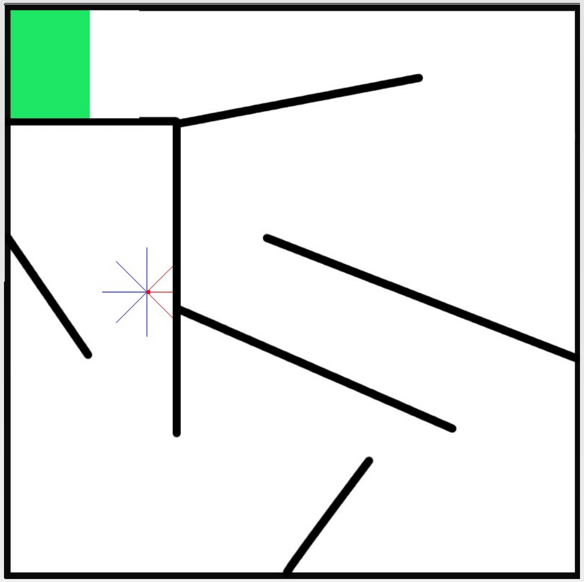
- 8 IR sensors, 2 motor commands (rotation, translation)
- $(\Theta = \mathbb{R}^{121})$ weight of a 1-hidden-layer feed-forward neural net
- Reproducibility
  - Simulator Roborobo http://www.lri.fr/~bredeche
  - Expert preferences emulated using ground truth
- Results averaged over 41 independent runs

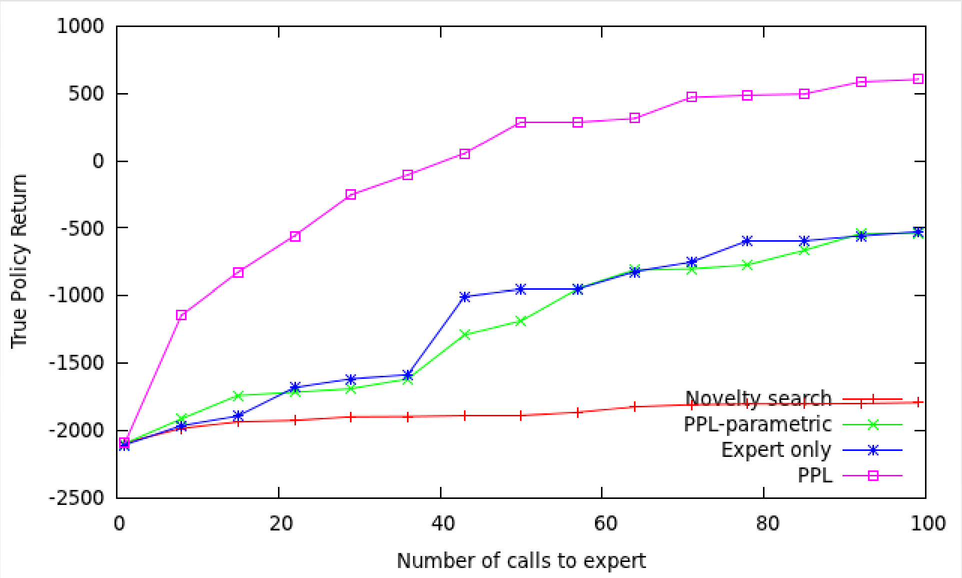
Baselines

- Parametric PPL: Learn PRE over parametric space
- Expert only: Black-Box optimization using emulated preferences
- Novelty Search *[Lehman & Stanley 08]*: Exploration only

The maze problem

- Goal: Shortest path to the green zone

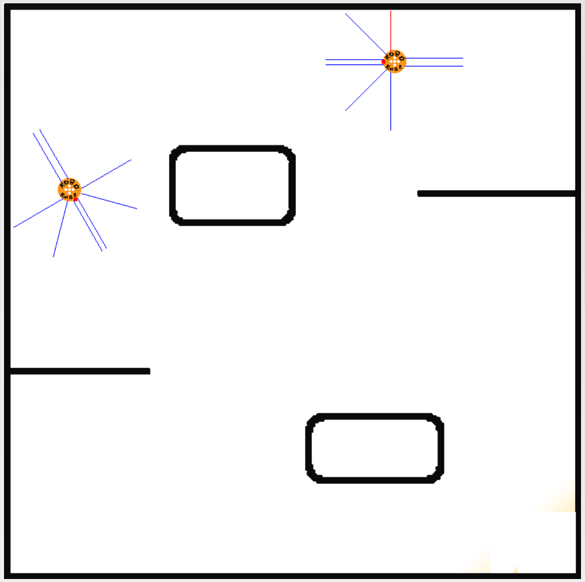


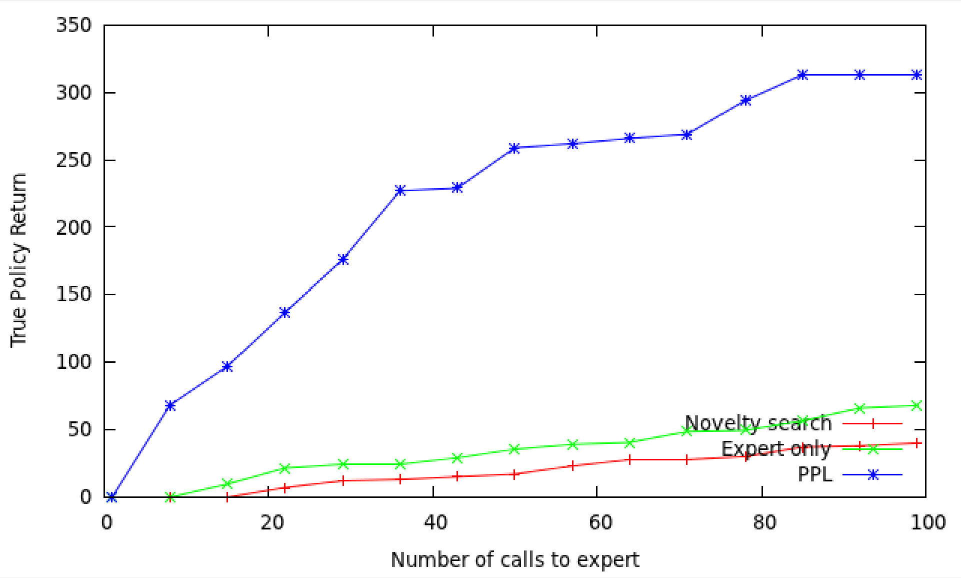


- Reaches the goal in average at the 39th trajectory shown to expert
- PPL performs +53% better than Expert only (¼ evaluations needed)
- PPL-parametric performs the same as Expert only
- Novelty search fails (large search space)

Synchronized exploration

- Goal: Two robots, must stay close while exploring arena





- More difficult problem
- Same conclusions: PPL >> Expert only > Novelty
- PPL performs even better (+354% from Expert Only)

Outline

- Background
- Preference-based Policy learning
  - Learning the PRE
  - Exploration/Exploitation dilemma
  - Self-training
  - Algorithm
  - Experiments
- **Discussion**

Preference Policy Learning

- Pros
  - ✔ Applicable with "informed outsider" experts
  - ✔ Applicable in partially observable settings
  - ✔ Affordable w.r.t. human effort

- Cons w.r.t. embedded robotics
  - ✗ Self-training phase is time/energy consuming

Future work

- Expert may prefer a trajectory because of sub-behavior
  - Cast learning as a Multiple Instance Problem
- Add hierarchy in the clustering algorithm when building μ, and link it to Exploitation/Exploration dilemma
  - Fine grain details for exploitation
  - Less granularity for exploration
- Improve self-training phase
  - See w as a reward and combine policy improvement with black box optimization

References

- Reinforcement learning

  R. Sutton and A. Barto. Reinforcement Learning: An Introduction. MIT Press, Cambridge, 1998.

  C. Szepesvári: Reinforcement Learning Algorithms for MDPs. Wiley Encyclopedia of Operations Research, Wiley, 2010

- Apprenticeship Learning

  P. Abbeel and A.Y. Ng. Apprenticeship Learning via Inverse Reinforcement Learning. on Machine Learning (ICML-2004)

  M. Bain and C. Sammut. A Framework for Behavioural Cloning. Oxford University Press, 1995

- Black-box optimization

  N. Hansen and A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. Evolutionary Computation, 2001

  A. Auger. Convergence Results for the (1,λ)-SA-ES using the Theory of φ-irreducible Markov Chains. Theoretical Computer Science, 2005

- SVM-rank

  T. Joachims. A Support Vector Method for Multivariate Performance Measures (ICML 2005)