TERESA : Telepresence Reinforcement Learning Social Agent.

Learning Social Skills

Kyriacos Shiarlis

University of Amsterdam

16/06/2014

Telepresence - What?

Remotely controlled robots that allow the user to interact with an environment, without being physically present.

Telepresence - Why?

Telepresence allows greater **control** and **interaction** for the remote user.

Telepresence - Why?

Telepresence allows greater **control** and **interaction** for the remote user.

The user also **feels** and **appears** more present

Telepresence - Why?

Telepresence allows greater **control** and **interaction** for the remote user.

The user also **feels** and **appears** more present

Applications include:

- **Assistive technologies:** Remote visits to elderly, disabled, or hospitalised individuals.
- **Industrial:** Remote inspections, conferences, visits.
- **Academic:** Conferences, supervisions.

Telepresence - Why?

Telepresence allows greater **control** and **interaction** for the remote user.

The user also **feels** and **appears** more present

Applications include:

- **Assistive technologies:** Remote visits to elderly, disabled, or hospitalised individuals.
- **Industrial:** Remote inspections, conferences, visits.
- **Academic:** Conferences, supervisions.

TERESA concentrates on deployment in elderly homes.

Limitations

- **Control** of the device can be hard.
- Interaction is not as **natural** as a result.
- Device only allows **audiovisual** interaction.

Project Aims

Practical

- Remove the cognitive load of control.
- Appear socially integrated.

Project Aims

Practical

- Remove the cognitive load of control.
- Appear socially integrated.

Scientific

- To what extent socially acceptable behaviour can be Learned.
- What sort of implicit feedback is needed to achieve this.

Example

Questions

How should a robot approach a group of people? What is the correct distance to stop?

Example

Questions

How should a robot approach a group of people? What is the correct distance to stop?

$\Rightarrow$ Hard Coding Social Norms is very complex.

Example

Questions

How should a robot approach a group of people? What is the correct distance to stop?

$\Rightarrow$ Hard Coding Social Norms is very complex.

Our approach

Experiment $\rightarrow$ Data $\rightarrow$ Offline Learning $\rightarrow$ Semi-autonomous behaviour.

Example

Questions

How should a robot approach a group of people? What is the correct distance to stop?

$\Rightarrow$ Hard Coding Social Norms is very complex.

Our approach

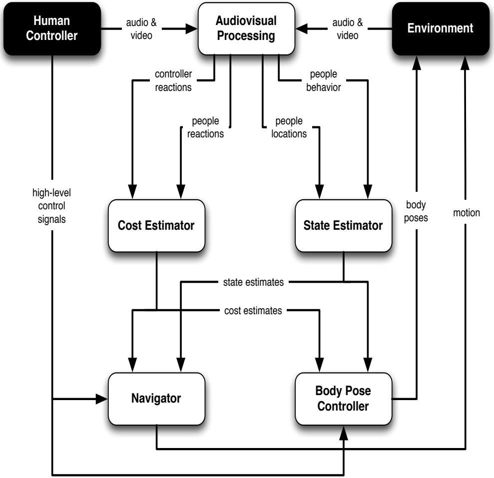
Experiment $\rightarrow$ Data $\rightarrow$ Offline Learning $\rightarrow$ Semi-autonomous behaviour.

$\Rightarrow$ More fluent and natural local-remote user interaction.

Cognitive Architecture

Feedback from:

- Facial analysis.
- Conversation flow/tone.
- Body poses.



Learning Social Skills

How can emotional/implicit feedback from the robot's environment improve its behaviour?

Learning Social Skills

How can emotional/implicit feedback from the robot's environment improve its behaviour?

Example

Robot comes **dangerously** close and at high velocity - Person **frowns** - After learning the robot **avoids** action in similar situations.

Learning Social Skills

How can emotional/implicit feedback from the robot's environment improve its behaviour?

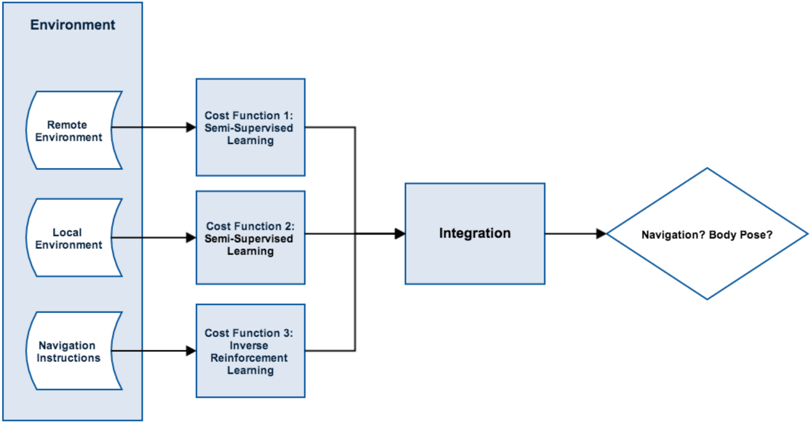
Example

Robot comes **dangerously** close and at high velocity - Person **frowns** - After learning the robot **avoids** action in similar situations.

Does that perform better than hand-coding social behaviour?

Learning Social Skills - Aims

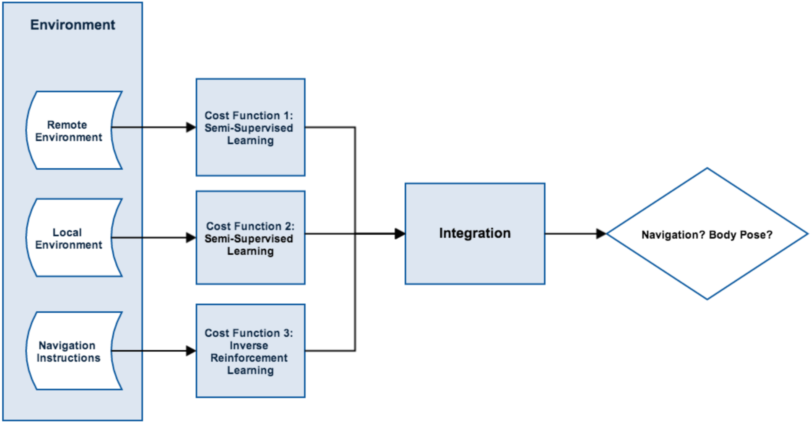
Extract



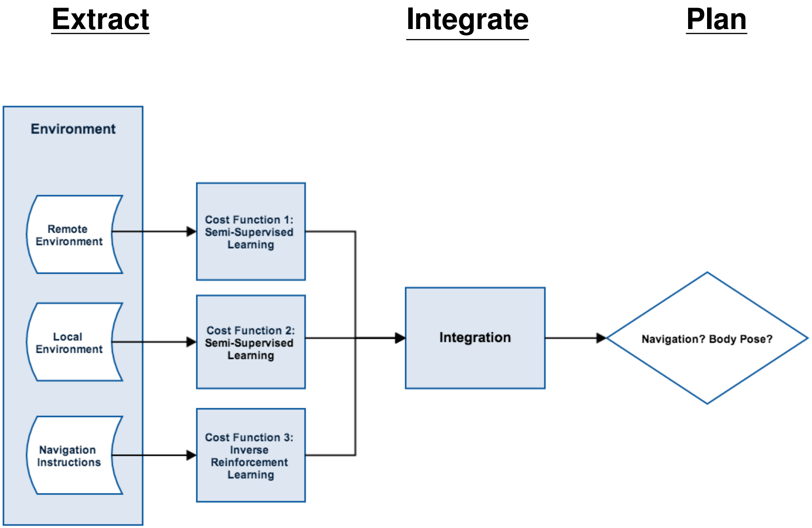
Learning Social Skills - Aims

Extract

Integrate



Learning Social Skills - Aims

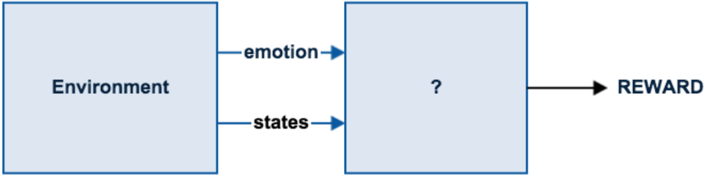


Challenges

Extraction

Extracting reward from the environment is an exercise in implicit feedback.

- Semi-Supervised Learning : Implicit emotional state $\Rightarrow$ Reward.
- Inverse Reinforcement Learning: Expert trajectories $\Rightarrow$ Reward

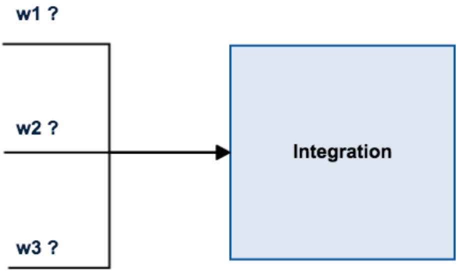


Challenges

Integration

Integration of cost functions should be done intelligently.

- Could be based on individual function confidence.
- Bayesian Approach.



Challenges

Planning

- UvA is responsible for planning body pose policies.

Challenges

Planning

- UvA is responsible for planning body pose policies.
- What are the priorities in social occasions?

Challenges

Planning

- UvA is responsible for planning body pose policies.
- What are the priorities in social occasions?

Challenges

Planning

- UvA is responsible for planning body pose policies.
- What are the priorities in social occasions?
- Collaborating with UPO on Navigation.

Challenges

Planning

- UvA is responsible for planning body pose policies.
- What are the priorities in social occasions?
- Collaborating with UPO on Navigation.
- How will the two be regulated?

Inverse Reinforcement Learning
Definition

Given:

1. Measurements of an agent's behaviour over time, in a variety of circumstances
2. Sensory inputs to the agent.
3. A model of the Environment.

Given:

1. Measurements of an agent's behaviour over time, in a variety of circumstances
2. Sensory inputs to the agent.
3. A model of the Environment.

Determine:

The reward function $R(s, a)$ being optimised.

Inverse Reinforcement Learning

- An **apprentice** observes a state action trajectory $[(s_1, a_1), (s_2, a_2), ..., (s_T, a_T)]$ from an **expert**.
- $MDP_E = < S, A, T, \gamma, R >$ - R is hidden from the apprentice.
- Usually $R = w^T \phi(s, a)$
- So the IRL algorithm takes as input the trajectory and outputs the feature weights **w**.
- These are used by the apprentice to mimic and generalise the expert's preferences.

Inverse Reinforcement Learning

Algorithms work by choosing weights to match certain trajectory statistics e.g:

Feature Expectation : $\Phi_E = \frac{1}{m} \sum_{m=0}^{M} \sum_{t=0}^{T} \phi(s_t, a_t)$

Likelihood : $P(s_{1-T}, a_{1-T} | \boldsymbol{w})$

Inverse Reinforcement Learning

Algorithms work by choosing weights to match certain trajectory statistics e.g:
Feature Expectation : $\Phi_E = \frac{1}{m} \sum_{m=0}^{M} \sum_{t=0}^{T} \phi(s_t, a_t)$
Likelihood : $P(s_{1-T}, a_{1-T} | \boldsymbol{w})$

Problems

- Many Reward functions will cause the observed behaviour. Additional constraints are many times used.
- Each iteration usually requires solving the MDP.

Many Approaches

Max margin + Projection

Ng and Abbeel (2004) successfully applied their algorithms on simulated car driving.

Max Entropy IRL

Ziebart et al (2010) added extra disambiguating constraints and applied to route prediction.

Maximum Margin Planning

Ratliff et al (2006) Posed the problem as a Structured Classification. Again applied to route prediction.

Many more....But.

Partial Observability in IRL

Observation 1:

All literature assumes the expert and apprentice have the same observational capabilities.

Partial Observability in IRL

Observation 1:

All literature assumes the expert and apprentice have the same observational capabilities.

Observation 2:

No principled reason why IRL is better than imitation.

Partial Observability in IRL

Observation 1:

All literature assumes the expert and apprentice have the same observational capabilities.

Observation 2:

No principled reason why IRL is better than imitation.

Motivation

Partial observability is possible the case in TERESA. e.g:

- The Pilot-Expert only senses through a camera.
- The Robot has 360 degree laser range finding capabilities.

Partial Observability in IRL

Observation 1:

All literature assumes the expert and apprentice have the same observational capabilities.

Observation 2:

No principled reason why IRL is better than imitation.

Motivation

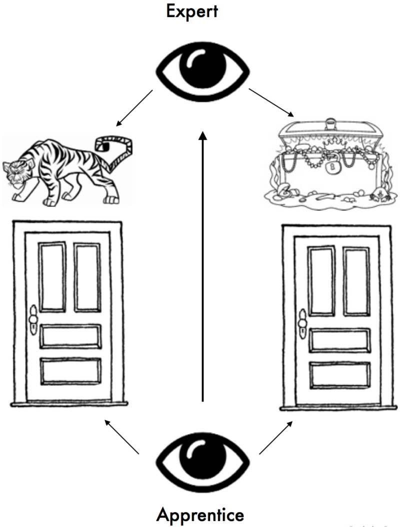
Partial observability is possible the case in TERESA. e.g:

- The Pilot-Expert only senses through a camera.
- The Robot has 360 degree laser range finding capabilities.

=> What are the implications of observability mismatch in IRL?

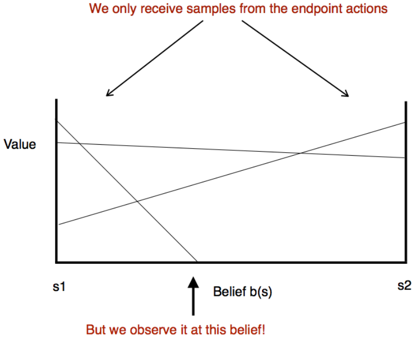
Extreme No 1
Tiger Problem

Apprentice $\rightarrow$ partial observability | Expert $\rightarrow$ full observability



Extreme No 1

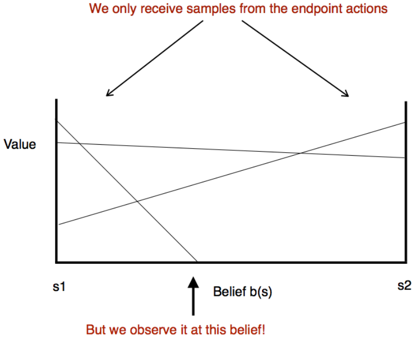
Apprentice $\rightarrow$ partial observability | Expert $\rightarrow$ full observability



- Apprentice receives belief-action trajectories $[(b(s)_1, a_1), (b(s)_2, a_2), ..., (b(s)_T, a_T)]$

Extreme No 1

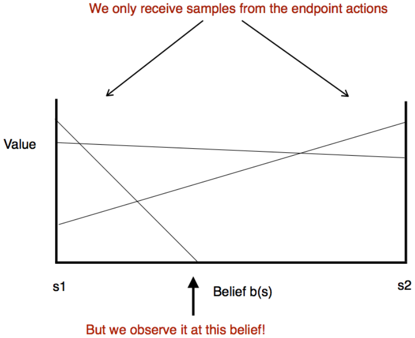
Apprentice → partial observability | Expert → full observability



- Apprentice receives belief-action trajectories $[(b(s)_1, a_1), (b(s)_2, a_2), ..., (b(s)_T, a_T)]$
- No information about what to do in uncertain belief states.

Extreme No 1

Apprentice $\rightarrow$ partial observability | Expert $\rightarrow$ full observability



- Apprentice receives belief-action trajectories $[(b(s)_1, a_1), (b(s)_2, a_2), ..., (b(s)_T, a_T)]$
- No information about what to do in uncertain belief states.
- Less information about what the expert is trying to do!

Extreme No 1

Possible Solutions

Extreme No 1

Possible Solutions

What is the expert trying to do?

Perform forward-backward procedure on beliefs. This will push our samples to the extremes of the simplex.

Extreme No 1
Possible Solutions

What is the expert trying to do?

Perform forward-backward procedure on beliefs. This will push our samples to the extremes of the simplex.

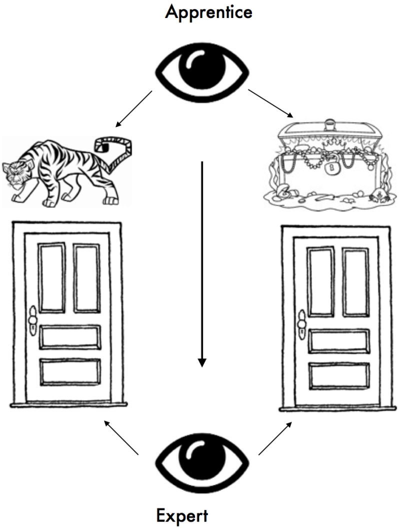
What do we do when uncertain?

Assume a dual controller for the Apprentice.

- The information gathering part of the Reward function is given.
- The control part of the Reward function is learned from the expert trajectories.

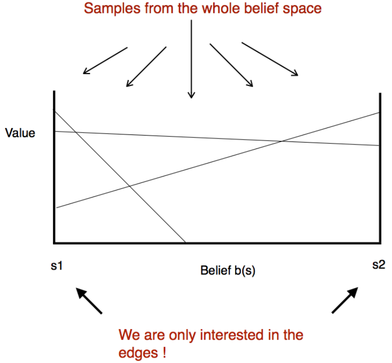
Extreme No 2
Tiger Problem

Apprentice $\rightarrow$ full observability | Expert $\rightarrow$ partial observability



Extreme No 2

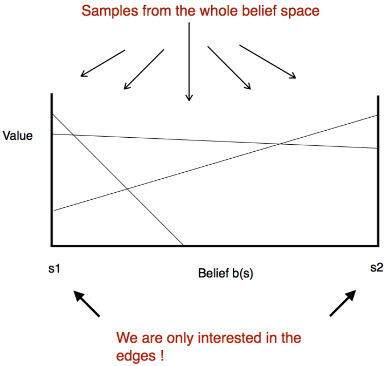
Apprentice $\rightarrow$ full observability | Expert $\rightarrow$ partial observability



- Apprentice receives state-belief(belief)-action trajectories $[(s_1, b_A(b_E(s))_1, a_1), (s_1, b_A(b_E(s))_2, a_2), ..., (s, b_A(b_E(s))_T, a_T)]$

Extreme No 2

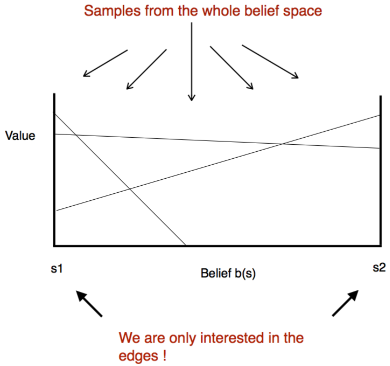
Apprentice $\rightarrow$ full observability | Expert $\rightarrow$ partial observability



- Apprentice receives state-belief(belief)-action trajectories $[(s_1, b_A(b_E(s))_1, a_1), (s_1, b_A(b_E(s))_2, a_2), ..., (s, b_A(b_E(s))_T, a_T)]$
- We don't want to know what to do in uncertain belief states.

Extreme No 2

Apprentice $\rightarrow$ full observability | Expert $\rightarrow$ partial observability



- Apprentice receives state-belief(belief)-action trajectories $[(s_1, b_A(b_E(s))_1, a_1), (s_1, b_A(b_E(s))_2, a_2), \ldots, (s, b_A(b_E(s))_T, a_T)]$
- We don't want to know what to do in uncertain belief states.
- Less information about what the expert is trying to do!

Extreme No 2

Possible Solutions

What is the expert trying to do?

Perform forward-backward procedure on beliefs of beliefs. Again, this will push our samples to the extremes of the simplex.

Extreme No 2
Possible Solutions

What is the expert trying to do?

Perform forward-backward procedure on beliefs of beliefs. Again, this will push our samples to the extremes of the simplex.

How do we ignore uncertain states?

Assume the expert is using a dual-controller.

- The information gathering part of the Reward function is given.
- The control part of the Reward function is learned from the expert trajectories.

We are no longer trying to replicate the expert's behaviour!

We are no longer trying to replicate the expert's behaviour!

$\Rightarrow$ This provides a much more clear motivation for IRL!

We are no longer trying to replicate the expert's behaviour!

$\Rightarrow$ This provides a much more clear motivation for IRL!

As posed, the problem seems unsolvable.

We are no longer trying to replicate the expert's behaviour!

$\Rightarrow$ This provides a much more clear motivation for IRL!

As posed, the problem seems unsolvable.

$\Rightarrow$ We need to make extra assumptions and approximations!