Every algorithmic decision-maker incentivizes people to act in certain ways to receive better decisions. These incentives can dramatically influence subjects’ behaviors and lives, and it is important that both decision-makers and decision-recipients have clarity on which actions are incentivized by the chosen model.

Why the incentives provided by algorithms matter:
- They are legally regulated (e.g. adverse action notices in credit scoring).
- They empower individuals to have control and agency over their own outcomes (e.g. [2]).
- Whether we study them or not, all algorithms already incentivize behaviors, and are having unobserved consequences for decision-makers and decision-recipients in the real world.

In this work, we propose a novel framework for analyzing algorithmic incentives through the lens of Markov decision processes (MDPs).

At a high level, we propose that to properly understand how an individual is incentivized to act, we must first define the actions available to an individual, and their effects. Then, the individual is incentivized to take whichever action will modify their current state such that, after executing a sequence of additional actions, they will reach a final state that maximizes their received decision.

We show, using this framework, that many traditional interpretability tools (e.g. LIME[1], input gradients) can provide poor advice policies when the decision-making model is non-linear.

Our key contribution is a method for identifying approximately optimal algorithmic incentives, by using planning algorithms like MCTS to solve the agency MDP. Furthermore, our method is model-independent and requires only query access to the decision-making model.

We show in experiments that this method outperforms local approximations’ advice in practical settings, including an online FICO scoring API, and a random-forest-based violent recidivism predictor.

### Experiments

We applied our incentive-evaluation framework to two decision-settings: pretrial risk assessment (based on the COMPAS dataset), and credit scoring (by querying FICO’s online credit score calculator).

We also trained a double deep Q-network on the agency MDP, but found that in both settings the network generally failed to learn a meaningful advice policy (not even equaling the greedy policy), and so we have excluded those results.

We compare these incentives to the greedy policy (Eq. 2), which maximizes the decision immediately after the current action, and to a random policy.

### Problems with Local Approximations as Advice Policies

Local-approximation-based advice can be dangerously wrong. Consider the example in Figure 1, in which a locally-improving policy would trap the individual at a local maximum and never achieve the better outcome that was available to them. Moreover, local advice may still be sub-optimal when the decision function is monotonic as is the case in Figure 2.

### Framework

Consider an individual \( s \in S \), defined as a feature vector. Individual \( s \) wants to maximize the outcome of positive-definite decision function \( D(s) \in \mathbb{R}^n \).

By taking an action \( a \in A \), individual \( s \) may change their state. Individual \( s \)'s next state is defined by sampling from transition model \( T \) where \( s' \sim T(s,a) \).

We combine \( S, A, \) and \( T \) together to form a Markov Decision Process (MDP). We specify a terminal function \( \text{end}(s) \) that determines whether the sequence has ended, and define the reward function \( \mathcal{R} \):

\[
\mathcal{R}(s) = \begin{cases} D(s) & \text{if end}(s) \\text{true} \\ 0 & \text{otherwise} \end{cases}
\]

An advice policy \( \pi \in \Pi \) recommends a certain action \( a = \pi(s) \) for each state \( s \) an individual may encounter.

For example, a locally-optimizing greedy policy \( \pi \) chooses actions based only on maximizing the immediate improvement in the received decision:

\[
\pi_{\text{local}}(s) = \arg \max_{a \in A} \mathbb{E}_{s' \sim T(s,a)} [D(s')] \tag{2}
\]

We say an action is incentivized if it is recommended by an optimal advice policy \( \pi^* \). More specifically, an individual with state \( s \) is incentivized to execute action \( a^* \) if that action will maximally improve their eventual expected decision, more than any alternative action:

\[
a^* = \max_{a \in A} \mathbb{E}_{s' \sim T(s,a)} [D(s')] - \pi^*(s) \tag{3}
\]

where \( H_t(s) \) is the distribution of end-states resulting from “rolling out” \( s \) starting at state \( s \).

We can approximate this optimal advice policy by leveraging planning algorithms such as reinforcement learning.

### Problems with Local Approximations as Advice Policies

Local-approximation-based advice can be dangerously wrong. Consider the example in Figure 1, in which a locally-improving policy would trap the individual at a local maximum and never achieve the better outcome that was available to them. Moreover, local advice may still be sub-optimal when the decision function is monotonic as is the case in Figure 2.

### High-Level Overview

Extracting Incentives From Black-Box Decisions

Yonaday Shavit\(^1\), William S. Moses\(^2\)

Harvard SEAS\(^1\), MIT CSAIL\(^2\)
yonaday@harvard.edu\(^1\), wmoses@mit.edu\(^2\)

---

**Acknowledgements & References**

We would like to thank Christian Bueno for her early insights into the challenges of incentivizing minorities, and to Cynthia Durov, Michael P. Kim, Sahil Vijaykhera, Botana Curts, and Charles Leiserson for their helpful feedback and advice.

William S. Moses was supported in part by a DOE Computational Sciences Graduate Fellowship DE-SC0019233. This research was supported in part by NSF Grant 1535044 and 1535044, in part by LANA grant 531711, and in part by BHN grant W1771646.
