A Data-Reweighting Approach to the Inverse Markov Decision Process with Unknown Transition Probabilities

Connor Thompson[1], Jacob Miller[2], Bo Lin[3,4], Brian Hartman[1], Timothy C.Y. Chan[3], and Nathan Sandholtz[1]

[1] Brigham Young University, Provo, USA
[2] Sacramento Kings, Sacramento, USA
[3] University of Toronto, Toronto, CAN
[4] National University of Singapore, Singapore, SGP

Abstract. We introduce a data-driven approach to the inverse Markov decision process (MDP) in which the reward function is known but the transition probabilities believed by the decision-maker are unknown. Given a dataset of observed state–action–next-state transitions, we learn a reweighting of the data such that, when the forward MDP is solved using the reweighted data, the resulting optimal actions align with the observed actions in the dataset. The estimated weights quantify the influence of historical transitions on the decision-maker's perceived transition dynamics, and indirectly yield an implied transition function under which the observed actions are approximately optimal. This formulation is inherently interpretable: the weights reveal which experiences the decision-maker effectively emphasizes or discounts, and as such can be used to identify systematic patterns in suboptimal behavior. We propose an inexpensive algorithm to sample from the approximate solution space to the inverse problem, yielding a plausible range of estimates on the weights and the corresponding implied transition function. We demonstrate the approach on fourth-down decision data from the National Football League. The resulting inferred weights suggest that coaches overweight negative outcomes relative to positive ones when making fourth down decisions.

Keywords: Inverse Markov decision process · Data reweighting · Transition probability inference · Behavioral modeling

1 INTRODUCTION

The Markov Decision Process (MDP) is a widely used model for sequential decision-making. An MDP is defined by a state space $\mathcal{S}$, an action set $\mathcal{A}$, a transition function $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, describing the probability of moving from state $s \in \mathcal{S}$ to $s' \in \mathcal{S}$ after taking action $a \in \mathcal{A}$, and a reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, specifying the immediate utility of that transition. A policy $\pi : S \to \Delta(A)$, prescribes behavior in the environment by mapping each state to a distribution over actions, $\Delta(A)$.

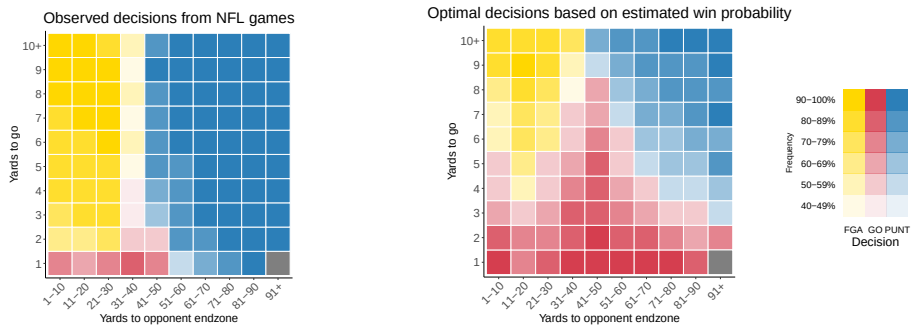
Given p and r, an optimal policy π^* can be determined that maximizes expected long-term reward. We refer to this as the *forward problem*. In the *inverse problem*, a policy $\tilde{\pi}$ representing a decision-maker's observed behavior is given (or can be estimated from data), and the goal is to infer parameters of the MDP that make this policy optimal or approximately optimal. When $\tilde{\pi}$ reflects decisions made under perceived optimality, the inferred MDP can be interpreted as the decision-maker's internal model of the environment that makes their behavior appear rational, even if it diverges from the MDP's true dynamics.

In the existing literature, the inverse problem is almost exclusively formulated by treating the reward function r as the estimand and the transition function p as known. The goal is to estimate an *implied* reward function $\tilde{r}$, such that it makes the observed policy minimally suboptimal conditional on p. A common formulation of this problem is known as Inverse Reinforecement Learning (IRL) [15, 11, 1, 21]. As shown by [2] and applied to MDPs by [7], this problem can be formulated as a finite-dimensional linear program by enforcing complementary-slackness conditions between the primal and dual forms of the forward MDP. In many settings, this is a sensible approach; the reward function guiding the agent is latent and can reasonably be treated as the estimand in the inverse problem. However, in some decision contexts the rewards are explicitly defined by the rules of the environment. In such cases, attributing differences in behavior between observed and optimal actions to alternative reward functions mischaracterizes the problem.

Consider the fourth down decision in the National Football League (NFL). As described in [16], in American football, each team has four downs (i.e., chances) to advance the football 10 yards. On fourth down, coaches face a decision: they can go for it, attempting to gain the remainder of the 10 yards and thus a first down, or they can kick the ball either by attempting a field goal (when the team is close enough to the opponent's end zone) or by punting (when the team is farther away). Punting concedes possession of the ball, but it does so by putting the other team in a worse field position, typically deeper in their own half of the field. Attempting a field goal can yield an immediate three points if successful, but a miss gives the opposing team possession of the ball from the location of the kick.

Numerous analyses have shown that coaches have historically chosen systematically suboptimal actions in fourth down situations, as illustrated in Figure 1 [14, 19, 13]. Historically, coaches have behaved far more conservatively than statistical decision models recommend—punting or kicking field goals in situations where going for it yields a higher expected value. To explain this seemingly irrational behavior, existing inverse MDP methods could be used to estimate an implied reward function. However, the resulting inferences could produce misleading conclusions since rewards in football are fixed and known (e.g., six points for a touchdown, three points for a field goal, etc.).

In contrast, other aspects of the decision process may be better suited to explain coach decisions. For example, it is possible that coaches are systematically risk averse [19], or do not optimize strictly based on win probability [16, 14].



(a) **(b)**

Fig. 1: (a) Most frequent fourth down decisions over the 2014-2022 NFL seasons with respect to yardline (x-axis) and yards to first down (y-axis). Decisions are represented by fill color: yellow denotes field goal attempt (FGA), red denotes go for it (GO), and blue denotes punt (PUNT). The amount of color saturation corresponds to the frequency of the decision—darker colors represent higher frequencies. **(b)** Most frequent fourth down decision prescriptions based on [3] with respect to yardline and yards to first down. Here, color saturation corresponds to the frequency with which the relative majority decision is estimated to be optimal. This figure is inspired by similar figures in [4] and [3] and is identical to that of Figure 1 in [16].

Alternatively, the transition probabilities between states (e.g., the probability of converting on fourth down) are genuinely unknown, and it is plausible that coaches do not estimate them accurately. In other words, the decision-maker's internal model of the transition dynamics—their implied transition function $\tilde{p}$—may differ substantially from the true dynamics. In this work, we treat the decision-maker's implied transition function as the estimand.

The idea of estimating the implied transition function has received little attention in the literature. To our knowledge, only one prior work, [8], adopts this formulation. They estimate $\tilde{p}$ in a model-based manner, analogous to how [7] estimates an implied reward function. They construct a non-convex bilinear program to determine transition probabilities that make a given policy optimal.

In this work we provide a data-based perspective on the same problem. We observe a dataset $\mathcal{D} = \{(s_i, a_i, s_i')\}_{i=1}^n$ of state–action–next-state transitions, of which the observed actions do not align with the empirical optimal policy $\hat{\pi}^*$.[5] The reward function r is also known. Our key idea is to systematically reweight the data so that, when the forward problem is solved under the reweighted dataset, the resulting optimal actions align with the observed actions. To accomplish this, we introduce nonnegative weights $w_1, \ldots, w_n$ on the rows of the dataset $\mathcal{D}$ and learn these weights such that the mismatch between the MDP's prescribed actions and the observed actions a_i is minimized. This indirectly

[5] We define the empirical optimal policy as the policy that results from solving the MDP under a data-based estimate of the transition function.

yields an implied transition function $\tilde{p}$ under which the observed behavior appears optimal.

The key advantage of this approach is interpretability. Each weight w_i corresponds to how much influence a specific past transition (s_i, a_i, s_i') has on the decision-maker's perceived transition dynamics, hence the learned weights provide a representation of which experiences the agent is effectively emphasizing or discounting. This in turn can shed light on cognitive biases of the decision-maker. For example, in our application to fourth down decisions in the NFL, we find that failed fourth-down attempts receive substantially higher weight than successful ones, suggesting that coaches tend to give more influence to negative outcomes when forming their beliefs about fourth down situations.

Our contributions are as follows:

1) We develop a novel data-driven framework for inferring an implied transition function in an inverse MDP setting. The algorithm we propose to approximate a solution to this problem is inexpensive and the stochastic nature of it enables exploration of the uncertainty in the solution space.
2) The data reweighting framework we propose is naturally interpretable. Each learned weight w_i quantifies the influence of the corresponding recorded $\{s_i, a_i, s_i'\}$ tuple on the inferred transition function, revealing which experiences the decision-maker effectively emphasizes or discounts. This can allow researchers to diagnose patterns in suboptimal decision-making and relate them to psychologically meaningful processes.

The remainder of the paper is organized as follows. In Section 2 we define the forward problem from two separate perspectives: model-based and data-driven. Section 3 defines and contrasts the corresponding inverse problems. In Section 4 we present an algorithm to approximate a solution to the data-driven inverse problem. Section 5 illustrates the method on a simple cliff walk example. In Section 6 we apply the method to the fourth-down decision problem using a dataset of NFL play-by-play data from 2014 to 2022. Section 7 provides discussion, including limitations and directions of future work.

2 FORWARD PROBLEM

To contextualize the inverse MDP problem we introduce in this paper, we first distinguish between two formulations of the forward MDP problem: a model-based version and a data-based version.

2.1 Model-Based Forward Problem

The *model-based* problem assumes the transition and reward functions are known and solves for an optimal policy under these primitives. Given stationary transition function p, reward function r, and policy π, the *value function* at state

s is defined as the expected long-term (possibly discounted) sum of rewards beginning in state s and following π thereafter:

$$v_\pi(s;p) = \mathbb{E}_{\pi,p}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right], \tag{1}$$

where $\gamma \in (0,1]$ is a discount factor. Under these conditions, the value function satisfies the Bellman equation:

$$v_\pi(s;p) = \sum_{a\in\mathcal{A}} \pi(a \mid s) \sum_{s'\in\mathcal{S}} p(s' \mid s,a)\left[r(s,a,s') + \gamma v_\pi(s';p)\right]. \tag{2}$$

Given p and r, an optimal policy can be found by solving the following linear program:

$$\begin{aligned} \min_{v} \quad & \sum_{s\in\mathcal{S}} v(s;p) \\ \text{s.t.} \quad & v(s;p) \geq \sum_{s'\in\mathcal{S}} p(s' \mid s,a)\left[r(s,a,s') + \gamma v(s';p)\right], \\ & \forall s \in \mathcal{S},\ a \in \mathcal{A}. \end{aligned} \tag{3}$$

The solution to (3) is the optimal value function v^*, from which an optimal policy π^* can be derived. This is the classical formulation of the forward problem typically associated with solving an MDP [12].

2.2 Data-Driven Forward Problem

In the *data-driven* formulation of the forward problem, we assume the reward function is known but the transition function is not. Instead, we observe trajectories from a decision-maker's interactions with the environment from which the transition function must be estimated. Let $\mathcal{D}$ denote this observed dataset, which contains n (s_i, a_i, s_i')-tuples generated from an MDP with known reward function r.

The first step in the data-driven forward problem is to estimate the transition function, which, for generality, we assume follows a parametric form $p(s' \mid s,a;\boldsymbol{\theta})$, $\boldsymbol{\theta} \in \Theta$. The transition parameters are estimated by solving

$$\hat{\boldsymbol{\theta}}_{\mathbf{w}} = \operatorname*{arg\,min}_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \sum_{i=1}^{n} w_i\, \ell_{\boldsymbol{\theta}}[s_i',\, p(s_i' \mid s_i, a_i;\boldsymbol{\theta})], \tag{4}$$

where $\mathbf{w} = (w_1,\ldots,w_n)$ is a vector of nonnegative weights and $\ell_{\boldsymbol{\theta}}(\cdot)$ is an appropriate loss function (e.g., cross-entropy for discrete next-state outcomes).[6] After

[6] Although the weight vector $\mathbf{w}$ is typically set to $\mathbf{1}_n$ in the forward problem, we include it here for generality. These weights take on substantive meaning in the inverse problem, where each w_i represents the relative importance the decision-maker implicitly assigns to their ith experience when forming beliefs about the transition dynamics.

estimating the transition function $\hat{p}_{\mathbf{w}} = p(\cdot \mid \hat{\boldsymbol{\theta}}_{\mathbf{w}})$, the forward problem reduces to solving the linear program in (3), substituting $\hat{p}_{\mathbf{w}}$ for p.

In principle, $p(s' \mid s, a; \boldsymbol{\theta})$ may be fully unstructured, with a distinct parameter $\theta_{s',s,a}$ governing each (s', s, a) triple. This specification includes the empirical transition probabilities as a special case. However, such a model is often severely underdetermined when data are sparse relative to the size of the state–action space. In practice, lower-dimensional parameterizations (e.g., logistic or multinomial regression) can be used to impose structure and reduce variance. We illustrate both the fully parameterized and structured settings in Sections 5 and 6.

3 INVERSE PROBLEM

We now define the model-based and data-driven versions of the inverse problem. We restrict our focus to the setting of interest, that is when the reward function is known and the goal is to learn a transition function that makes an observed policy optimal.

3.1 Model-Based Inverse Problem

We define the model-based inverse problem as determining a transition function $\tilde{p}$ under which an observed policy $\tilde{\pi}$ is optimal, given a known reward function r. This is the formulation studied in [8]. Conceptually, this amounts to identifying $\tilde{p}$ such that

$$\tilde{\pi} \in \underset{\pi \in \Pi}{\operatorname{argmax}} \{v_\pi(s; \tilde{p}) : s \in \mathcal{S}\}. \tag{5}$$

In their implementation, [8] treat the transition model as fully unstructured, specifying a distinct parameter $\tilde{p}(s'|s, a)$ for each (s, s', a) triple. They enforce the optimality condition above using the primal and dual linear programming formulations of the forward MDP, leading to a nonconvex bilinear program (their equation (12)). Feasible solutions that attain an objective value of zero (i.e., zero duality gap) yield transition probabilities for which the optimality condition in (5) holds.

3.2 Data Driven Inverse Problem

In the data-driven inverse problem, we assume the same inputs as in the data-driven forward problem defined in Section 2.2: a dataset $\mathcal{D} = \{(s_i, a_i, s'_i)\}_{i=1}^n$ of observed transitions and a known reward function r. We treat the observed actions in $\mathcal{D}$ as generated by the forward problem described in Section 2.2: the decision-maker first estimated a transition function by solving (4) and then chose actions according to the optimal policy obtained by solving (3) under this estimate.

We assume that each observation (s_i, a_i, s_i') in $\mathcal{D}$ has an associated (unobserved) weight w_i reflecting the relative importance the decision-maker assigned to that experience when forming beliefs about the transition dynamics. Our goal in the data-driven inverse problem is to estimate a weight vector $\mathbf{w} = (w_1, \ldots, w_n)$, such that when the data-driven forward problem is solved with this weight vector, the resulting prescribed actions best match the observed actions.

Formally, for a given weight vector $\mathbf{w}$, let $p_{\mathbf{w}}(s'|s,a)$ denote the transition model obtained by solving (4) under weights $\mathbf{w}$, and let $v^*(\cdot; p_{\mathbf{w}})$ denote the corresponding optimal value function obtained by solving (3). Let the optimal action in arbitrary state s under transition model $p_{\mathbf{w}}$ be defined as

$$a^*(s; p_{\mathbf{w}}) = \arg\max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} p_{\mathbf{w}}(s'|s,a) \big[r(s,a,s') + \gamma\, v^*(s'; p_{\mathbf{w}}) \big]. \tag{6}$$

The data-driven inverse problem is then

$$\min_{\mathbf{w} \geq 0} \; \ell_{\mathbf{a}}(\mathbf{a},\, \mathbf{a}^*(\mathbf{s}; p_{\mathbf{w}}))\,, \tag{7}$$

where $\mathbf{s} = (s_1, \ldots, s_n)$ denotes the sequence of observed states in $\mathcal{D}$, $\mathbf{a} = (a_1, \ldots, a_n)$ denotes the corresponding observed actions, $\mathbf{a}^*(\mathbf{s}; p_{\mathbf{w}}) = (a_1^*(s_1; p_{\mathbf{w}}), \ldots, a_n^*(s_n; p_{\mathbf{w}}))$ denotes the vector of optimal actions at the observed states under $p_{\mathbf{w}}$, and $\ell_{\mathbf{a}}$ measures the discrepancy between the observed and induced actions.

We also consider a more general formulation in which the goal is to estimate weights $\mathbf{w}$ such that a given policy $\tilde{\pi}$ is minimally suboptimal under transition model $p_{\mathbf{w}}$. Specifically, we solve

$$\min_{\mathbf{w} \geq 0} \ell_\pi(\tilde{\pi}, \pi^*_{\mathbf{w}}) \tag{8}$$

where $\tilde{\pi}$ is the target policy (supplied as an additional input in the problem), $\pi^*_{\mathbf{w}}$ denotes an optimal policy induced by $p_{\mathbf{w}}$ via (3), and ℓ_π is a loss function measuring the discrepancy between the target policy and the induced policy.

We note that introducing weight vector $\mathbf{w}$ is not mathematically required to pose the data-driven inverse problem. One could instead optimize directly over the parameters of p to minimize the discrepancy between the observed and optimal actions. Our choice to include $\mathbf{w}$ is motivated by interpretability rather than necessity. Reweighting the data shifts the object of inference from the decision-maker's believed transition function to the influence of specific experiences on those beliefs. The implied transition model $p_{\mathbf{w}}$ still follows from (4), but the weights reveal *how* different transitions must be emphasized or discounted for the observed actions to be rationalized. This shift makes the resulting inferences easier to interpret in empirical settings, where understanding how past experiences shaped a decision-maker's beliefs may be more informative than the transition probabilities in isolation.

3.3 Non-Identifiability of the Inverse Problem

A basic property of the inverse problems defined above is that the transition function in an MDP is not always identifiable from a reward function and policy alone. In general, many distinct transition models can make the same policy optimal.

Proposition 1. *Fix a reward function r and a policy π, and define*

$$\mathcal{P}_\pi = \{\, p : v_\pi(s;p) \geq v_{\pi'}(s;p)\ \forall s \in \mathcal{S},\ \forall \pi' \in \mathit{\Pi}\,\}.$$

In general, $\mathcal{P}_\pi$ is not a singleton.

Proof. See Appendix A.

As a counter-example, consider a two-armed bandit with actions $\{A, B\}$ and a deterministic policy with $\pi(A) = 1$. Any transition structure that produces values (v_A, v_B) satisfying $v_A > v_B$ renders π optimal. Because there are infinitely many such value pairs, and because each value pair can arise from infinitely many transition models, the inverse mapping cannot be unique.

Because the inverse problem generally admits many distinct solutions, our goal is not to identify a single transition model but to explore plausible regions of the solution space. In the next section, we introduce a simple heuristic method that iteratively searches over weight vectors and provides a practical means of approximating such solutions.

4 A HEURISTIC SOLUTION METHOD

Although the data-driven inverse problems in (7) and (8) can be written as well-defined optimization problems, solving them directly is difficult. Doing so would require embedding (4) *within* (3). This is challenging for two reasons. First, for most machine-learning models and training losses, we do not have a compact representation of the optimality conditions associated with (4). Second, even if such a representation were available, the resulting formulation would contain bilinear terms between the choice probabilities p and the value function v, leading to the same challenges observed in [8].

To avoid these difficulties, we adopt a simple heuristic that separates (4) and (3). The heuristic iteratively searches over weight vectors $\mathbf{w}$ using a user-specified proposal function to generate candidate updates, accepting those that improve the fit between the induced optimal actions and the observed actions. As summarized in Algorithm 1, each iteration consists of two steps: (i) re-estimating the transition model through (4) under the proposed weights, and (ii) solving the forward MDP via (3) to determine whether the resulting optimal policy reduces the loss. We write the algorithm using the action-based loss (i.e., assuming the objective in (7)), but the policy-based version can be obtained by simply replacing $\ell_{\mathbf{a}}(\mathbf{a}, \mathbf{a}^*)$ with $\ell_\pi(\tilde{\pi}, \pi^*_{\mathbf{w}})$ in Lines 4 and 9.

Algorithm 1 Iterative Reweighting

Require: Dataset $\mathcal{D} = \{(s_i, a_i, s_i')\}_{i=1}^n$, number of iterations T
Ensure: Final weight vector $\mathbf{w}^{(T)}$

$\mathbf{w}^{(0)} \leftarrow \mathbf{1}_n$
$\boldsymbol{\theta}^{(0)} \leftarrow \arg\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \frac{1}{n}\sum_{i=1}^n w_i^{(0)} \ell_{\boldsymbol{\theta}}[s_i', p(s_i'|s_i, a_i; \boldsymbol{\theta})]$
$p^{(0)} \leftarrow p(\cdot \mid \boldsymbol{\theta}^{(0)})$
$l^{(0)} \leftarrow \ell_{\mathbf{a}}\Big(\mathbf{a},\, \mathbf{a}^*(\mathbf{s}; p^{(0)})\Big)$
for $j = 1$ to T **do**
  $\mathbf{w}^{\text{prop}} \leftarrow u(\mathbf{w}^{(j-1)})$
  $\boldsymbol{\theta}^{\text{prop}} \leftarrow \arg\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \frac{1}{n}\sum_{i=1}^n w_i^{\text{prop}}\, \ell_{\boldsymbol{\theta}}[s_i',\, p(s_i' \mid s_i, a_i; \boldsymbol{\theta})]$
  $p^{\text{prop}} \leftarrow p(\cdot \mid \boldsymbol{\theta}^{\text{prop}})$
  $l^{\text{prop}} \leftarrow \ell_{\mathbf{a}}(\mathbf{a},\, \mathbf{a}^*(\mathbf{s}; p^{\text{prop}}))$
  if $l^{\text{prop}} < l^{(j-1)}$ **then**
    $\mathbf{w}^{(j)} \leftarrow \mathbf{w}^{\text{prop}}$
    $l^{(j)} \leftarrow l^{\text{prop}}$
  else
    $\mathbf{w}^{(j)} \leftarrow \mathbf{w}^{(j-1)}$
    $l^{(j)} \leftarrow l^{(j-1)}$
  end if
end for

The proposal function u in Algorithm 1 is chosen by the user and determines how candidate weight vectors are generated. In the examples in Sections 5 and 6, we use a bootstrap resampling proposal, which draws $\mathbf{w}^{\text{prop}}$ from a multinomial distribution with probabilities proportional to the current weights $\mathbf{w}^{(j-1)}$. This choice restricts the weights to nonnegative integers and favors proposals that emphasize transitions already influential under the current weighting.

Using a stochastic update proposal is important as this is what helps the algorithm explore the space of plausible solutions. Since the inverse problem is non-identifiable (Section 3.3), many different weight vectors (corresponding to many different implied transition models) can rationalize the observed behavior. A deterministic update rule would tend to return a single solution or become trapped in a narrow region of the weight space. In contrast, a stochastic proposal function allows the algorithm to produce alternative weight vectors across different runs. This variability is a feature rather than a bug: it provides a practical mechanism for sampling weight vectors from the solution space and assessing the stability of the behavioral patterns inferred from them.

5 CRUMBLY CLIFF WALK

We illustrate our method on a modified Cliff Walk environment. The state space consists of a 5×4 grid, illustrated in Figure 2a. At each epoch, the agent can move to any adjacent cell, as shown by Figure 2b, incurring a reward of -1. The goal is to travel from $(1, 1)$ to $(5, 1)$ in as few steps as possible while moving along a "cliff" in rows 2–4. The closer the agent is to the cliff edge (row 2), the

higher the chance that the cell crumbles, causing the agent to fall and return to the start.

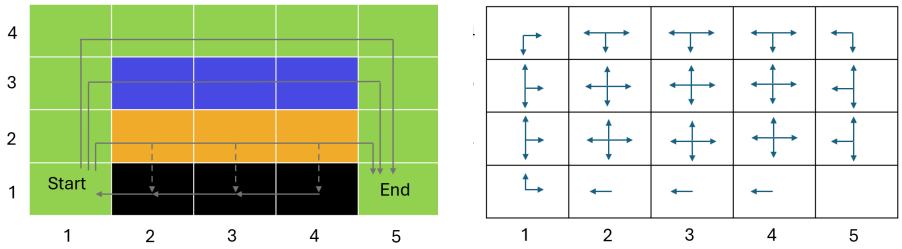


Fig. 2: Crumbly Cliff Walk environment **(a)** and corresponding action space **(b)**. Green cells never crumble, blue cells crumble with probability p^L, and yellow cells crumble with probability p^H (with $p^L < p^H$). Black cells represent the bottom of the cliff. The agent's objective is to travel from $(1, 1)$ to $(5, 1)$ as quickly as possible. Three illustrative routes are shown: a safe route through the green cells, a risky route through the yellow cells, and a middle route through the blue cells.

Yellow cells crumble with probability p^H and blue cells crumble with probability p^L, where $p^L < p^H$. When a cell crumbles, the agent transitions to the black cell directly below, regardless of the action taken. The black cells represent the bottom of the cliff; if the agent lands in one of these, it must return to the start and try again. This creates a natural risk–reward tradeoff: the fastest path lies along row 2, but the safer and more reliable route is along row 4.

5.1 Crumbly Cliff Walk Forward Problem

In the forward problem for the Crumbly Cliff Walk, we assume the agent knows the reward function $r(s, a, s') = -1$ and has completed an exploration phase yielding a dataset $\mathcal{D} = \{(s_i, a_i, s'_i)\}_{i=1}^n$ of observed transitions. The agent's goal is to estimate a policy that minimizes the expected number of steps required to reach the end state $(5, 1)$. Throughout this subsection we assume uniform weights $w_i = 1$, so the transition model is estimated using the unweighted version of (4). This reflects the standard forward-learning setting in which all observed transitions are treated as equally informative.

In this environment, the transition model reduces to estimating the fall probabilities for the yellow (high-risk) and blue (low-risk) cells. Let $\mathcal{S}_{\text{YEL}}$ and $\mathcal{S}_{\text{BLU}}$ denote the sets of yellow and blue cells, respectively, and let $\mathcal{S}_{\text{BLK}}$ denote the black cliff-bottom cells to which the agent transitions when a cell crumbles. The

fall probabilities are estimated as empirical proportions:

$$\hat{p}^H = \frac{\sum_{i:s_i \in \mathcal{S}_{\text{YEL}}} \mathbb{I}\{s_i' \in \mathcal{S}_{\text{BLK}}\}}{\sum_{i:s_i \in \mathcal{S}_{\text{YEL}}} 1}, \qquad \hat{p}^L = \frac{\sum_{i:s_i \in \mathcal{S}_{\text{BLU}}} \mathbb{I}\{s_i' \in \mathcal{S}_{\text{BLK}}\}}{\sum_{i:s_i \in \mathcal{S}_{\text{BLU}}} 1}.$$

These two probabilities determine the estimated transition model $\hat{p}$, from which the estimated optimal policy $\hat{\pi}$ is obtained by solving the forward MDP in (3).

In this MDP, three natural *policy classes* arise, distinguished by the path they take from start to end. We refer to these as the risky, middle, and safe classes. A policy belongs to the risky class if it moves the agent along row 2 (the yellow cells) toward the goal; to the middle class if it moves along row 3 (the blue cells); and to the safe class if it moves along row 4 (the green cells). Each class contains many distinct policies—for example, policies may differ in how they resolve ties or behave off the main path—but all such policies share the same qualitative route. For every fall-probability pair (p^H, p^L) considered, at least one of the three policy classes is optimal. In this sense, the three classes partition the (p^H, p^L) parameter space, as illustrated in Figure 3a.

5.2 Action-Matching Inverse Problem

Suppose we observe an agent's experience dataset $\mathcal{D} = \{(s_i, a_i, s_i')\}_{i=1}^n$ generated from the Crumbly Cliff Walk representing the exploration phase, and a set of actions $\mathbf{a} = (a_1, \ldots, a_n)$ that can come from a dataset or be prespecified. In the action-matching inverse problem, our aim is to infer a weight vector $\mathbf{w}$ such that the optimal actions moving from the start to the end induced by the transition model $p_{\mathbf{w}}$ align as closely as possible with the observed actions. These actions describe the observed route the agent takes.

From $\mathcal{D}$, we can estimate $\hat{p}^H_{\mathbf{w}}$ and $\hat{p}^L_{\mathbf{w}}$ with weights of 1, then compare the given $\mathbf{a}$ to the estimated optimal actions, $\mathbf{a}^*(s, p_{\mathbf{w}})$. The problem then is to estimate $\mathbf{w}$ such that $\ell(\mathbf{a}, \mathbf{a}^*(\mathbf{s}, p_{\mathbf{w}}))$ is minimized. We define $\ell(\cdot, \cdot)$ to be

$$\ell_{\mathbf{a}}(\mathbf{a}, \mathbf{a}^*(\mathbf{s}; p_{\mathbf{w}})) = \sum_{i=1}^{n} q(s_i, a_i^*; p_{\mathbf{w}}) - q(s_i, a_i; p_{\mathbf{w}}) \tag{9}$$

where $q(s, a; p_{\mathbf{w}}) = \sum_{s' \in \mathcal{S}} \hat{p}_{\mathbf{w}}(s'|s, a) v^*(s')$. This loss function quantifies the sub-optimality of the desired actions given $\hat{p}_{\mathbf{w}}$.

We apply Algorithm 1 a bootstrap proposal for $\mathbf{w}$. For each simulation we generate exploration-phase data under one of three sets of transition probabilities, shown as the black stars in Figure 3a.[7] We then select one of the two other action sets as the target. Resulting samples of (p^H, p^L) are shown as points in Figure 3a.

[7] The probabilities are (.21,.15) for the safe action set, (.1,.05) for the risky set, and (.15,.05) for the middle set.

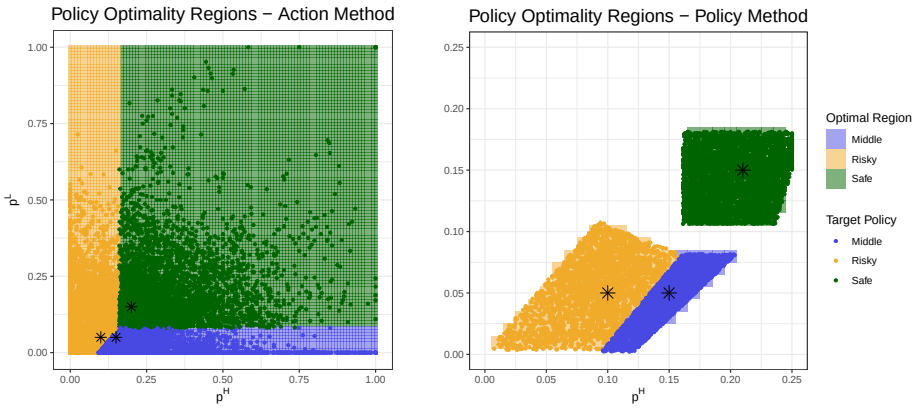


Fig. 3: Both figures show results from a simulation study solving the inverse problems described in Sections 5.2 and 5.3. The filled in regions show the probability space for which the given action set/policy is optimal, estimated with a grid search. The points show the solutions from Algorithm 1, colored by the target action set/policy. Note that apparent mismatches between the dots and the filled regions are mostly due to the fineness of the grid search. The black stars show fixed probabilities for which the given action set/policy is optimal, and the probabilities used to generate data for the algorithm. **(a)** Sampled points from the probability space using the method described in Section 5.2. **(b)** Sampled points from the probability space using the method described in Section 5.3.

5.3 Policy Matching Inverse Problem

Instead of matching specific actions, we may target a full policy $\tilde{\pi}$ instead. We define three policies, with the defining (p^H, p^L) sets and optimal regions shown in Figure 3b. Note that these regions are subsets of those in Figure 3a because there are many different policies within a policy class.

We define the loss between the target policy $\tilde{\pi}$ and the optimal policy based on the weighted transition probabilities $\pi_{\mathbf{w}}$ as

$$\ell_\pi\left(\tilde{\pi}, \pi_{\mathbf{w}}\right) = \sum_{s \in S} \left|\hat{v}_{\pi_{\mathbf{w}}}(s; \hat{p}_{\mathbf{w}}) - \hat{v}_{\tilde{\pi}}(s; \hat{p}_{\mathbf{w}})\right|, \tag{10}$$

where $\hat{v}_\pi(s; \hat{p}_{\mathbf{w}})$ is the estimated value of the state s with respect to a given policy π and the transition dynamics $\hat{p}_{\mathbf{w}}$.

We again apply Algorithm 1 with bootstrap proposals, with the results visualized in Figure 3b.

6 FOURTH DOWN PROBLEM

We now apply our method to NFL fourth-down decision-making, where coaches' actions frequently diverge from statistically optimal recommendations [14, 19,

13], as illustrated in Figure 1. The data for this section comes from the *nflreadr* package [9], which is available for use under the MIT license.

[16] examined a related version of the fourth-down problem using inverse optimization to recover the degree of risk aversion consistent with coaches' decisions. Their approach estimates the quantiles of the empirical transition distribution that best rationalize observed actions, thereby characterizing how conservative a coach is relative to the empirical transition probabilities. In contrast, our method directly estimates transition probabilities that coaches may *believe*, with corresponding uncertainty, providing a more granular and structurally interpretable view of how their perceived environment differs from the empirical one.

6.1 Problem Setup

We model fourth-down plays as an MDP. A fourth down state $s \in S_4$ encodes: possession team, yardline, yards to go, time remaining (seconds), score differential, and fourth quarter indicator. Actions are $A = \text{GO}, \text{FGA}, \text{PUNT}$. Rewards correspond to next scoring outcomes from team A's perspective:

- $+7$ for touchdown,
- $+3$ for made field goal,
- -2 for conceded safety,
- 0 for no score before end of half.

We estimate transition probabilities using weighted multinomial logistic regression following [20]. For state s and action a, let $\hat{p}_{a,s,\mathbf{w}}$ be the predicted distribution over scoring outcomes under weights $\mathbf{w}$. Complete details of our fourth down environment specification can be found in Appendix B.

Rather than estimating $\tilde{\pi}$ and comparing it to an estimated optimal policy, we compare (via (7)) the history of observed actions $\mathbf{a} = (a_1, ..., a_N)$ and the corresponding states $\boldsymbol{s} = (s_1, ..., s_N)$ to estimated optimal actions $a^*(s; p_{\mathbf{w}})$ as defined in (6). We use misclassification rate as the loss function (see (15) in Appendix B), and we use Algorithm 1 with the bootstrap update to obtain solution weights.

6.2 Results

Expected Points Shift The simplest way to interpret the algorithm's results is to focus on a specific field position. For example, Figure 4 shows the empirical and reweighted distributions from a single run of the algorithm for each of the actions when between the 31 and 40 yardlines with 3 or 4 yards to go. The empirical average of Going for it is the highest, but the field goal is better after reweighting. Weights concentrate only on successful field goal outcomes, suggesting coaches overestimate field goal success rates while underestimating success rates when going for it.

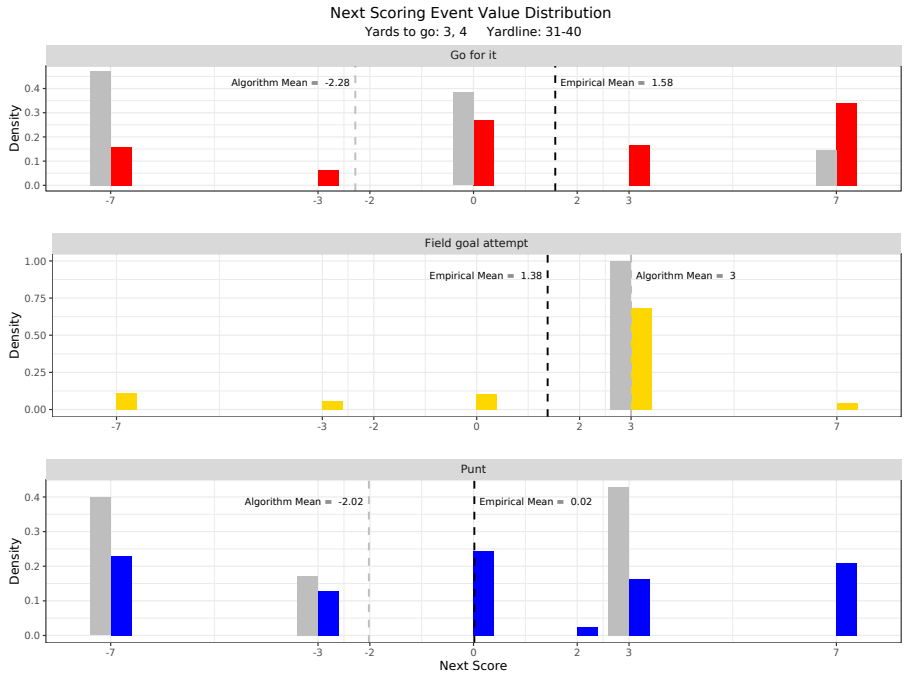


Fig. 4: Empirical and weighted density plots of the three actions when between the 31 and 40 yardlines with 3 or 4 yards to go. Colored densities show the empirical distribution, while the grey shows the reweighted distribution.

Quantifying Bias One of the main strengths of our algorithm is the ability to characterize variability across plausible solutions.. We run Algorithm 1 for 100 iterations to sample from the solution space. For each run, we compute the difference between believed and empirical value functions $\Delta(s,a) = q(s,a;\hat{p}_{\mathbf{w}}) - q(s,a;\hat{p})$ where $\hat{p}_{\mathbf{w}}$ are the estimated weighted transition probabilities, and $\hat{p}$ are the empirical transition probabilities. Figure 5 summarizes the range and mean of Δ. It is clear that there is a large region in the Go for which every sampled solution was negative, indicating that there is strong evidence coaches systematically undervalue the expected value of going for it across broad regions. There is also a region where field goals are similarly undervalued, though overall the beliefs about field goals and punts are relatively consistent with empirical values. These patterns are consistent with established behavioral explanations and risk attitudes predicted by prospect theory, as discussed in Appendix B.

7 DISCUSSION

We introduce a data-driven framework for inferring a decision-maker's beliefs about the transition probabilities of an MDP. We infer these implied beliefs by

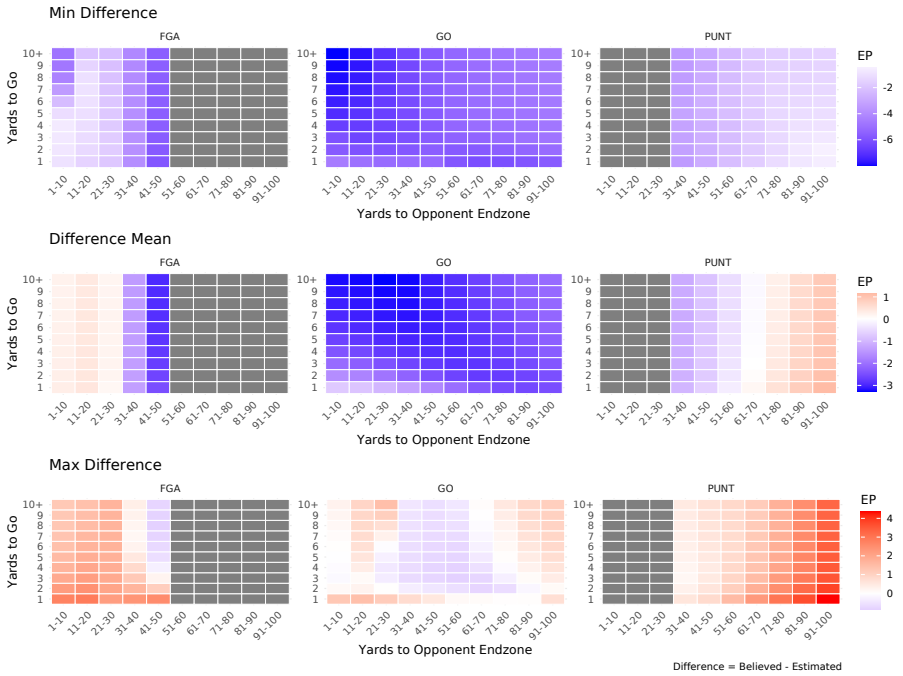


Fig. 5: Range of the estimated value functions obtained from 100 runs of Algorithm 1. States are visualized by how many yards to go for a first down and binned by position on the field. Positive values in a cell indicate that coaches overvalue that action in that state, and negative values are the opposite.

assuming that agents systematically reweight past experiences (operationally, the rows of a data set) when estimating transition probabilities. The objective is to learn a set of weights on past experiences such that, when the MDP is solved using the reweighted data, the resulting policy aligns with the observed behavior. This provides an interpretable way to quantify how decision-makers implicitly over- or under-emphasize past experiences when forming beliefs about their environment.

The application to the fourth down decision problem shows that this approach can reveal meaningful behaviorial patterns consistent with human biases. The learned weights offer a measure of each observation's influence on perceived dynamics, yielding a novel quantitative window into coaches' biases.

The primary limitation of our method is that it relies on a heuristic approach. While the method performs well on the examples we have explored, we provide no formal guarantees of convergence, identifiability, or uniqueness. Formal proofs of theoretical properties, such as existence, consistency, or asymptotic behavior of the learned weights, are important directions for future work.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Bibliography

[1] Abbeel, P., Ng, A.Y.: Apprenticeship learning via inverse reinforcement learning. In: Proceedings of the twenty-first international conference on Machine learning. p. 1 (2004)

[2] Ahuja, R.K., Orlin, J.B.: Inverse optimization. Operations research **49**(5), 771–783 (2001)

[3] Baldwin, B.: nfl4th. https://www.nfl4th.com/articles/articles/4th-down-research.html (2021), accessed: 2021-03-31

[4] Burke, B., Carter, S., Giratikanon, T., Quealy, K., Daniel, J.: 4th down: When to go for it and why. https://www.nytimes.com/2014/09/05/upshot/4th-down-when-to-go-for-it-and-why.html (2014), accessed: 2021-03-23

[5] Chan, T.C., Fernandes, C., Puterman, M.L.: Points gained in football: Using markov process-based value functions to assess team performance. Operations Research (2021)

[6] Chick, C., Pardo, S., Reyna, V., Goldman, D.: Decision making (individuals). In: Ramachandran, V. (ed.) Encyclopedia of Human Behavior (Second Edition), pp. 651–658. Academic Press, San Diego, second edition edn. (2012). https://doi.org/https://doi.org/10.1016/B978-0-12-375000-6.00122-1, https://www.sciencedirect.com/science/article/pii/B9780123750006001221

[7] Erkin, Z., Bailey, M.D., Maillart, L.M., Schaefer, A.J., Roberts, M.S.: Eliciting patients' revealed preferences: An inverse markov decision process approach. Decision Analysis **7**(4), 358–365 (2010)

[8] Ghatrani, Z., Ghate, A.: Inverse markov decision processes with unknown transition probabilities. IISE Transactions **55**(6), 588–601 (2023)

[9] Ho, T., Carl, S.: nflreadr: Download 'nflverse' Data (2025), https://nflreadr.nflverse.com, r package version 1.5.0

[10] Kahneman, D., Tversky, A.: Prospect theory: An analysis of decision under risk. In: Handbook of the fundamentals of financial decision making: Part I, pp. 99–127. World Scientific (2013)

[11] Ng, A.Y., Russell, S., et al.: Algorithms for inverse reinforcement learning. In: Icml. vol. 1, p. 2 (2000)

[12] Puterman, M.L.: Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons (2014)

[13] Roach, M.A., Owens, M.F.: Updating beliefs based on observed performance: Evidence from nfl head coaches. Journal of Sports Economics p. 15270025231222633 (2024)

[14] Romer, D.: Do firms maximize? evidence from professional football. Journal of Political Economy **114**(2), 340–365 (2006)

[15] Russell, S.: Learning agents for uncertain environments. In: Proceedings of the eleventh annual conference on Computational learning theory. pp. 101–103 (1998)

[16] Sandholtz, N., Wu, L., Puterman, M., Chan, T.C.: Learning risk preferences in markov decision processes: An application to the fourth down decision in the national football league. The Annals of Applied Statistics **18**(4), 3205–3228 (2024)
[17] Spranca, M., Minsk, E., Baron, J.: Omission and commission in judgment and choice. Journal of experimental social psychology **27**(1), 76–105 (1991)
[18] Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
[19] Yam, D.R., Lopez, M.J.: What was lost? a causal estimate of fourth down behavior in the national football league. Journal of Sports Analytics **5**(3), 153–167 (2019)
[20] Yurko, R., Ventura, S., Horowitz, M.: nflwar: a reproducible method for offensive player evaluation in football. Journal of Quantitative Analysis in Sports **15**(3), 163–183 (2019)
[21] Ziebart, B.D., Maas, A.L., Bagnell, J.A., Dey, A.K., et al.: Maximum entropy inverse reinforcement learning. In: Aaai. vol. 8, pp. 1433–1438. Chicago, IL, USA (2008)

A PROOF OF PROPOSITION 1

Proof. For a fixed policy π, transition model p, and reward function r, the value function v_π is uniquely determined as the solution to the Bellman equation:

$$v_\pi(s;p) = \sum_a \pi(a|s) \sum_{s'} p(s'|s,a)\big[r(s,a,s') + \gamma v_\pi(s';p)\big].$$

Define the *effective transition kernel* induced by π:

$$p_\pi(s'|s) := \sum_a \pi(a|s)p(s'|s,a),$$

and the *policy-conditioned reward*:

$$r_\pi(s,s') := \frac{\sum_a \pi(a|s)p(s'|s,a)r(s,a,s')}{p_\pi(s'|s)}.$$

Then the Bellman equation can be rewritten as

$$v_\pi(s;p) = \sum_{s'} p_\pi(s'|s)\big[r_\pi(s,s') + \gamma v_\pi(s';p)\big].$$

Hence, v_π depends only on p_π, the *effective transition dynamics* under π, and not on the individual components $p(s'|s,a)$.

Consequently, any two transition models p_1 and p_2 satisfying

$$\sum_a \pi(a|s)p_1(s'|s,a) = \sum_a \pi(a|s)p_2(s'|s,a), \quad \forall s,s',$$

induce the same effective transition kernel p_π, and therefore the same value function v_π. Since the optimality of π depends only on the relative values $v_\pi(s;p)$ and $v_{\pi'}(s;p)$, all such transition models p will make π optimal.

Define the inverse set of transition models consistent with π as

$$\mathcal{I}_R(\pi) := \{\, p \mid \pi \in \arg\max_{\pi'} v_{\pi'}(s|p) \,\}.$$

Then $\mathcal{I}_R(\pi)$ may contain infinitely many elements whenever distinct p's induce the same p_π.

Therefore, the transition model p is *not identifiable* from (f, π): the dependence of v_π on p occurs only through the product $\pi(a|s)p(s'|s,a)$, so π and p cannot be separated.

B APPLICATION TO THE NFL FOURTH DOWN DECISION

B.1 The Fourth Down Decision as an MDP

The fourth down MDP was briefly described in Section 6, but here we will be more specific about its components.

We define the state space, $\mathcal{S}$, to be the union of two mutually exclusive sets: $\mathcal{S}^{\text{score}}$, the set of scoring states, and $\mathcal{S}^{\text{play}}$, the set of game states at the start of a play. A scoring state is a tuple consisting of the team that scored (denoted generically as either team A or B) and the type of score (touchdown, field goal, safety, or No Score):

$$\mathcal{S}^{\text{score}} = \{\text{A}, \text{B}\} \times \{\text{TD}, \text{FG}, \text{SAF}, \text{NS}\}. \tag{11}$$

A play state is a tuple consisting of the team with possession, down, yardline, yards to go until first down, seconds left in the half, the score differential (possession team's score - opponent team's score), and whether or not it is the fourth quarter:

$$\mathcal{S}^{\text{play}} = \{\text{A}, \text{B}\} \times \{1, 2, 3, 4\} \times [1, 99] \times$$
$$\{1, \ldots, 9, 10+\} \times [1, 1800] \times \mathbb{Z} \times \{0, 1\}.$$

Let $\mathcal{S}^4 \subseteq \mathcal{S}^{\text{play}}$ denote the set of fourth down play states. Going forward, elements of $\mathcal{S}^4$ will be denoted s.

Actions Given that we treat future actions as deterministic according to fixed policy π, the only action set we are concerned with is the set of fourth down actions.[8] Hence $\mathcal{A} = \{\text{GO}, \text{FGA}, \text{PUNT}\}$, denoting go for it, field goal attempt, and punt, respectively. The set of realistic options for a given fourth down situation is typically either $\{\text{GO}, \text{FGA}\}$ or $\{\text{GO}, \text{PUNT}\}$, depending on the field position. We use $\mathcal{A}$ for simplicity.

[8] We make this assumption because we are interested in how changing fourth down decision making may affect results while keeping all other decision making the same.

Reward Function We define the reward function from the perspective of team A:

$$r(s) = \begin{cases} 7, & s = (\text{A, TD}) \in \mathcal{S}^{\text{score}}, \\ 3, & s = (\text{A, FG}) \in \mathcal{S}^{\text{score}}, \\ -2, & s = (\text{A, SAF}) \in \mathcal{S}^{\text{score}}, \\ 0, & s = (\text{A, NS}) \in \mathcal{S}^{\text{score}}. \end{cases} \tag{12}$$

If team B scores, the reward values are the negative of the values in (12). The reward state of NS only occurs if no other scoring state is observed before the end of the half. For simplicity, we model the reward of a touchdown as being worth 7 points.[9]

Objective Function In a fourth down situation, the coach wants to choose the action that will maximize the long term reward, assuming they will follow a given policy π thereafter. Thus, the optimization problem will be of the form

$$\max_{a \in \mathcal{A}} q_\pi(s, a; p) \tag{13}$$

where $q_\pi(s, a; p)$ represents the value associated with taking action a in fourth down state s, under a given set of transition probabilities p. In our analysis, we will fix π as the *league-average policy* $\bar{\pi}$, which we assume to be stationary. We will use $q_{\bar{\pi}}(s_t, a; p)$ as defined by [18], with the rewards $r(s)$ as defined in (12).

B.2 Estimating the Objective Function

In order to estimate the objective function based on our dataset, we will use multinomial logistic regression. Specifically, for each fourth down play s in our dataset, we follow a similar procedure described by [20] to calculate the expected points in a given state and action pair. To do this, we fit a separate multinomial logistic regression model for each action (GO, PUNT, FGA), predicting the distribution over subsequent scoring outcomes. Then these three models can be evaluated on a new state to determine whether going for it, punting, or kicking a field goal yields the highest expected points. This approach implicitly marginalizes over intermediate transitions and the league-average continuation policy, allowing us to estimate expected points directly from observed data. It is also simple to fit and evaluate on a new dataset, which is ideal for solving the inverse problem.

We define

$$\hat{\mathbf{p}}_{a,s,\mathbf{w}} = \begin{bmatrix} \hat{p}(r = 7|s, a, \mathbf{w}) \\ \hat{p}(r = 3|s, a, \mathbf{w}) \\ \vdots \\ \hat{p}(r = -7|s, a, \mathbf{w}) \end{bmatrix},$$

[9] Previous studies ([5], [14], and [20]) have found that the league-average value is close to this, and we don't wish to include the complexity of one- and two-point conversion attempts in this work.

with $r = \sum_{k=t+1}^{K} r(s_k)$ the reward of the next scoring event, and the probabilities estimated with weighted logistic regression using weights $\mathbf{w}$. Note negative rewards arise when the next scoring event is by the opposing team. We then let $v = \begin{bmatrix} 7 & 3 & \dots & -7 \end{bmatrix}^T$, and can estimate

$$\hat{q}_{\pi_{\mathbf{w}}}(s, a; \hat{\mathbf{p}}_{a,s,\mathbf{w}}) = \hat{\mathbb{E}}_{\pi} \left[\sum_{k=t+1}^{K} r(s_k) | s, a, \mathbf{w} \right] = v^T \hat{\mathbf{p}}_{a,s,\mathbf{w}} \tag{14}$$

for a given (s, a) combination, and then identify the action a which maximizes $\hat{q}_{\pi_{\mathbf{w}}}$.

B.3 The Inverse Problem

For the fourth down problem, Rather than estimating $\tilde{\pi}$ and comparing it to an estimated optimal policy, we compare the history of observed actions $\mathbf{a} = (a_1, ..., a_N)$ and the corresponding states $\boldsymbol{s} = (s_1, ..., s_N)$ to estimated optimal actions $\mathbf{a}^*(s; \hat{\mathbf{p}}_{a,s,\mathbf{w}})$ as defined in (6).

The Loss Function Our goal will be to make $\mathbf{a}^*(s; \hat{\mathbf{p}}_{a,s,\mathbf{w}})$ as defined in (6) as similar as possible to $\mathbf{a}$, the vector describing the decisions which coaches made in a given state. As such we will use accuracy as our objective, or the misclassification rate as our loss function:

$$\ell_{\mathbf{a}}(\mathbf{a}, \mathbf{a}^*(\mathbf{s}; \hat{\mathbf{p}}_{a,s,\mathbf{w}})) = \frac{1}{N} \sum_{t=1}^{N} \mathbb{I}\left[a^*(s_t; \hat{\mathbf{p}}_{a,s,\mathbf{w}}) \neq a_t \right]. \tag{15}$$

This loss function is not differentiable, but this is not a problem since Algorithm 1 does not require a differentiable loss.

B.4 Results

Unless otherwise specified, all the results in this section come from Algorithm 1 using the bootstrap update allowed to run for 1000 iterations, and with 100 different runs of the algorithm.

Expected Points Shift The simplest way to interpret the algorithm's results is to drill down on a specific field position. Figure 4 shows the empirical and reweighted distributions from a single run of the algorithm for each of the actions when between the 31 and 40 yardlines with 3 or 4 yards to go.

Policy Visualization We can visualize broadly how our algorithm changes the suggested policy. In Figure 6, the top left represents what coaches most commonly do in a given situation, and the top right represents what the expected points model recommends. Finally, the bottom left represents the most common optimal action after reweighing the dataset with Algorithm 1. This third plot is visually much more similar to that of the observed policy, showing that our algorithm was generally successful in matching the observed actions.

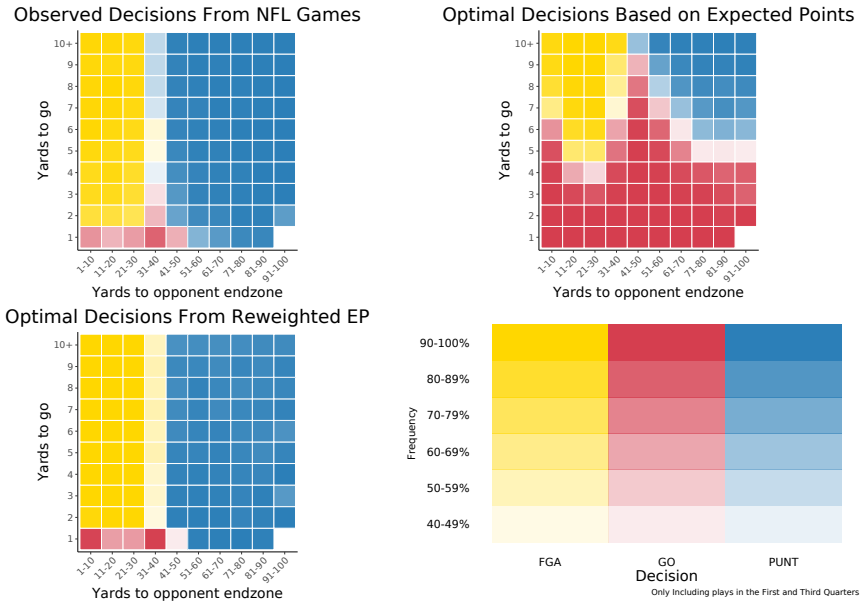


Fig. 6: Plot of three decisions maps representing average fourth down policies by location. Color indicates the most common action, with the alpha level being how frequent it is.

Bias over Time We can also explore coach bias over another aspect of the state space, time left in the game. Figure 7 shows curves representing the expected points (averaged over the rest of the state space) for both the "true", empirical model and the "believed", reweighted model. We use the multiple runs of Algorithm 1 on the football data to obtain the range of observed estimates fpr the believed value, and also estimate intervals on the empirical value estimate with bootstrap techniques. This allows us to compare the estimates while accounting for uncertainty.

We can see that the Punt and Field goal estimates are largely overlapping, while the belief estimate for Going for it is almost entirely separate from the true/empirical estimate. This allows us to confidently conclude that there is evidence that coaches have incorrect views about the value of going for it, at least over the Time dimension.

Bias Over the Field We can also visualize how biased coaches are in different locations on the field and for different actions. Figure 8 shows in the first row the average expected points in each location for each action based on the original data, and the second row shows the same for the reweighted data. We treat these reweighted expected points as what coaches believe. Finally, the third row is the difference between the two.

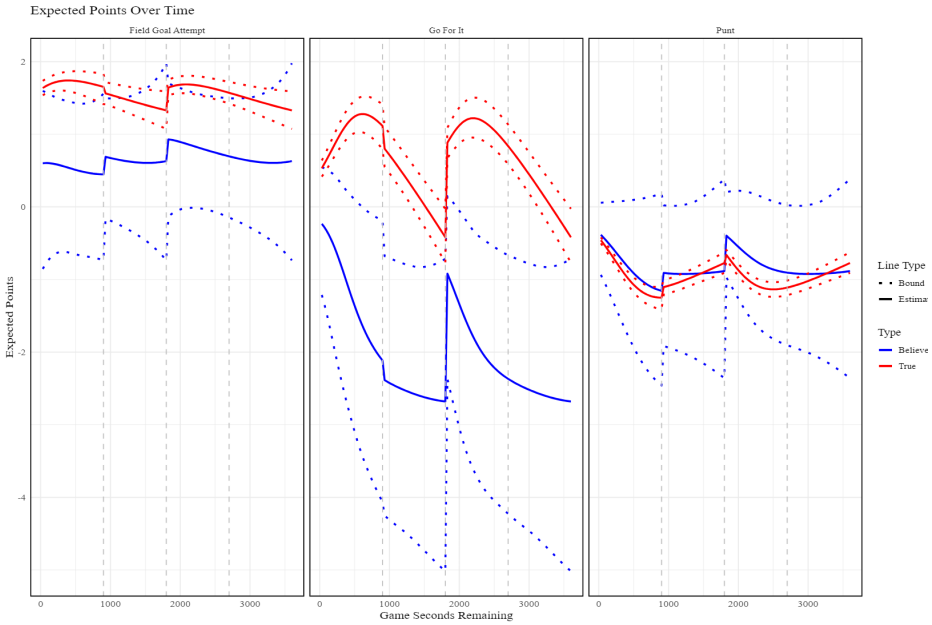


Fig. 7: Plots of the true (empirical) and believed (reweighted) average expected points estimates over the time dimension. Intervals show the range of observed values in the bootstrap process.

Additionally, taking advantage of the uncertainty quantification of our method, Figure 5 shows intervals for the difference at each location, as shown earlier.

From Figure 8, we can learn a couple of interesting things from the difference row. First, the go for it plot is mostly negative, indicating that the coaches undervalue going for it. Inversely, the punt plot is mostly positive, indicating coaches overvalue punting it. Interestingly, the field goal plot is generally near 0, suggesting that generally have correct beliefs about the outcomes of kicking the field goal.

From Figure 5, we can confirm that some of the intervals for the differences do not contain zero, suggesting that there are real differences. Unsurprisingly, go for it has the largest region of significant differences, confirming that the conclusions we drew from 8 are actually valid for those.

These results make sense from both an intuitive and psychological perspective. We can easily observe that coaches often punt when they should go for it, so the result of undervaluing going and overvaluing punting makes sense. These are also consistent with psychological theories such as omission bias [17] and prospect theory [10].

In omission bias, decision makers seem to judge harm as a result of comission (action) more negatively than harm from omission (inaction). Going for it is a much more active decision while punting it is inactive, and both have a high

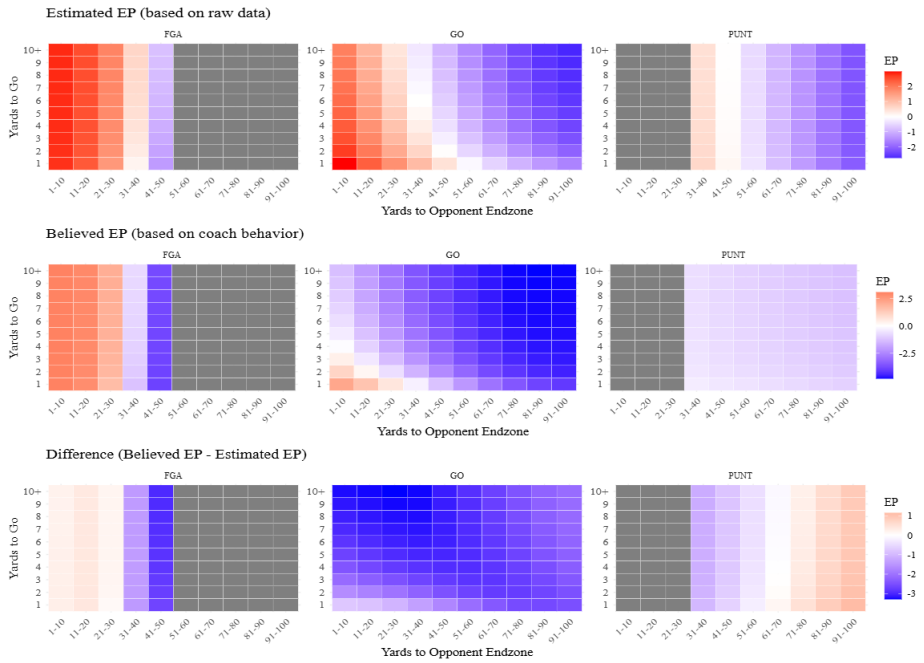


Fig. 8: Representations of Estimated Value Functions Based on 100 Runs of Algorithm 1. The Empirical Value Function (top), Mean Believed Value Function (middle), and Believed-Empirical (bottom).

potential of leading to negative outcomes. Thus, it follows that coaches view negative outcomes from going for it more harshly than punting it.

Prospect theory gives us another method to interpret coach behavior. In prospect theory, people tend to be risk averse for gains relative to the status quo, but risk seeking for losses relative to the status quo. For example, imagine that you were given $2000, but you had to then choose between two options: losing $1000 for sure or gambling on a 0.50 probability of losing $2000 and a 0.50 probability of losing nothing. In response to this problem most people prefer to gamble (they are risk seeking). However, whereas most people prefer the gamble when faced with loss, they tend to prefer the sure option when given the option between gaining $1000 for sure or $2000 with a 0.50 probability [6].

In the fourth down situation, coaches seem to be risk averse. This seems to indicate that coaches see their decision as choosing between a gain relative to the status quo. In other words, they are trying to make the decision that will give them the best outcome. However, it may be more logical for them to view their option as choosing between a loss. Most commonly on fourth down, coaches can either choose between punting it (losing $1000 for sure, since the other team will definitely gain possession) or going for it (losing $2000 with some probability, since they may succeed but if they fail the put the other team in a

better position). This insight may provide a way to communicate with coaches about the way they view decisions during the game and how they can change it to better act as analytics suggest.