5.16 Use of Inverse Reinforcement Learning for Identity Prediction

Use of Inverse Reinforcement Learning for Identity Prediction

Roy Hayes, Jonathan Bao, Peter Beling, Barry Horowitz
Department of Systems Engineering
University of Virginia
Charlottesville, VA 22904
{rlh8t, jhb6f, pb3a,bh8e}@virginia.edu

Abstract. We adopt Markov Decision Processes (MDP) to model sequential decision problems, which have the characteristic that the current decision made by a human decision maker has an uncertain impact on future opportunity. We hypothesize that the individuality of decision makers can be modeled as differences in the reward function under a common MDP model. A machine learning technique, Inverse Reinforcement Learning (IRL), was used to learn an individual's reward function based on limited observation of his or her decision choices. This work serves as an initial investigation for using IRL to analyze decision making, conducted through a human experiment in a cyber shopping environment. Specifically, the ability to determine the demographic identity of users is conducted through prediction analysis and supervised learning. The results show that IRL can be used to correctly identify participants, at a rate of 68% for gender and 66% for one of three college major categories.

1 INTRODUCTION

There has been significant work in the field of machine learning to understand human decision making. Inverse Reinforcement Learning (IRL) is a method for computers to learn to perform complex tasks by watching human operators [2]. IRL is built upon Markov Decision Processes (MDPs), which examine sequential decision making over time. Decision makers are modeled to choose actions based upon maximizing reward, which is captured by a reward function that assigns preferences to being in certain states. Decisions made in the present directly impact future decisions and opportunities, often stochastically, so short-term gain must be balanced against future goals. Decisions are complex because an individual may have many actions to choose between and may have to assimilate various pieces of information and trade-offs between conflicting goals. These types of decisions are commonplace in daily life, from choosing which lane to drive in on the interstate to choosing when to buy or sell stocks.

Our thesis is that IRL techniques can be used to understand human decision making by creating a mathematical model of the human's decision strategy. We do not claim that people solve complex mathematical formulae mentally while making difficult decisions; however, a projection of their preferences can be captured through machine learning. Specifically, we can begin to understand under which conditions an individual would take a certain action and therefore find if people adopt different strategies to the same problem. There is reason for optimism that IRL can model decision making. Researchers have run controlled experiments where a participant is instructed to exhibit certain preferences and have shown heuristically that a computer is able to mimic the behavior by solving a mathematical version of the problem [2]. We feel that IRL does indeed capture aspects of an individual's true decision rules, but the previous work has not tried to verify this important requirement for many applications through rigorous analysis.

1.1 Expected Contribution

This work identifies a bridge between those who develop solutions to sequential decision problems and those who have methods to test and quantify human behavior. In broad terms, the two fields can be defined as machine learning and cognitive science. Machine learning encompasses artificial intelligence and reinforcement learning as researchers who train computers to solve decision problems that may be too difficult for humans to solve. Cognitive science studies how the human brain uses information, and cognitive scientists run controlled experiments to investigate the impact of some changing condition on human performance. The two fields join when researchers use machine learning algorithms to understand human decision making. This work lies in this middle area, as we investigate the potential of IRL to analyze decision strategies through human experimentation.

In the machine learning literature found predominantly in the engineering field, researchers have not validated that IRL captures human decision making through robust experimentation. The literature is focused on improving algorithms in terms of speed and accuracy [6], or adapting work to apply to a larger class of problems [1,3]. The algorithms are heuristically validated by instructing human experts to follow different strategies that map well onto the qualities the computer was trained to learn. The machine learning literature lacks hypothesis testing that would demonstrate that IRL can find differences in decision making between groups of people, and we therefore look to the cognitive science field to find studies analyzing human sequential decision making.

Cognitive science is devoted to understanding how humans make use of information in the brain and is therefore closely related to characterizing decision making. Researchers in cognitive science make use of human experiments to perform hypothesis testing; often to compare two groups of people to one another. There have been studies where IRL and MDPs could be used to analyze the data gathered from human experiments, but researchers lost power by only using results-based analysis. For instance, [4] performed a sophisticated experiment with a motorcycle simulator and asked the riders to identify potential hazards and collected eye-gaze data. The researchers could have sought to understand where the user was looking as a function of the objects on the screen, but instead were relegated to analyzing the higher-level metric of general size of viewing area.

There has been a great deal of work in the economics field to investigate the ability of mathematical models to describe real human behavior. Ref. [5] completed a survey of research in predominantly the economic field that analyzed human decision making with respect to MDPs. They found that humans perform near-optimal behavior in discrete decision problems, but the opposite was true for continuous decision problems. As a case study, they highlighted work by RAND where the decision of Air Force pilots to remain in service or retire to the civilian sector was analyzed. Among other practical conclusions, the work showed that prediction is a valid method for testing MDPs as a decision framework.

2 MATHEMATICAL FORMULATION

IRL refers to any method where a reward function is learned to mimic expert behavior through observation [2]. The foundational premise is that a rational actor may choose between several actions and may conduct analysis to determine the best course of action. Decisions are captured in a mathematical model that can be analyzed and optimized to find the best action. The theory is applied to sequential decision making where the actor will have to make a series of time-ordered decisions. This raises a difficult problem that requires analysis to solve because current actions impact future decisions and opportunities.

2.1 Markov Decision Processes

IRL uses the well-understood framework of Markov Decision Processes (MDP). MDPs are built upon the idea that all of the information one needs to make a decision is characterized by the state of the system. Markov chains become powerful when applied to decision making because the probability of transitioning to a certain state is dependent on the current state and the decision maker's action. The decision maker chooses actions at every time point with the updated knowledge of his or her situation. Decisions can be chosen greedily to maximize short-term gain, but it is clear that since decisions made in the present directly affect future opportunities that a farsighted strategy is needed to make the best possible decisions.

We use the notation from [2] to formulate Markov Decision Processes. An MDP is fully described by the tuple (S, A, T, γ, D, R), where:

- S is the set of all possible states, and the state at time t is given by s_t.
- A is the set of all possible actions, and the particular action chosen at time t is a_t
- T is the function of state transition probabilities.
- $\gamma \in [0,1)$ is a discount factor
- D is the initial-state distribution
- R is the transition reward gained from taking action a_t at s_t while transitioning to s_{t+1}

Once the MDP has been completely formulated, the goal is to solve the problem by developing an optimal policy π that maps an optimal action to every state. Due to the stochastic nature of MDPs, the objective is to choose actions that maximize total expected reward. The goal of the decision maker is to find π that maximizes V_π and therefore know which action to choose at $t = 0$. Once the system transitions to the next state at $t = 1$, then the actor has the information necessary to take the best action, i.e. the actor does not determine at $t = 0$ how he or she will act in the future. Once the problem has been formulated as such, the optimal policy may be derived through dynamic programming or reinforcement learning.

2.2 Discretized-Reward Search Method for IRL

As discussed above, the computer learns to mimic a human by learning the problem that the expert is attempting to solve. [2] places constraints on the problem definition so that IRL uses a linear reward function in order to apply standard optimization techniques to perform policy evaluation. If we relax these constraints, then we void the developed algorithm and must perform IRL in another manner. We have developed an exhaustive search algorithm by discretizing the space of reward functions to a finite set in order to attribute reward functions to actions which, although it has its limitations, works for a broader class of problems.

The process of mapping a reward function to an observed action path x is as follows:

1. Start with initial weight w^0, which is the starting point for weight iteration. There must be some method to iterate through all of the feasible weights. For example, if we choose $|w|_1 = 1$, then the first weight could be $w^0 = (1,0, \dots)$, and the next weight would be $w^1 = (0.9,0.1, \dots)$. Set $i = 0$.
2. Solve or approximate the optimal policy to the MDP where $R = w^i \cdot \varphi(s)$. Simulate an action path using the optimal policy and set as x^i. Use the size of observed actions x as stopping criteria if necessary.
3. Use a reward distance function to find the difference in the rewards generated by x and x^i with respect to w^i and set as d^i.
4. If w^i is not the last weight, then find w^{i+1} and set $i = i + 1$. Return to Step 2.
5. Find the minimum value for d^i and create a set of all the w^i with that value. These are all of the weights and corresponding reward functions that match the observed actions.

There are several design choices in the problem definition that are necessary to implement this method. The set of all weight vectors must be discretized into a finite countable set, and there must be a method for iterating through the set. The MDP must either be able to be solved through dynamic programming or an optimal solution must be approximated with reinforcement learning. Finally, a distance function must be developed to compare the expert's policy and optimal policies generated for candidate reward functions.

3 METHODOLOGY

We conducted human experiments to investigate the capability of using inverse learning methods to perform identity prediction. A task that meets the criteria of a sequential decision problem is online shopping. Shoppers navigate an online environment searching for items, and their actions can be readily extracted from looking at browsing history data. By recording their browsing history, we have a noninvasive sequential view of their actions and can determine how the user assimilated information to make decisions. Inverse learning calculates the user's policy in all situations and will describe the user's objective function. We will be able to characterize how a particular user performs the task of shopping for an item.

We developed an experiment to test how participants perform the task of purchasing a gift using an online shopping website. Each participant underwent a 30 minute experiment, during which they performed 4 trials. At the start of each trial, the participant is given a profile of a person to buy a gift for, which includes personal characteristics and possible suggestions of what that person may like or dislike. The user was given 5 minutes and a budget of $100 to perform the task, during which time he or she browsed the item selection provided by the website and selected one or more gifts to purchase. Participants were not given any instruction except for the profile of the participant and to remain on the shopping site and not view another site. After some pretesting, we determined there were 10 predominant types of pages available at Walmart.com (e.g. store department page, item list page, and checkout page).

3.1 Setup of the MDP and Corresponding IRL Method

We set the state vector to represent the number of pages of each type the user has viewed. State transitions are deterministic, as the user fully decides which page type to view next. With a standard reward function, the optimal policy would simply choose to view the page type with the highest reward over and over again. A reward function that causes users to switch pages, as opposed to choosing the same one over and over again, would be one that took into account the law of diminishing returns. A user may prefer to view one type over another, but as they view that page multiple times they receive decreasing reward. If we let M be the maximum number of pages a user wishes to view of a certain type, we could scale the reward gained from choosing a page by a factor that is inversely proportional to the number of times the page was viewed up to M visits. In Eq. (3.1), a is the action corresponding to the page the user wants to view next, s is the complete state, s_a is the current number of pages of type a that the user has viewed, and w_a is the weight corresponding to that page type.

$$R(s,a) = w_a * \left(M - (s_a + 1)\right) \quad (3.1)$$

This reward function is nonlinear; it is not a linear combination of the state variable because only the part of the state regarding the action taken contributes to the transition reward. We therefore use the Discretized-Reward Search Method. There are many different ways to discretize the space. We chose to have each weight be nonnegative, and the sum of the weights was equal to 1, so that the possible value for each weight w_i was [0, 1]. We also set the granularity of each weight, such that a value of 10 meant we divided the range of [0, 1] into 10 equal parts, i.e., $w_i = 0.0, 0.1, \ldots, 0.9, 1.0$. The analysis reported here was performed using finer granularity of 20.

We developed a method to find the distance between two policies under a single reward function. Instead of simply counting how many times the user policy and optimal policy differed, we used the amount of reward each policy generated as a differencing metric. The Incremental Reward Difference method (IRD) compares two action paths by sequentially examining each time period and finding the difference in the total accumulated reward up to that point. For example, consider a simple reward function of $R = 0.4s_1 + 0.6s_2$, and we had one policy of (1,1,2,2) and another policy of (2,2,1,1). The total reward accumulated by both policies is 2.0, so it is important to have a metric that takes into account sequence order. In our method, the difference of the total reward accumulated after the first period is 0.2 (0.6-0.4), after the second it is 0.4 (1.2-0.8), after the third it is 0.2(1.6-1.4), and after the fourth it is 0 (2.0-2.0). Therefore, the difference between the policies is 0.8, which takes into account sequence and end result.

For each experiment observation, we store all of the reward functions that were closest to the expert and use a measure of central tendency as the point estimate of the true reward function. The standard method to measure distance between two n-tuple vectors is Euclidean distance. Standard cluster analysis uses the centroid as the averaging measure for a group of points, but this most likely will lead to an impossible reward function. Instead, we find the medoid (found in k-medoid cluster analysis), which is the element in the cluster that has the shortest average distance to every other point in the cluster.

3.2 Weights of Evidence Prediction Models

Rating the quality of generated rewards by IRL is directly dependent on the application. We have chosen to examine identity prediction in the sense that we could find someone's reward function and correlate identifying information by comparing against known data. We therefore desire the reward functions to group people into clusters based upon demographic similarities. In this section we discuss how we rate whether meaningful clusters are formed by analyzing experimental data.

Scoring models can be used to identify separation in the data and provide a means for prediction. Weights of Evidence (WOE) are used to convert data from an individual into a single score, and it is desired that scores are able to differentiate people. Scoring models predict a binary outcome, such as good (G) or bad (B), according to a vector of features. Given a feature vector x, the quatities of interest are $P(G|x)$ and $P(B|x)$. The score s is the log odds score, which can be broken into a population score s_{pop} and an information odds score s_{inf} by using Bayes Rule and the properties of logarithms, as shown in Eq. (4.7).

$$\begin{aligned} s(x) &= ln\frac{P(G|x)}{P(B|x)} \\ &= ln\frac{p(G)}{p(B)} + ln\frac{f(x|G)}{f(x|B)} \\ &= s_{pop} + s_{inf} \end{aligned} \tag{4.7}$$

The information odds score can be calculated from the data as the distribution that the feature vector takes a value given the person is good or bad. If each variable in the feature vector is conditionally indendent given the individual is good or bad, then the information score is given by Eq. (4.8).

$$s_{inf} = ln\frac{f(x_1|G)}{f(x_1|B)} + \cdots + ln\frac{f(x_n|G)}{f(x_n|B)} \tag{4.8}$$

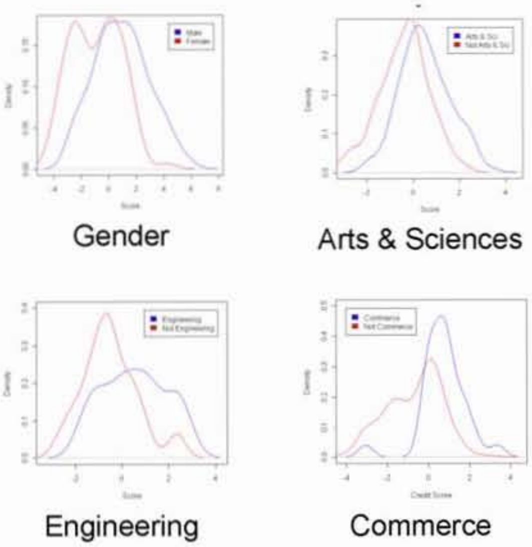
Each log odd in the information score is the WOE indicating *G* for that particular variable. The WOE is the log odds that the feature x_i takes on a particular value given the person is good, and can be directly calculated from the data. For instance, the value $f(x_1 = 0.1|G)$ is the proportion of the number of good people where $x_1 = 0.1$ over the total number of good people. This method requires a descritization of each variable x into multiple bins.

4 IMPLEMENTATION AND RESULTS

We discuss our findings with the caveat that the analysis was exploratory, and there was no previous work or principles that people grouped according to the tested demographic factors are expected to perform the task differently (e.g., there is no definitive theory that males utilize a different shopping strategy than females). However, IRL methods that find more correlation between demographic group and strategy are preferable, and this metric can be used in model selection when choosing between several predictive methods.

4.1 Results from WOE Scoring Models

For each IRL model, we developed credit scoring models for the gender and major variables. For the binary variable gender we calculated for male and not male, while for major we had to make three models for arts and not arts, engineering and not engineering, and commerce and not commerce. Each model was built using the 10 weights from the reward function as predictive features. The features were separated into bins based upon taking values of 0 through 0.3 and an additional bin for being greater than 0.3. Once the weights of evidence were calculated by determining the log odds that a feature took a particular value, scores were assigned to each trial based upon the reward function. Frequency plots showed the distribution of scores according to the group the individual belonged to. The frequency plots for the model are shown in Fig. 4.1.



Gender

Arts & Sciences

Engineering

Commerce

Fig. 4.1. Results from WOE scoring according to gender and school

The scoring models based upon WOE had the potential to perform the two tasks of identifying separation and predictive power. A benefit of the scoring analysis was the ability to visualize the data. If there was separation between the two demographic curves, we could have determined an optimal score threshold and tested for accuracy with training and testing data as in the regression analysis.

To investigate the predictive power of the scoring models, Receiver Operating Characteristic (ROC) curves were built to show the tradeoff between sensitivity and specificity with choosing a particular cutoff point. For instance, one may choose a cutoff such that a high percentage of males were correctly labeled as males, but in general there is a tradeoff associated with having an increased number of females that are incorrectly labeled as males. The former is the true positive rate, while the latter is the false positive rate, and a sample ROC curve is shown in Fig. 4.2 for gender.

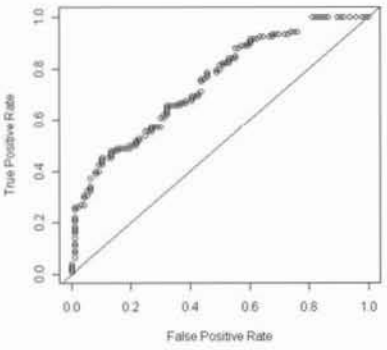


Fig. 4.2. ROC Curve for gender

An ideal ROC curve would be one that included the point (0,1) indicating it was possible to achieve a 100% true positive rate with a 0% false positive rate. Using this logic, curves are measured by the area under the curve (AUROC) where a value of 1 is considered the best while 0.5 is the worst. We show the AUROC score for the model in Table 4.1.

We developed decision rules to identify each participant and record the number of correct identifications. As an example, we found that classifying those with an engineering score above 1.05 as engineers and below as non-engineers yielded a 78% success rate. To further discriminate, we separated the non-engineers based upon the commerce score threshold of 0.92, and subsequently had a total success rate based on major of 66%.

Table 4.1. Performance metrics to predict user identity

	AUROC	% Correct
Gender	0.745	67.6%
Arts & Sci	0.718	68.6%
Engineering	0.716	77.9%
Commerce	0.810	86.2%
Total Major	N/A	66.2%

5 CONCLUSION

Inverse reinforcement learning has the capability to quantify human decision making through observation. This machine learning method can be used in many applications, including attribution. However, the literature does not verify that IRL captures real decision making. IRL has been tested to heuristically demonstrate its merit through controlled experimentation. In this work, IRL was used to analyze human behavior in experiments where the participants were not given any instruction regarding strategy. The most difficult aspect of performing IRL is developing an MDP that can capture the different strategies real participants use when performing a task. We provided a methodology that allows researchers to statistically test the ability of various IRL models to map reward functions to actions with respect to some application, in this case attribution. Models were compared based upon group significance testing and predictive power. These statistical methods can be used with any IRL scheme to test their usefulness with respect to attribution.

Without IRL, it is very difficult to understand the strategy that each participant used to perform shopping. At the most, the other study could only analyze the relative frequencies of the number of times each page type was visited, and would lose any information on the *order* that the participant viewed pages. People choose the next page as a direct result of the page they are currently viewing and overall preferences of the final goal and the required steps to achieve satisfaction. Most work on analyzing differences in humans choose to test the change in an observable variable, and it is rare to see analysis on the mathematical formulation of strategy.

The next step in assessing IRL as it pertains to capturing decision making is to analyze individual consistency. This work focused on analyzing differences between groups, whereas consistency analysis would investigate similarities of an individual over time. The primary goal of consistency analysis would be to show that an individual has an underlying strategy to perform tasks, and although actions may appear to be different across trials where the individual is placed in new situations, the strategy captured by the reward function would remain constant. This would serve to demonstrate that the user has a reward function and that IRL could recover the correct one. Users would need to be observed performing the same task multiple times, which would require additional testing than the data gathered for this experiment.

REFERENCES

[1] Abbeel, P.; Dolgov, D.; Ng, A.Y.; Thrun, S., "Apprenticeship learning for motion planning with application to parking lot navigation," Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, pp.1083-1090, 22-26 Sept. 2008

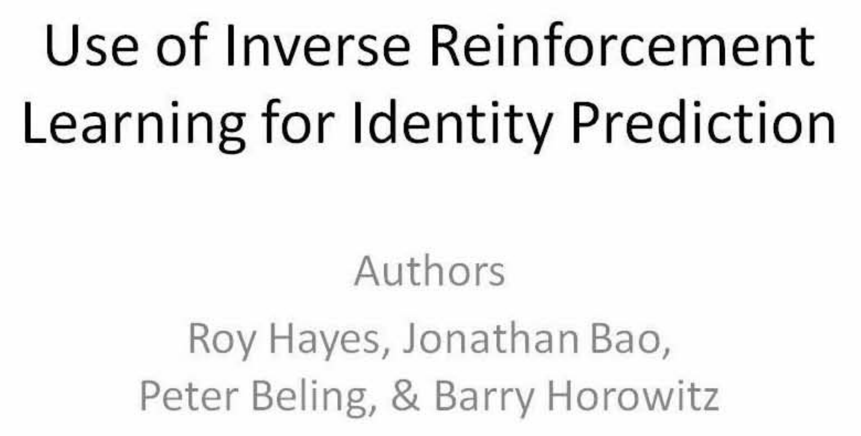
[2] Abbeel, P.; Ng, A., "Apprenticeship learning via inverse reinforcement learning," Proceedings of the twenty-first international conference on Machine learning, pp.1, 04-08 July 2004

[3] Choi, J. and Kim, K. 2009. Inverse reinforcement learning in partially observable environments. In Proceedings of the 21st international Jont Conference on Artifical intelligence (Pasadena, California, USA, July 11 - 17, 2009). H. Kitano, Ed. International Joint Conference On Artificial Intelligence. Morgan Kaufmann Publishers, San Francisco, CA, 1028-1033.

[4] Hosking, S., Liu, C., Bayly, M. The visual search patterns and hazard responses of experienced and inexperienced motorcycle riders, Accident Analysis & Prevention, Volume 42, Issue 1, January 2010, Pages 196-202

[5] Rust, J. Do people behave according to Bellman's principle of optimality? Stanford - Hoover Institution, 1992

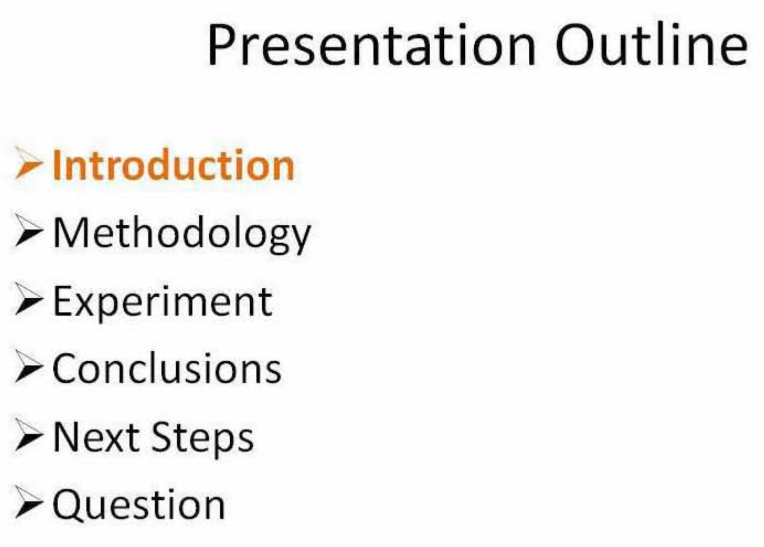
[6] Syed, U.; Schapire, R. E., "A game-theoretic approach to apprenticeship learning," Advances in Neural Information Processing Systems, 2008



Use of Inverse Reinforcement Learning for Identity Prediction

Authors

Roy Hayes, Jonathan Bao,
Peter Beling, & Barry Horowitz



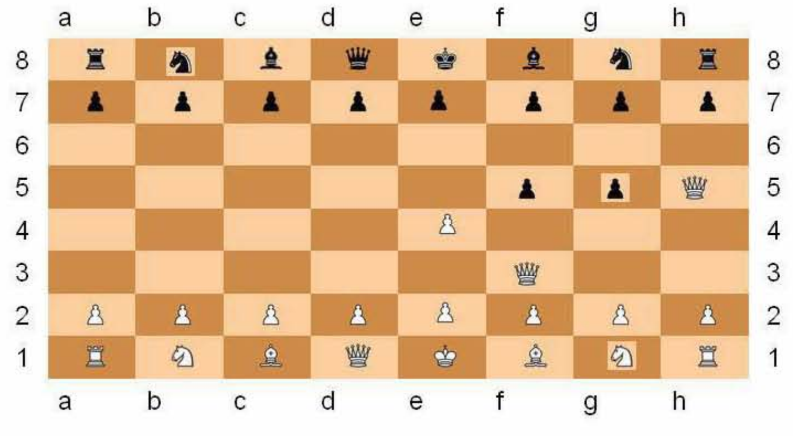
Presentation Outline

- ➢ **Introduction**
- ➢ Methodology
- ➢ Experiment
- ➢ Conclusions
- ➢ Next Steps
- ➢ Question

Markov Decision Processes

- S is set of all possible states at time t, given by S_t
- A is the set of all possible actions, given your in state S_t
- T is the Transition Probabilities, given your in S_t and chose a_t
- $\gamma \in [0,1)$ is a discount factor
- D is the initial-state distribution
- R is the Transition reward gained from taking action a_t at S_t

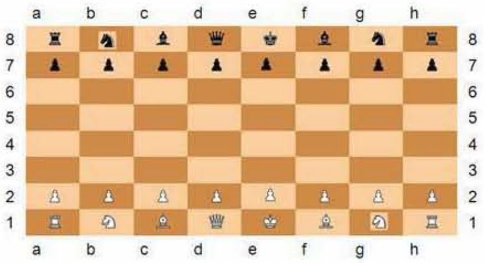
Inverse Reinforcement Learning (IRL)



1.e2 e4
2.d1 f3
3.f3 h5

Defining Reward function

What move to make?



$R(s,a) = \sum W_i * \phi_i$

What to eat?

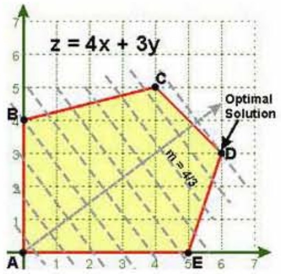
McDonald's
Panda Garden Express Chinese Cuisine
Panera BREAD
Ruby Tuesday
Olive Garden ITALIAN RESTAURANT

$R(s,a) = W_i*(M-(S_a+1))$

Selecting Optimal Reward Function

Linear Programming

- Only works on linear reward functions
- Computationally efficient



http://people.richland.edu/james/lecture/m116/systems/linear.png

Discretizing the space & perform exhaustive search

- Conceptually easy
- More robust
- Computationally expensive

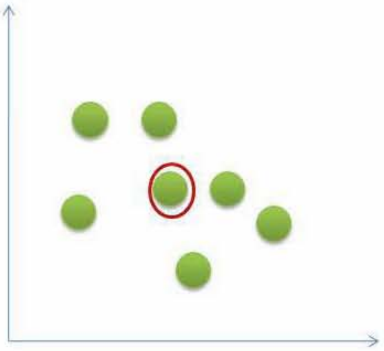
$R(s,a)_1 = 0.9* \phi_1 + 0.1* \phi_2$
$R(s,a)_2 = 0.8* \phi_1 + 0.2* \phi_2$
$R(s,a)_3 = 0.7* \phi_1 + 0.3* \phi_2$

Presentation Outline

- Introduction
- **Methodology**
- Experiment
- Conclusions
- Next Steps
- Question

Comparing Different Reward Functions

Medoid - Which is the element in the cluster that has the shortest average distance to ever other point in the cluster



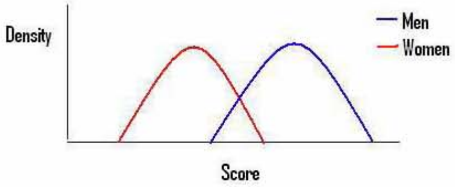
Predicting Identity

$$S(X) = ln(\frac{P(G|x)}{P(B|x)})$$

$$= ln\frac{P(G)}{P(B)} + ln\frac{f(x|G)}{f(x|B)}$$

$$= S_{pop} + S_{inf}$$

$$S_{inf} = ln(\frac{f(x_1|G)}{f(x_1|B)} + \cdots + ln(\frac{f(x_n|G)}{f(x_n|B)}$$



Presentation Outline

- ➢Introduction
- ➢Methodology
- ➢**Experiment**
- ➢Conclusions
- ➢Next Steps
- ➢Question

Experimental Setup

Goal : To determine attributes of an individual shopping on walmart.com

Procedure: Participant is given 4 portfolio of people to shop for and 5 mins per person to complete the shopping

Data Collected: The sequence of different types of pages viewed

Number of Participants :30

Save on school supplies, kids' apparel and more. Plus, Free Shipping* on school uniforms.

Put everything in its place for less. Shop drawers and carts from $15.

Walmart.com

Experiment's Markov Decision Process

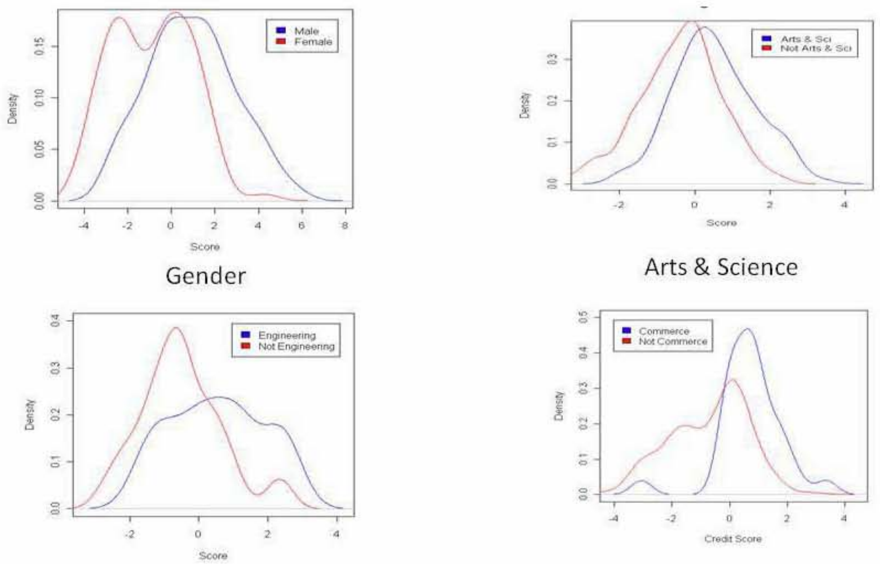
State – The current type of page you are viewing (e.g. store department page, item list, and checkout page)

Action – The next type of page selected

Transition – The transition probability is 100%

Reward Function – R(s,a) = W_i*(M-(S_a+1))

Experimental Results



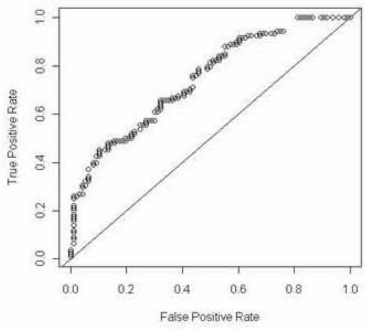
Gender

Arts & Science

Engineering

Commerce

Experimental Results



ROC Curve for gender

Performance metrics to predict user identity

	AUROC	**% Correct**
Gender	0.745	67.6%
Arts & Sci	0.718	68.6%
Engineering	0.716	77.9%
Commerce	0.810	86.2%
Total Major	N/A	66.2%

Conclusions

Inverse Reinforcement Learning can be set up in many different ways

Machine Learning methods can be applied to attribution

The statistical techniques presented are a good way to harness the predictive power of Inverse Reinforcement Learning

Next Steps

To determine consistency of an individual’s reward function

To examine Inverse Reinforcement Learning for training purposes

Any Questions?