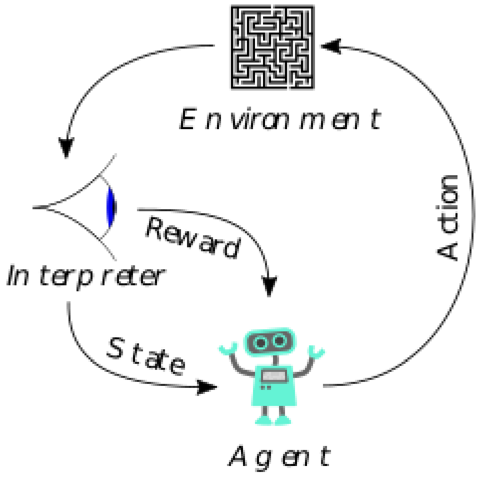
Introduction

My project will be in the area of inverse reinforcement learning (IRL). In brief, an IRL agent learns its own reward function by observing human behaviour and performing Bayesian inference to estimate the true human preferences. This saves the programmer from having to hand-code a reward function, which is difficult and can lead to dangerous unwanted behaviour.

Inverse Reinforcement Learning

IRL is related to reinforcement learning (RL), a paradigm in machine learning where an agent learns to maximise some reward signal.

In a reinforcement learning setup, at each timestep the agent can take an action, then it can observe the updated state and receives a reward signal based on the state of the environment. This reward signal must be computed in some way by an interpreter; for example, by a human overseer, or by some module that takes in states of the environment and returns a reward.



(Image credit wikipedia creative commons)

Building an interpreter that truly captures the desired behaviour and cannot be ‘hacked’ is extremely challenging, and failures can result in very undesirable behaviour. For an example see https://blog.openai.com/faulty-reward-functions/

In this example, the RL agent is trained to play the game CoastRunners. When humans play this game, they race each other around the track, collecting tokens and trying to finish as fast as possible. An RL agent that achieved a superhuman score on this game instead went backwards on the track, crashing into things and catching fire, but managing to collect three tokens exactly as they popped up. It successfully optimised the reward function it was given, but this yielded very undesirable behaviour.

IRL is an approach that aims to solve this problem. The basic idea of IRL is that the RL agent does not have a fixed reward function. Instead, it has a probabilistic model of the true human reward function. This probabilistic model is learnt by observing human actions and trying to infer what human preferences motivated these actions.

In order to infer human preferences from human actions, a model of human decision-making is required - a function that takes human preferences and gives a probability distribution over human actions.

The IRL agent performs Bayesian inference to obtain a model of the true human reward function. [I'll add equations to describe the inference when I put this into Latex]

Prior work

Human decision-making is usually modelled as softmax (Boltzman) rational - the human noisily chooses the highest-reward action. This is obviously not a realistic model of how humans actually behave. There are many extensions of this model that, to the author's knowledge, have not been explored - for example, if the human is biased against taking certain kinds of actions, or if the human is observing only a subset of the true features.

This lack of sophisticated models is problematic, because the IRL agent is unable to infer when a particular outcome is a result of true human preferences, and when it is a result of human irrationality. For example, with a Boltzmann model, if an IRL agent sees a human repeatedly losing a game of chess to a superior opponent, it will infer that the human prefered losing to winning.

A more sophisticated (chess-specific) model of human decision-making might allow the IRL agent to observe that the human took the action most likely to lead to winning according to a specific board value heuristic, even if this was not in fact the action truly most likely to lead to winning.

The field of IRL is very young, and most work so far deals with proof-of-concept experiments with toy examples rather than real-life applications in robotics. Examples of prior work include a game where the IRL agents has two options, to obey or disobey a human’s order, or a setup with a multi-armed bandit where the IRL agent can pull the levers on the human’s behalf. [citations]

My proposed approach

The model I will investigate is one in which the softmax parameter can vary with the state. This is a more realistic model of human reasoning than the simple softmax model. Humans become more noisy than usual in certain situations, such as when distracted, tired or drunk.

$$\mathbb{P}(\pi(s) = a) \propto exp(\frac{q(a, s)}{\beta(s)})$$

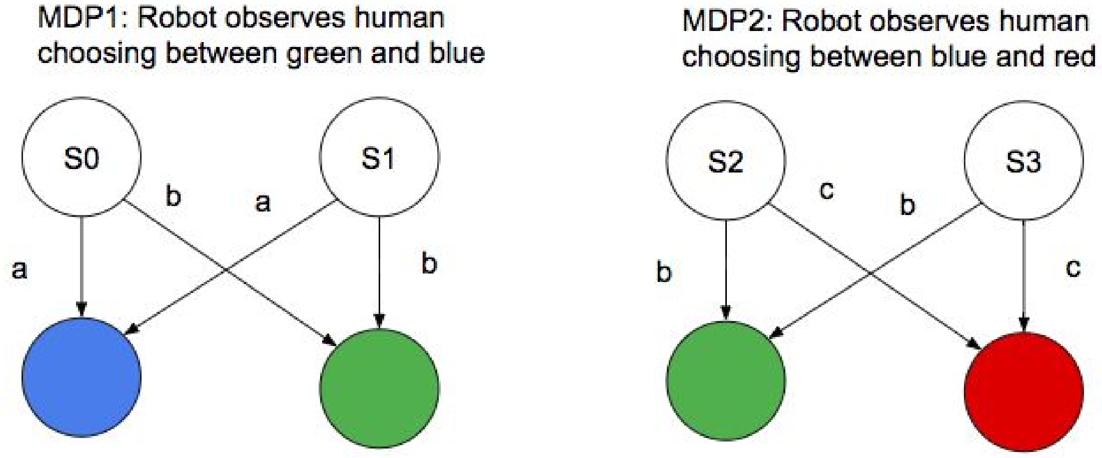
Success Criterion

I will attempt to build a model that outperforms current models in a certain set of toy examples.

The success criterion is to demonstrate empirically that this model provides an advantage over the simpler model. More precisely, it should provide an advantage in a simple Markov Decision Process (MDP) setup where a simulated robot can act on the simulated human’s behalf based on its model of human preferences.

This MDP setup should correspond to a reasonable real-life example of human behaviour - as a slightly silly example, we could imagine a human who might usually prefer to drive to save money, but might order a taxi instead when drunk because of increased risk of accidents. Current IRL models would be unable to learn the true motivations behind this behaviour - they would either conclude that the human did not have a preference to save money, or conclude that the fact they chose to take a taxi was due to random noise. I will show that my model outperforms current IRL models in some setup similar to this.

Another example of where the variable-irrationality model might be useful is this setup, where the human starts randomly in one of the starting states S, and then can choose a colour. Assume that the human's true preference ordering is green < blue <red. Assume further that the human is very irrational in S3 and chooses randomly. Assume that in S0 and S1 the human chooses almost perfectly rationally, choosing blue 90% of the time, and in S2 they also choose almost perfectly rationally, choosing red 95% of the time. Now the choice frequency of red is 0.725 and the choice frequency of blue is 0.9. An IRL agent with a constant model of human irrationality would assign a higher value to blue because it is chosen more frequently. [citation] However, an IRL agent with a variable irrationality model would correctly infer that red has a higher value, and moreover would infer that the human is very noisy in S3 and that future observations of human actions in that state should not be trusted.


MDP1: Robot observes human choosing between green and blue
S0
S1
a
b
a
b
MDP2: Robot observes human choosing between blue and red
S2
S3
b
c
b
c

To succeed, I should empirically demonstrate the advantage of the variable-rationality model. In the example MDP above, I could randomly generate many different values of the irrationality parameters and the human's preferences, then simulate the IRL agents with the variable and constant rationality models, and compare the score they achieved. The score is the utility measured according to the human's true values when the IRL agent makes choices on the human's behalf.

The human behaviour will be simulated using a variable-rationality model, and I will compare the performance of the simulated robot when the robot has the constant rationality model of human behaviour to where it has a variable-rationality model of human behaviour. Using simulated human behaviour rather than real human behaviour is not ideal, but it is a standard approach in the field of IRL, and gathering real human data is outside the scope of this project.

General points on process

I will keep a project log book with my thoughts, questions and results from each stage. In addition I will send weekly updates to my supervisors each Friday. I will be able to use these when I start writing up the dissertation.

As this is a research project, it is probably best to be reasonably flexible, and to pursue whichever direction seems most promising, especially once I have met my basic success criterion.

I will generally aim for proof-of-concept results with small, toy models rather than trying to use real-life data or solve robotics problems. Most of the work on IRL so far is in this vein, and solving IRL perfectly is computationally intractable (although some work has been done on approximations). As only a small number of different models have been explored it seems productive to shallowly investigate several rather than scaling up one particular approach without exploring others.

Overview of work stages

Stage	Description	End date	Hours
1 (wk 3-4)	Do a literature review - the 5 most relevant papers are probably: CIRL, Ignorant, Inconsistent Agents, Robot Obedience, The off-switch game, Stuart Armstrong's paper (forthcoming). Work through WebPPL tutorial	03/11	30 (¼ of 2 60hr wks)
2 (wk 5-6)	Formulate several promising toy MDP setups that are relevant to real-world scenarios. Prove properties about the advantage gained by using the variable irrationality model in the toy environments	17/11	20 (wk 5 blues!)
3 (wk 7-8)	Code up the model of human action and the robot's inference algorithm in WebPPL.	1/12	30
4 xmas	Empirically demonstrate the gain from using the variable irrationality model in at least one toy setup, using WebPPL	15/12	40
5 xmas	Try to prove the advantages of using the variable irrationality model in a wider class of environments	29/12	30

6 xmas	Try to empirically demonstrate the gain from using the variable irrationality model in a wider class of MDPs (maybe certain types of gridworld)	12/01	40
7 (wk 1-2)	Make visualisations of the results from stages 3 + 5	26/01	30
8 (wk 3-4)	1st week: Prepare progress report [Spare week to catch up if earlier parts run over, or work on extensions].	09/02	30
9 (wk 5-6)	Write up preparation and implementation. (I will have been writing up as I go along a reasonable amount)	23/02	20
10 (wk 7-8)	Write up evaluation, conclusions + introduction, add bibliography, appendices, cover sheets etc and make sure everything is correctly formatted and meets the requirements	09/03	30

Detailed description of work stages

Stage 1

Read and make notes on papers that:

- Formulate the IRL problem setup
- Discuss models of human irrationality within this setup
- Investigate what happens when the model of human irrationality is mis-specified

References will be properly formatted in the final version!

Currently the papers I am aware of include:

- Inverse Reinforcement Learning
- CIRL https://arxiv.org/abs/1606.03137
- MaxEnt IRL
- Bayesian IRL
- Algorithms for inverse reinforcement learning
- The Off-switch game
- ‘Should Robots Be Obedient?’
- ‘Learning the preferences of ignorant, inconsistent agents
- Stuart Armstrong’s paper on the fundamental difficulty of the IRL problem (forthcoming)
- Concrete problems
- MIRI research Agenda
- FLI research agenda

- Superintelligence

The most relevant ones are:

- Bayesian IRL
- ‘Should Robots Be Obedient?’
- ‘Learning the preferences of ignorant, inconsistent agents
- Stuart Armstrong’s paper on the fundamental difficulty of the IRL problem (forthcoming)

I will check the references of these papers and check with my supervisors to ensure I’m not missing any prior work, as well as contacting the author of the most relevant paper (ignorant inconsistent agents).
I will work through the online tutorial and textbook for using WebPPL to work with agent models in an IRL setting.

Stage 2

I will attempt to formulate several Markov Decision Process (MDP) setups that:

- are relevant to real-world scenarios (i.e., can be viewed as a model of a real-life and important human decisions that involve variable irrationality, such as whether to drive when drunk)
- allow human behaviour to be more accurately modelled if the human’s rationality can depend on the state
- are simple and easy to work with (contain only a handful of states)
- are relatively general (e.g. do not require the states to have very specific relative values)

I will then assume that real human behaviour follows the variable irrationality model in this MDP. This assumption is justified by the fact that the setup of the MDP is chosen to reflect a real-life situation where a common-sense interpretation of human behaviour is that they are being variably irrational. I will then attempt to prove that using a variable irrationality model allows the robot to form a significantly more accurate model of human behaviour, and allows it to generate more value for the human when it can take actions on the human’s behalf. Working on this will involve mostly pen-and-paper maths, and maybe writing some code or plotting some graphs to get a better intuition about what is going on.

Stage 3

I will code a ‘human simulator’ in WebPPL - an agent that noisily chooses the highest-reward action, with the noise dependent on the state. This should be straightforward.

Coding the robot’s inference algorithm may be more complicated, as it is much more computationally intensive so may need to be more heavily optimised. It needs to compute the posterior probability of the state values given human actions, which requires it to compute the probability of the observed action for all possible state values, for all human irrationality functions. The best way to do this is probably Markov Chain Monte Carlo (MCMC) approximation, which is built into WebPPL. MCMC methods are a way of approximating samples from a desired distribution. A Markov Chain is set up that has the desired posterior distribution as its equilibrium distribution, by basing the transition probabilities on the ratio of the likelihoods of the current state and the proposed next state. [MCMC citation]

If this is too slow or complicated, I will use a discrete parameter grid rather than a continuous parameter space, which makes it possible to perform the Bayesian inference exactly by summing over all possible parameter values. This is the approach used in 'ignorant inconsistent agents'. It is also possible it will be easier and/or more efficient to write a custom MCMC algorithm in Python - I may be able to extend my supervisor's (Dylan Hadfield-Menell's) codebase which includes custom MCMC for IRL.
I will also need to build an inference model that assumes the human always has constant rationality as a baseline to compare the more sophisticated model against.

Stage 4

I will run experiments using the 'human simulator' and the robot's inference algorithm implemented above to empirically demonstrate the advantage of the robot using the more sophisticated variable irrationality model.
In practice this will involve:

- randomly generating different MDPs that are examples of the toy models investigated in stage 2 - for instance, they might have the same transition structure, but different state values
- simulating the human and robot decision-making on these MDPs, and calculating the score for the robot using variable/constant model of human irrationality

By the end of this stage I should have met my success criterion.

Stage 5

This is essentially an extension of stage 3 - using similar techniques and approaches, but trying to prove things that hold more generally. This stage is flexible, and I may try out different extensions as long as I have achieved the success criterion in stage 4. See 'extensions' section for more details.

Stage 6

Again, this stage involves working on an extension to stage 4, empirically demonstrating the properties proved in stage 5. As above, it is flexible and I may work on different extensions depending on what direction seems most promising

Stage 7

I will produce high-quality illustrations of the results I obtained in earlier stages. These will include:

- State diagrams of the MDPs discussed
- Illustrative possible trajectories through the MDP that give an intuition why the approach works
- Graphs of the values obtained using the different methods
- Possibly other illustrations that help explain why the approach is useful, or how the inference is proceeding

Stage 8

I will spend the first week producing the progress report. I will practice the presentation by myself, to a friend, and to a supervisor.
Subsequently I will have a spare week to pursue other extensions or catch up if I have got behind

Stage 9

I will write up the preparation and implementation. For this, I will draw on:

- the notes I made from reading the papers in stage 1
- project log book
- weekly reports for supervisors
- the visualisations I made in stage 7

Stage 10

I will write up the evaluation and conclusions, then add the introduction and ensure that the dissertation is correctly formatted, contains all the required content, etc. I will draw on the same 4 sources as above.

Possible extensions:

1. Explore more complex MDP environments, for example with probablistic transitions or partial observability.
2. Instead of just generating ‘human behaviour’ from a model, use real data of human decisions - for example, route choice while driving. Finding, cleaning and processing this data may be quite a time-consuming task, however.

3. Include a simple communication channel, for example a button for the human to indicate that they made a bad decision
4. Use a model where not just the softmax noise but the bias can vary with state
5. Investigate the priors required to bootstrap an accurate model of human cognition, within this model of human decision-making
6. Investigate the priors required to bootstrap an accurate model of human cognition, in full generality

Resources Declaration

I will be using my own computer:
OS: macOS Sierra 10.12.6
Processor: 1.4 GHz Intel Core i5
Memory: 4 GB 1600 MHz DDR3
Graphics: Intel HD Graphics 5000 1536 MB
Disk: 120GB SSD

I accept full responsibility for this machine and I have made contingency plans to protect myself against hardware and/or software failure.

In case of failure of this machine, I will use the MCS machines

I do not need special resources of any kind. The most intensive computations I will be doing will be MCMC inference, but it is acceptable to use small numbers of features if the computation is too resource-intensive.