Learning the Preferences of Bounded Agents

Owain Evans
University of Oxford

Andreas Stuhlmüller
Stanford University

Noah D. Goodman
Stanford University

Introduction

A range of work in applied machine learning, psychology, and social science involves inferring a person's preferences and beliefs from their choices or decisions. This includes work in economics on *Structural Estimation*, which has been used to infer beliefs about the rewards of education from observed work and education choices [1] and preferences for health outcomes from smoking behavior [2]. In machine learning, *Inverse Reinforcement Learning* has been applied to diverse planning and decision tasks to learn preferences and task-specific strategies [3, 4]. Large-scale systems in industry also learn preferences from behavior: for example, people's behavior on social networking sites is used to infer what movies, articles, and photos they will like [5].

Existing approaches to inferring human beliefs and preferences typically assume that human behavior is optimal up to unstructured "random noise" [6, 7]. However, human behavior may deviate from optimality in systematic ways. This can be due to biases such as time inconsistency and framing effects [8, 9] or due to planning or inference being a (perhaps resource-rational) approximation to optimality [10, 11]. If such deviations from optimality are not modeled, we risk mistaken inferences about preferences and beliefs. Consider someone who smokes every day while wishing to quit and regretting their behavior. A model that presupposes optimality up to noise will infer a preference for smoking. Similarly, consider someone who always eats at the same restaurant and never tries anywhere new. One explanation is that the person has strong prior beliefs about the (low) quality of other restaurants; however, another explanation is that the person fails to fully take into account the information-value of trying a new restaurant.

This paper explicitly models structured deviations from optimality when inferring preferences and beliefs. We use models of bounded and biased cognition as part of a generative model for human choices in decision problems, and infer preferences by inverting this model [12]. We are not primarily interested in investigating models of bounded cognition for their own sake, but instead in *using* such models to help accurately infer beliefs and preferences. We aim to highlight the potential value of models of bounded cognition as tools to aid inference in major applications in social science and in machine learning.

This paper focuses on inferring *preferences* (rather than beliefs) and bases inferences on observed choices in sequential decision problems (MDPs and POMDPs) [13]. We model the following types of agents:

1. **Time-inconsistent (hyperbolic-discounting) agents**, which are based on standard models of temptation, procrastination, and precommitment in psychology [8, 14, 15].
2. **Agents that use Monte Carlo approximations of expected utility**, which are related to sampling-based approximate inference and planning [16].
3. **Myopic agents**, which have short planning horizons.
4. **Bounded Value-of-Information agents**, which approximate value-of-information computations and have been used to investigate human performance in POMDPs [17].

We perform Bayesian inference over the space of such agents to jointly infer an agent's preferences, beliefs, and their bounds or biases from observed choices.

Computational Framework

This section describes the generative models of bounded and unbounded agents that we use to infer preferences from observed choices. We first describe the structure that is common to all of our agents. We then show that varying different parts of this structure lets us construct agents with different constraints. Finally, we describe how to use these models to infer preferences.

Agent structure

We treat an agent as a stochastic function that returns an action $a \in A$, denoted by $C\colon \varnothing \to A$ ("choice function"). The structure of the agent is given by a mutual recursion between C and an expected utility function, $\mathrm{EU}\colon A \to \mathbb{R}$. Both functions are parameterized by the agent's state, μ, which has a form that varies depending on agent type. To refer to the probability that C returns a, we write $C(a;\mu)$.

Agents choose actions in proportion to exponentiated expected utility (*softmax*):

$$C(a;\mu) \propto e^{\alpha \mathrm{EU}(a;\mu)} \tag{1}$$

The noise parameter α modulates between uniformly random choice ($\alpha = 0$) and perfect maximization (as $\alpha \to \infty$). Expected utility depends on current and future utility; future utility in turn depends on the agent's simulation of its future internal states and choices:

$$\mathrm{EU}\,(a;\mu) = U(\mu,a) + \mathop{\mathbb{E}}_{\substack{\mu' \sim M(\mu,a) \\ a' \sim C(h(\mu'))}} [\mathrm{EU}\,(a';\mu')] \tag{2}$$

To fully define an agent type, we have to specify how the agent computes immediate utility (U), how it predicts its future internal state for simulation of future choices and expected utility (M), and how it modifies the state when it gets passed to the choice function (h).

Agent types

The two optimal agents we consider as a basis for bounded agents are the standard ones used in MDP and POMDP settings (Figure 1). In the **MDP** setting, the agent's internal state corresponds

Agent type	**Type of μ**	$\boldsymbol{U(\mu,a)}$	$\boldsymbol{M(\mu,a)}$	$\boldsymbol{h(\mu)}$
MDP	$s \in S$	$U(s,a)$	$T(s,a)$	μ
MDP + myopia	$(s,d) \in S \times \mathbb{N}$	$U(s,a)$ if $d < k_m$ 0 otherwise	$(T(s,a), d+1)$	μ
MDP + discounting	$(s,d) \in S \times \mathbb{N}$	$\frac{1}{1+k_h d} U(s,a)$	$(T(s,a), d+1)$	(s,d) if naive $(s,0)$ if soph.
POMDP	$(p_s, o) \in P_s \times O$	$\mathop{\mathbb{E}}_{s \sim p_{s\mid o}} [U(s,a)]$	(p'_s, o') with $p'_s = p_{s\mid o,a}$ $s' \sim p'_s$ $o' \sim p_{o\mid s'}$	μ
POMDP + bounded VOI	$(p_s, o, d) \in P_s \times O \times \mathbb{N}$	$\mathop{\mathbb{E}}_{s \sim p_{s\mid o}} [U(s,a)]$	$(p'_s, o', d+1)$ with $p'_s = \begin{cases} p_{s\mid o,a} \text{ if } d < k_{voi} \\ p_{s\mid a} \text{ otherwise} \end{cases}$ $s' \sim p'_s$ $o' \sim p_{o\mid s'}$	μ

Figure 1: Different ways of filling in the components of Equation 2 result in different kinds of bounded and unbounded agents. μ refers to the agent's internal state, U to utility, M to the function used by the agent for a single-step update of its internal state when simulating the future, and h to a (potential) modifier of internal state for the purpose of simulating future choices. Other combinations, such as POMDP + bounded VOI + discounting, follow analogously.

to a world state $s \in S$, immediate utility is computed using a utility function on state-action pairs, $U\colon S \times A \to \mathbb{R}$, and transitions on the agent's internal state correspond to simulated world state transitions, $T\colon S \times A \to S$. In the **POMDP** setting, the agent maintains a distribution p_s on world states together with a current observation o, calculates immediate utility by averaging over likely world states, and updates its internal state by updating p_s on the current observation and action, and simulating a next observation. (To simplify notation, we have folded the transition function into p_s.)

The bounded agents additionally keep track of a delay d that reflects how far into the future a given iteration of the planning recursion happens. The **myopic** agent simply assumes that all utilities are 0 when the delay exceeds some constant k_m. This corresponds to planning with k_m-step lookahead. The agent with **hyperbolic discounting** discounts future utility by a multiplicative factor $\frac{1}{1+k_h d}$, where $k_h \geq 0$ controls the discount rate. This may influence both choice and utility prediction ("Naive agent"), or only utility prediction ("Sophisticated agent") [15]. The agent with **bounded value-of-information** only simulates belief updates when $d < k_{voi}$, and (mistakenly) assumes that any subsequent observations will not result in belief updates, thus exhibiting a form of time inconsistency. Finally, we can turn any of these agents into an approximate **Monte Carlo** agent by replacing the expectation in Equation 2 with an average over a finite number of samples.

Preference inference

By combining all of the agents above, we create a large space of possible agents with many parameters to be learned. In our examples (next section), we perform inference on subsets of this space. Here we illustrate inference for Example 3, where we infer the latent parameters for a "POMDP + Bounded VOI" agent from an observed sequence of actions. The parameters are a utility function U, prior p_s, VOI bound k_{voi}, and noise parameter α. Note that, when k_{voi} is greater than the total number of time steps, the agent is equivalent to an optimal POMDP agent. An agent is defined by a tuple $\theta := (p_s, U, k_{voi}, \alpha)$, and we perform inference over this space given observed actions. The posterior joint distribution on agents conditioned on action sequence $a_{0:T}$ is:

$$p(\theta|a_{0:T}) \propto p(a_{0:T}|\theta)p(\theta) \tag{3}$$

The likelihood function $p(a_{0:T}|\theta)$ is given by the multi-step generalization of the choice and expected utility functions corresponding to θ. For the prior $p(\theta)$, we use independent uniform priors on bounded intervals for each of the components. In the following, "the model" refers to this generative process that first samples an agent (including utility function), then choices given the agent. We implemented all agents as probabilistic programs. See Figure 4 for an illustration.

Examples of Preference Inferences

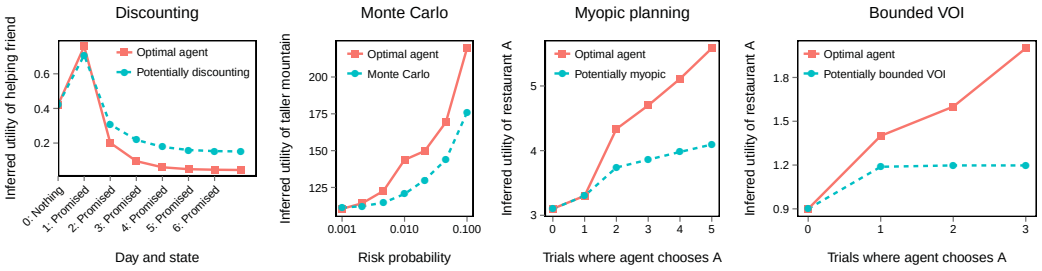
We now contrast preference inferences using models that assume optimality with inferences from models that allow for bounded agents. We exhibit decision problems where we expect the assumption of optimality to be particularly inappropriate. These problems may appear overly simple and hence unrealistic. However, if assuming optimality is problematic for very simple problems (with few states and parameters), then it is likely to be problematic for complex problems as well; complex problems are likely to "contain" simple problems in ways that preserve their problematic features.

Examples 1 and 2 below are MDPs and contrast optimality with time-inconsistent and Monte-Carlo agents respectively. Example 3 (below) involves POMDPs and compares optimality with myopic planning and Bounded VOI. All "bounded" models (see dotted curves in Figure 2) include the optimal model as a special case, hence inference implicitly involves model selection. This means that our approach does not assume that bounded models provide a superior fit to agent behavior.

1. Procrastination (hyperbolic discounting)

Consider the following decision problem on which some time-inconsistent agents will "procrastinate" [cf. 18, 19]:

> A friend is looking for comments on a paper. You know your comments would improve their paper and you assign positive utility to this outcome. However, writing



(a) (b) (c) (d)

Figure 2: Examples of inferences about utilities for optimal and bounded agents

> the comments has negative utility to you because it is tedious and will take a whole day. The paper will be submitted in T days and comments are more helpful earlier.

There are two decisions to make. First, you decide whether to promise your friend that you will offer prompt comments, i.e., move from "do nothing" to "promise" node in Figure 3. After you promise, they send you the paper and the next day you decide whether to "do work" (which results in the "help friend" outcome) or to stay in the "promise" state. There is no cost to staying in "do Nothing", but there is a tiny cost of $-\epsilon$ for every day in "promise". Doing the work has a one-time cost of -1 and, after you have done the work, you receive $+R$ for every day until T.

Suppose the agent moves to "promise" but never moves to "help friend". This results in an outcome that is worse than staying at "do nothing" the entire time. We call this *procrastinating*. The optimal agent (without softmax noise) never procrastinates. It either does the work without unnecessary delay or does nothing[1]. Time-inconsistent agents can procrastinate depending on R and the discount rate k_h. The Naive discounting agent hallucinates that it will "do work" after first moving to "promise", but once actually at "promise", it delays the work indefinitely.

We set $T = 8$ and condition on the observation that the agent procrastinates, i.e. moves directly to "promise" and then stays there for the remaining 7 days. The goal is to infer R (the utility of helping the friend). We compare the "optimal" model (no time-inconsistency) to a "potentially discounting" model that includes both Naive discounting and optimal planning. Figure 2a shows that under both models, the expected posterior value of R is low. However, the value for the discounting model is higher, as it can explain away the agent's not helping by a higher discount rate k_h. Additionally (not shown), we infer high noise when we assume optimality, since the optimal agent only intentionally endures the $-\epsilon$ cost of moving to "promise" if it will then do the work. Since the agent did not do the work, it must have high noise if it is (otherwise) optimal.

2. Neglect of low-probability events (Monte Carlo approximation)

Consider the following problem:

> John is hiking and has to choose between climbing up to the *Tall* peak or the *Short* peak. The Tall peak is more spectacular, but comes with a small probability p_d of disaster (e.g. death or injury). We assume John has no uncertainty about his utilities for Tall and Short, and that John knows p_d.

We aim to infer John's utility for climbing the Tall peak, U_t, relative to the cost of disaster. We compare an "optimal" model (which solves the MDP exactly) with a Monte Carlo model ("MC") where the agent samples N times from the state transition function to approximate an action's expected utility. We set a low prior on U_t being close in magnitude to the cost of disaster. The MC model has a broad prior on N and includes planning behavior indistinguishable from optimal as a special case. We condition on the observation that John moves directly to the Tall peak. Figure 2b shows the posterior mean for U_t as a function of the probability of disaster p_d. For both models, as p_d

[1] It does the work if $R(T-2) > -(1+\epsilon)$.

increases we infer a higher U_t (as Tall is chosen despite increasing risk of disaster). The MC model infers consistently lower values for U_t. It partially explains away John's choice by positing small N, which will sometimes lead to overestimates of the expected value of climbing Tall.

3. Failure to explore (myopic and Bounded VOI agents)

Human performance on bandit problems has been studied extensively [20]. For example, Zhang and Yu [17] show that human performance in a low-dimensional bandit problem is sub-optimal and is better captured by the Knowledge-Gradient algorithm than by optimal play. The Knowledge-Gradient algorithm is analogous to our Bounded VOI agent with one level of lookahead. This work suggests that assuming optimality may lead to inaccurate inferences even for low-dimensional bandit problems. For higher-dimensional problems, optimal play is intractable and so any realistic agent will use approximations. Consider the following bandit-style problem:

> You get out of a meeting and choose between two nearby restaurants, A and B. You know the utility U_A exactly (e.g. A is a chain) but are uncertain about U_B. For the next T months, you have meetings in the same place and will face the same choice between A and B.

The goal is to infer U_A, the agent's utility for A. We run inference repeatedly for increasing values of T (i.e. we increase the expected value of exploration). For each T, we condition on the observation that the agent chooses restaurant A for each of the T months. That is, the agent never explores, even as T grows.

The "optimal" inference model assumes that the agent solves the POMDP perfectly given their prior on U_B. On the "potentially myopic" model, the agent can either plan optimally or else plan myopically with the time horizon set to one trial. As T increases, exploration becomes more valuable to the optimal agent. Assuming optimality therefore leads to progressively higher inferred values for U_A. In contrast, a myopic agent will not explore more as T increases, resulting in a flatter curve in Figure 2c.

The Bounded VOI model with $k_{voi} > 0$ behaves optimally for this bandit-style problem. But consider the following elaboration:

> You are choosing somewhere to eat, but there are no restaurants nearby. You can head to restaurant A (which again has known utility U_A) or try somewhere new. Most restaurants are worse than A, but some might be better. Before trying a restaurant you ask for advice in two distinct steps. First you ask which neighborhood has the best restaurants and later you ask a local of that neighborhood which restaurant is best. When you try a restaurant you learn its utility. There are no costs to getting advice and your distance traveled is not a constraint. As above, this choice is repeated for each of T months.

The goal for inference is again to infer U_A. We fix the problem parameters[2] and vary T. The observation we condition on is the agent choosing A every time (same as before). The Bounded VOI agent with $k_{voi} = 1$ deviates from optimal on this problem. This agent models itself as updating on the first question ("which neighborhood?") but neither on the second question nor the direct observation of restaurant quality. It can fail to explore even when T is high (without having a strong preference for A). The Bounded VOI model only matches optimal behavior when $k_{voi} > 2$. Figure 2d compares a "potentially bounded VOI" model (which includes the optimal model as a special case) with the optimal model.

[2]We assume two neighborhoods with two restaurants each. The agent has a prior over the utility of unknown restaurants such that most are bad. Our inference prior on U_A is such that we think it likely that the agent expects the best unknown restaurant to be better than A.

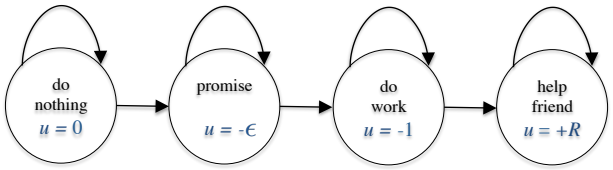


Figure 3: Transition graph for Example 1 (procrastination). Nodes represent states; u is the utility the agent receives for every day spent at the state. Arcs indicate possible (uni-directional) deterministic transitions between states. The agent takes $T = 8$ actions, one for each day. We condition on observing the agent moving directly to "Promise" and staying there for the remaining 7 days.

```
var agent = function(state, delay, timeLeft){
  return Marginal(function(){
    var action = uniformDraw(actions)
    var eu = expUtility(state, action, delay, timeLeft)
    factor(alpha * eu)
    return action
  })
}

var expUtility = function(state, action, delay, timeLeft){
  var u = discountedUtility(state, action, delay, K)
  if (timeLeft == 1){
    return u
  } else {
    return u + expectation(INFER_EU(function(){
      var nextState = transition(state, action)
      var nextAction = sample(agent(nextState, delay+1, timeLeft-1))
      return expUtility(nextState, nextAction, delay+1, timeLeft-1)
    }))
  }
}

var simulate = function(startState, totalTime){
  var sampleStateSequence = function(state, timeLeft, history){
    if (timeLeft==0){
      return history
    } else {
      var delay = 0
      var action = sample(agent(state, delay, PLANNING_HORIZON))
      var nextState = transition(state, action)
      return sampleStateSequence(nextState, timeLeft-1, update(history, nextState))
    }
  }
  return Marginal(function(){
    return sampleStateSequence(startState, totalTime, initHistory(startState))
  })
}
```

Figure 4: Implementation of a generative model for agents in the MDP setting. The language is WebPPL (with minor implementation details omitted) [21]. Note the mutual recursion between **`agent`** and **`expUtility`**: the agent's reasoning about future expected utility includes a (potentially biased) model of its own decision-making. The function `Marginal` computes exact distributions over the output of its function argument. The `factor` statement implements soft conditioning—it is used here for softmax "planning-as-inference" [22]. To generate agent behavior, we specify a decision problem by providing implementations for `transition` and `utility`. We then call `simulate(startState, totalTime)`. For exact planning, we set `INFER_EU` to `Marginal`. For the Monte Carlo agent, we set `INFER_EU` to a function that computes sampling-based approximations. If the constant `K` is set to zero, the agent does not discount (and so is optimal); otherwise, the agent performs Naive hyperbolic discounting.

References

[1] Victor Aguirregabiria and Pedro Mira. Dynamic discrete choice structural models: A survey. *Journal of Econometrics*, 156(1):38–67, 2010.

[2] Michael Darden et al. Smoking, expectations, and health: a dynamic stochastic model of lifetime smoking behavior. *Health Econometrics and Data Group-University of York, Working Paper*, 10:28, 2010.

[3] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, page 1. ACM, 2004.

[4] Stefano Ermon, Yexiang Xue, Russell Toth, Bistra Dilkina, Richard Bernstein, Theodoros Damoulas, Patrick Clark, Steve DeGloria, Andrew Mude, Christopher Barrett, et al. Learning large-scale dynamic discrete choice models of spatio-temporal preferences with application to migratory pastoralism in east africa. In *Meeting Abstract*, 2014.

[5] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems: an Introduction*. Cambridge University Press, 2010.

[6] Dongho Kim, Catherine Breslin, Pirros Tsiakoulis, Milica Gašic, Matthew Henderson, and Steve Young. Inverse reinforcement learning for micro-turn management. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 328–332. International Speech and Communication Association, 2014.

[7] Jiangchuan Zheng, Siyuan Liu, and Lionel M Ni. Robust bayesian inverse reinforcement learning with sparse behavior noise. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[8] George Ainslie. *Breakdown of Will*. Cambridge University Press, 2001.

[9] Daniel Kahneman and Amos Tversky. Prospect theory: an analysis of decision under risk. *Econometrica: Journal of the Econometric Society*, pages 263–291, 1979.

[10] Edward Vul, Noah Goodman, Thomas L Griffiths, and Joshua B Tenenbaum. One and done? optimal decisions from very few samples. *Cognitive Science*, 38(4):599–637, 2014.

[11] Thomas L Griffiths, Falk Lieder, and Noah D Goodman. Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic. *Topics in Cognitive Science*, 7(2):217–229, 2015.

[12] Joshua B Tenenbaum, Charles Kemp, Thomas L Griffiths, and Noah D Goodman. How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022):1279–1285, 2011.

[13] Alessandro Panella and Piotr Gmytrasiewicz. Learning policies of agents in partially observable domains using bayesian nonparametric methods. In *AAMAS Workshop on Multiagent Sequential Decision Making Under Uncertainty*, 2014.

[14] David Laibson. Golden eggs and hyperbolic discounting. *The Quarterly Journal of Economics*, pages 443–477, 1997.

[15] Owain Evans, Andreas Stuhlmüller, and Noah D Goodman. Learning the preferences of ignorant, inconsistent agents. In press.

[16] Christian Robert and George Casella. *Monte Carlo Statistical Methods*. Springer Science & Business Media, 2013.

[17] Shunan Zhang and J Yu Angela. Forgetful bayes and myopic planning: Human learning and decision-making in a bandit setting. In *Advances in Neural Information Processing Systems*, pages 2607–2615, 2013.

[18] Jon Kleinberg and Sigal Oren. Time-inconsistent planning: a computational problem in behavioral economics. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, pages 547–564. ACM, 2014.

[19] George A Akerlof. Procrastination and obedience. *The American Economic Review*, pages 1–19, 1991.

[20] Mark Steyvers, Michael D Lee, and Eric-Jan Wagenmakers. A bayesian analysis of human decision-making on bandit problems. *Journal of Mathematical Psychology*, 53(3):168–179, 2009.

[21] Noah D Goodman and Andreas Stuhlmüller. *The Design and Implementation of Probabilistic Programming Languages*. 2014.

[22] Matthew Botvinick and Marc Toussaint. Planning as inference. *Trends in Cognitive Sciences*, 16(10):485–488, 2012.

[23] Richard A Posner. *Catastrophe: Risk and Response*. Oxford University Press, 2004.