Essays in Econometrics and Machine Learning

by

Colleen O'Briant

A dissertation accepted and approved in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in Economics

Dissertation Committee:
Keaton Miller, Chair
Mark Colas, Core Member
David Evans, Core Member
Brittany Erickson, Institutional Representative

University of Oregon
Spring 2024

© 2024 Colleen O'Briant All rights reserved.

This work is openly licensed via CC BY 4.0.



DISSERTATION ABSTRACT

Colleen O'Briant

Doctor of Philosophy in Economics

Title: Essays in Econometrics and Machine Learning

This dissertation aims to enhance transparency in AI systems by integrating methods from Machine Learning and Econometrics, specifically focusing on Dynamic Discrete Choice (DDC) models. In Chapter 2, I compare the Nested Fixed Point (NFXP) algorithm from Econometrics with Max-Margin Inverse Reinforcement Learning (IRL) methods from AI/ML, using Monte-Carlo experiments to demonstrate that preference shocks from Econometrics can resolve fundamental identification issues in IRL. The simulation results show that while Projection IRL is slightly less accurate than NFXP, IRL significantly reduces computational demands, requiring 20 times fewer dynamic programming problems to be solved. Chapter 3 investigates the practical applications of these methods by analyzing publicly available 2013 taxi data to compare IRL and NFXP in estimating payoffs for New York City taxi drivers during the morning commute. The analysis highlights that IRL's flatter objective function has the problem of allowing a broader range of acceptable payoff functions, however its feature expectation matching technique provides valuable feedback on the smoothing parameter for kernel density estimation of the transition probability function. This chapter offers recommendations and identifies potential drawbacks of using IRL, thereby deepening our understanding of the real-world performance of the algorithm.

In Chapter 4, the dissertation explores how small business owners may misattribute noise for profit signals using an instrumental variables approach and a rich dataset of product ordering decisions by Washington State marijuana dispensaries over the first three years of recreational marijuana legalization. The study examines whether entrepreneurs' predictions about product profitability are influenced by exogenous weather shocks, assessing if owners with previous retail experience make more informed decisions, if attentiveness improves over time, and if living further from the dispensary increases the likelihood of conflating weather shocks with profitability signals.

Contents

1 INTRODUCTION **8**

2 THE ECONOMETRICS OF INVERSE REINFORCEMENT LEARNING **10**

2.1 Introduction . . . 10
2.2 Background on Methods . . . 11
2.3 Model . . . 14
2.3.1 Nested Fixed Point Implementation Details . . . 14
2.3.2 Linear Programming IRL Implementation Details . . . 16
2.3.3 Projection IRL Implementation Details . . . 18
2.4 Methods . . . 19
2.5 Results . . . 20
2.6 Discussion . . . 21
2.6.1 Estimates Under Preference Shocks . . . 21
2.6.2 Identification Problem from Wind Shocks . . . 21
2.7 Conclusion . . . 23
2.8 Tables and Figures . . . 24

3 COMPARING INVERSE REINFORCEMENT LEARNING TO RUST'S ESTIMATOR **37**

3.1 Introduction . . . 37
3.1.1 Related Work . . . 38
3.2 Model . . . 39
3.3 Data . . . 40
3.4 Methods . . . 41
3.4.1 Estimating the Transition Probability Function . . . 41
3.4.2 Nested Fixed Point Algorithm . . . 43
3.4.3 Projection IRL . . . 43
3.5 Results . . . 45
3.6 Discussion . . . 46
3.7 Conclusions . . . 47
3.8 Tables and Figures . . . 48

4 LIMITED ATTENTION AND NEW PRODUCT ADOPTION FOR MARIJUANA DISPENSARIES **56**

4.1 Introduction . . . 56
4.2 Data . . . 58
4.2.1 Traceability Data . . . 58

4.2.2 Weather Data 58
4.2.3 Owner characteristics data 59
4.3 Model 60
4.4 Results 62
4.5 Conclusion 64
4.6 Tables and Figures 65

5 REFERENCES CITED **83**

List of Figures

1 Simulated Data on a 3-Parameter GridWorld 24
2 Simulated Data on a 4-Parameter GridWorld 26
3 Simulated Data on a 5-Parameter GridWorld 28
4 Cliff GridWorld with Preference Shocks 30
5 Cliff GridWorld with Wind Shocks (Coefficient = 0.7) 31
6 Cliff GridWorld with Wind Shocks (Coefficient = 0.5) 32
7 Preference Shocks 33
8 Wind Shocks 33
9 GridWorld Simulation Data and Results with 3 Parameters 34
10 GridWorld Simulation Data and Results with 4 Parameters 35
11 GridWorld Simulation Data and Results with 5 Parameters 36
12 Total Workers Per Census Tract 51
13 Feature Expectations under the Smoothing Parameter 0.01 54
14 Feature Expectations under the Smoothing Parameter 0.10 55
15 Median Wholesale Price for a Gram of the Twenty Most Popular Strains Each Month 65
16 Monthly Revenue for Forty Dispensaries 66
17 Number of Unique Products Ordered by Dispensaries Each Month 66
18 Seasonality of Daily High Temperatures 68
19 Residual Daily High Temperatures 68
20 Another View of Residual Daily High Temperatures 69
21 Residual Daily High Temperatures In Eastern versus Western Washington 69
22 Seasonality of Precipitation 70
23 OLS Precipitation Residuals 70
24 ZIP Precipitation Residuals 71
25 ZIP Precipitation in Western versus Eastern Washington 71
26 ZIP Precipitation in Western versus Eastern Washington 73
27 Effect of Residual Precipitation on Debut Sales by Season 74
28 Effect of Temperatures on Debut Popularity 75
29 Effect of Temperatures on Debut Popularity by Season 76

List of Tables

1 Results for the Estimation of a 3-Parameter GridWorld 25
2 Results for the Estimation of a 4-Parameter GridWorld 27
3 Results for the Estimation of a 5-Parameter GridWorld 29
4 Pickups and Dropoffs by Borough 48
5 Taxi Pickup and Dropoff Boroughs as Transition Probabilites 49
6 Morning Commute Payoff Function: Weekdays, 7:45am-9:45am 50
7 Average Means of Transportation to Work by Borough (2013) 52
8 Correlation Matrix: Means of Transportation to Work by Census Tract . 53
9 Correlation Matrix: Means of Transportation to Work by Census Tract (%) 53
10 Distribution of Dispensaries by Years of Previous Experience 66
11 Distance From Owner's Home Address to Their Dispensary 72
12 First Stage Results: Effect of Precipitation Shocks on a Product's Debut Sales 72
13 Effect of Residual Precipitation on Debut Sales by Season 75
14 Reduced Form Results: Effect of Temperature Shocks on a Product's Probability of Reorder 76
15 2SLS Results 77
16 First Stage Results: Effect of Temperature Shocks on a Product's Debut Sales By Owner Previous Experience 78
17 Reduced Form Results: Effect of Temperature Shocks on a Product's Probability of Reorder By Owner Previous Experience 79
18 First Stage Results: Effect of Temperature Shocks on a Product's Debut Sales By Distance to Owner's Home 80
19 First Stage Results: Effect of Temperature Shocks on a Product's Debut Sales By Accumulated Experience at the Dispensary 81
20 First Stage Results: Effect of Weekly Revenue on a Product's Debut Sales 82
21 2SLS Results: Effect of Weekly Revenue on a Product's Probability of Being Reordered, Through Debut Sales 82

1 INTRODUCTION

This dissertation aims to address the need for transparency in AI systems by bridging the gap between Machine Learning and Econometrics. It compares methods for estimating models of Dynamic Discrete Choice (DDC) from both fields, highlighting the benefits of integrating preference shocks from Econometrics to resolve identification issues in AI/ML methods. Using Monte-Carlo experiments and real-world taxi data, it explores the practical performance of Inverse Reinforcement Learning (IRL) versus the Nested Fixed Point algorithm (NFXP). Additionally, it investigates how small business owners may misattribute noise for profit signals. I use weather shocks as an instrument to analyze decision-making in Washington State marijuana dispensaries.

In Chapter 2, I build a bridge between several methods to estimate models of Dynamic Discrete Choice (DDC): one from the Econometrics literature and two from the AI/ML literature. The method from Econometrics is the Nested Fixed Point (NFXP) algorithm (Rust, 1987). The two methods from the AI/ML literature are both under the umbrella of Max-Margin Inverse Reinforcement Learning: first, the linear programming algorithm presented in Ng and Russell (2000) (Linear Programming IRL) and second, the projection algorithm presented in Abbeel and Ng (2004) (Projection IRL). Using Monte-Carlo experiments on three different GridWorld environments, I show that IRL's wind shocks lead to a fundamental identification issue and that preference shocks from the Econometrics literature resolve that issue. I suggest a way to extend Projection IRL to be able to use preference shocks to gain full identification. I also show that although Projection IRL's accuracy lags slightly behind that of NFXP, Projection IRL significantly reduces the need for solving as many dynamic programming problems –as many as 20 times fewer problems need to be solved compared to NFXP– and can be up to 27 times faster.

To explore the potential benefits and limitations of using Inverse Reinforcement Learning (IRL, Abbeel and Ng 2004) instead of the Nested Fixed Point algorithm (NFXP, Rust 1987) in estimating models of dynamic discrete choice, investigations focusing on theory, simulations, and practical applications are all required. Chapter 3 focuses on the latter (a real-world use case) by making use of publicly available 2013 taxi data to conduct a head-to-head comparison of IRL and NFXP. I estimate three models to determine the payoff for New York City taxi drivers during the morning commute as a function of the borough, the density of workers in a neighborhood, the percentage using personal vehicles to get to work, and the percentage commuting by public transit. This analysis reveals that while IRL's objective function is flatter, allowing for a broader range of acceptable payoff functions, its feature expectation matching technique provides useful feedback on the validity of the chosen smoothing parameter for the kernel density estimation of the transition probability function. This

investigation presents recommendations and highlights potential drawbacks of using IRL instead of NFXP, thereby enhancing our understanding of their practical performance in real-world scenarios.

In Chapter 4, I use a rich dataset on marijuana dispensaries and an instrumental variables approach to identify how small business owners may misattribute noise for profit signals. I analyze thousands of product ordering decisions by Washington State marijuana dispensaries over the first three years of legalization of recreational marijuana. In a traditional economic framework, entrepreneurs make accurate predictions about the relative profitability of certain products over others. But a behavioral update would allow for some degree of inattentiveness by the firm. I use the exogeneity of weather shocks to measure how dispensary owners respond to noise in the signal of a product's profitability. I test whether owners with previous retail experience seem to make more informed decisions, whether owners learn to pay more attention over time, and also whether living further from the dispensary makes it more likely that the owner conflates weather shocks with profitability signals.

2 THE ECONOMETRICS OF INVERSE REINFORCEMENT LEARNING

2.1 Introduction

We are witnessing a turning point in AI adoption. ChatGPT, launched in 2022, was estimated to have hit 100 million monthly active users within two months, earning the title of the fastest-growing consumer application ever (Hu, 2023). AI and Machine Learning (ML) Specialists are projected to be the fastest-growing job field over the next five years (World Economic Forum, 2023). However, the excitement surrounding generative AI has also sparked growing concern. In 2023, 52% of Americans reported feeling more concerned than excited about AI in daily life, an increase from 38% in 2022 (Tyson and Kikuchi, 2023). Indeed, one of the top five new trends in ML is *Responsible AI*: developing guidelines about how to manage risk, design more transparent models, and understand sources of bias (Gartner, 2023). In this paper, I demonstrate that the field of Econometrics has important contributions to make toward developing more responsible AI because Econometricians have a long history of developing methods that yield highly explainable and transparent models, while often sharing many of the same goals as ML.

In this paper, I focus on Inverse Reinforcement Learning (IRL), especially the two fundamental max-margin IRL methods: Linear Programming IRL (Ng and Russell, 2000) and Projection IRL (Abbeel and Ng, 2004). IRL was designed to take, as inputs, data on the states and actions of an agent who is maximizing the sum of discounted per-period payoffs in a Markov Decision Process (MDP). Using that data, IRL attemps to estimate the agent's payoff function: the relative benefit or cost to the agent for being in one state over another. The motivating example (Ng and Russell, 2000) is the autonomous car problem: it has proven to be prohibitively difficult to write down all the intructions for a robot to do a task as complicated as driving safely in a city, but theoretically, we could let a robot watch humans drive safely and infer the underlying payoffs they seem to be maximizing (how they subconsciously make tradeoffs between factors like, for example, maintaining the speed limit and a certain stopping distance).

In the field of Econometrics, the Nested Fixed Point (NFXP) algorithm (Rust, 1987) is the workhorse estimation strategy for fitting models of Dynamic Discrete Choice (DDC). Same as IRL, NFXP takes, as inputs, data on the states and actions of an agent who is maximizing the sum of discounted per-period payoffs in a MDP, and uses that data to try to estimate the agent's payoff function. Although NFXP and max-margin IRL address the same problem, they use fundamentally different approaches. Surprisingly, until now, there has been no discussion on what distinguishes these methods and which might have specific advantages over the other. To date, only

one other piece of research has attempted to bridge the gap between IRL and DDC. Sanghvi et al. (2021) established that IRL and DDC share the same objective and then showed that one IRL approach, Maximum Causal Entropy IRL (Ziebart et al., 2010) is equivalent to NFXP when payoffs are linear in features. My research is unique because, unlike Sanghvi et al., I compare the Nested Fixed Point algorithm not with Maximum Causal Entropy IRL, but with two earlier fundamental IRL methods developed by Ng and Russell, and Abbeel and Ng. These methods are conceptually straightforward, easy to execute, and less computationally demanding, making them potentially appealing for many applications.

In this paper, I study the differences between max-margin IRL and NFXP through Monte-Carlo experiments. I discuss one significant modeling difference, which is how noise in the data is treated: in ML, noise is modeled with wind shocks, while in Econometrics, it is modeled with preference shocks. This difference is actually absolutely critical, because under wind shocks, estimation is limited to partial identification, whereas under preference shocks, estimation can achieve full identification.

My Monte Carlo GridWorld experiments evaluate the performance of NFXP against Projection IRL and Linear Programming IRL, focusing on stress-testing differences in standard applications, particularly with respect to the size of the data, the number of parameters to estimate, and the way noise enters into the data generating process. Finally, I make recommendations on which methods may be best for specific use cases. I suggest extending Projection IRL and Linear Programming IRL to estimate models using data subject to preference shocks. Projection IRL, for instance, achieves up to 99.5% of the accuracy of NFXP at a much lower computational cost, being up to 27 times faster and requiring up to 20 times fewer dynamic programming problems to be solved.

2.2 Background on Methods

In the field of Econometrics, Dynamic Discrete Choice (DDC) is concerned with estimating structural parameters and conducting counterfactual analysis in dynamic settings. These settings feature either competition among multiple agents (entry and exit; product repositioning), or decisions by a single agent, such as whether to renew a patent (Pakes, 1986), or when to replace capital equipment (Rust, 1987). In his seminal paper, John Rust (1987) modeled the capital replacement of city bus engines in order to do structural estimation of the demand for bus engines. Given the prolonged stability in bus engine prices, not enough variation in that variable existed in order to use a regression based method (Rust, 1994). Instead, Rust offered the Nested Fixed Point (NFXP) algorithm to estimate models of DDC, which has since become the standard

DDC estimation strategy.

The NFXP algorithm features an outer loop and a nested inner loop. The outer loop does maximum likelihood estimation through iterative gradient search. The nested inner loop takes a guess for structural parameters, calculates the corresponding payoff function, solves the dynamic programming problem to get the value function, uses that value function combined with an assumption on the distribution of the shocks to calculate the choice probabilities, and combines the choice probabilities with the observed data to determine the likelihood of such a guess for parameter values.

The model assumptions focus on preference shocks: the state variables which are not observed but have an effect on the choice of the agent. As I show in this paper, preference shocks are crucial for the full identification of DDC estimation strategies. The effect of preference shocks must be additively separable from the effect of observed state variables and conditionally independent across time. If assumptions on these unobserved variables hold, NFXP has consistency guarantees and is the most efficient estimator asymptotically, but NFXP is limited by high computational demands from having to solve the dynamic programming problem in each iteration of the inner loop. This limitation has prompted further research on DDC methods (Hotz and Miller, 1993). In this paper, I show that two other algorithms exist from the Machine Learning literature that are computationally cheaper than NFXP. With some adjustments to incorporate preference shocks, they can be used to estimate DDC models. But without preference shocks, they are only partially identified.

The IRL literature in Machine Learning originates from Ng and Russell (2000). In that paper, authors Andrew Ng and Stuart Russell proposed a *Linear Programming IRL* algorithm to estimate payoffs in a Markov Decision Process given an optimizing agent's trajectory data and a model of the environment. The authors' goal was that IRL would one day be able to solve the self-driving car problem. Their idea was that it is too difficult to write down a set of policy rules for a robot to do tasks as complicated as driving a car safely, but perhaps a robot could watch a driver handle a number of traffic situations and infer the underlying payoff function which the driver is maximizing. Then the robot would have the simple task of solving the Markov Decision Process (MDP) under that payoff function for its optimal policy rule. In this way, the robot might be able to drive safely even in situations which are very different from the training dataset.

The key idea in Ng and Russell's Linear Programming IRL algorithm is to match the given dataset with simulated data according to *feature expectations*: a discounted tally of the number of times the agent was observed with each feature. Each iteration of the Linear Programming IRL algorithm includes making a guess for the payoff function and solving the dynamic programming problem for the value function, just like the inner loop of the NFXP algorithm. From there, Linear Programming IRL diverges from NFXP: it uses the value function to simulate a synthetic dataset, calculates the feature

expectations of that synthetic dataset, and defines the *margin* m as the difference between the synthetic feature expectations and the feature expectations of the original dataset. Over successive iterations of the algorithm, the margin is an accumuluted sum of all past margins, and the next guess for the payoff function $\pi(s)$ is the one which maximizes $m^T\pi(s)$ (a linear programming problem). In this paper, I show that only corner solutions are possible in the linear programming problem, and as a result, the algorithm is limited to making estimates on the set {-1, 0, 1}.

The Ng and Russell (2000) model differs from the model in Rust (1987) primarily in how it explains noise: idiosyncratic discrepancies between optimal policies predicted by the payoff function and those observed in the data. Rust models this noise as originating from preference shocks: random shocks drawn each period added to the payoff of taking each action. Ng and Russell, on the other hand, model noise through *wind shocks* with a wind coefficient that is a model parameter: under a wind coefficient of 0.7 for example, the agent takes an action according to its optimal policies with a probability of 0.7, and with a probability of 0.3, she samples randomly from all possible actions. The main contribution of my paper is to show that the distinction between wind shocks and preference shocks is significant and it has important implications for the full identification of the estimation strategy.

Andrew Ng then published a paper describing a second IRL algorithm with Pieter Abbeel (Abbeel and Ng, 2004), this time using projection instead of linear programming. In this paper, I refer to this algorithm as *Projection IRL*. Each iteration of this algorithm takes a guess for the payoff function and solves the dynamic programming problem for the value function, just like NFXP and Linear Programming IRL. Then just like Linear Programming IRL, Projection IRL uses simulation to create a synthetic dataset and calculates its feature expectations. From there, Projection IRL diverges from Linear Programming IRL: it uses projection and previous synthetic feature expectations to approach the expert feature expectations. The margin is the difference between the projection and the expert feature expectations, and the algorithm stops when the length of the margin becomes small enough. The next guess for the payoff function is the direction of the margin. Abbeel and Ng do not present an alternative to modeling noise through wind shocks. However, in this paper I demonstrate that by making certain adjustments to the Projection IRL algorithm, preference shocks can be effectively applied within Projection IRL. This adaptation significantly enhances its identification capabilities.

Recent studies in Inverse Reinforcement Learning (IRL) have expanded its applications across various fields. These include helicopter aerobatics (Abbeel et al., 2006), navigating crowds and pedestrians in a socially compliant way in order to avoid hindering pedestrians (Ziebart et al. 2009, Kretzschmar et al. 2016, and Kim and Pineau 2016), playing video games (Tucker et al., 2018), boat sailing (Neu and

Szepesvari, 2012), robotic legged locomotion (Ratliff et al., 2009), and energy efficient driving (Vogel et al., 2012).

2.3 Model

Consider a single-agent discrete decision process where an expert is observed over time in states $s_t \in S = \{1, ..., L\}$ taking actions $a_t \in A = \{0, 1, ..., K\}$ so that the data takes the form $(a_t, s_t)_{t=1}^T$. Each period, the agent is endowed with a state variable s_t and a vector of pay-off shocks $\varepsilon_t \in \mathbb{R}^K$ (the pay-off shocks are ultimately not observable to the econometrician). The pay-off shocks are independently and identically normally distributed. There are K pay-off shocks but $K + 1$ actions because only the differences between pay-off shocks are relevant, so the first action's pay-off shock can be normalized to zero (McFadden, 1973).

The agent receives a pay-off each period according to a_t, s_t, and ε_t equal to

$$\pi(s_t) + \sum_{k=1}^{K} \varepsilon_{t,k} \cdot \mathbf{1}(a_t = k) \tag{1}$$

and then the state transition probability matrix G governs how the agent moves to s_{t+1}. G is made up of elements $g(a_t, s_t, s_{t+1})$, which is the probability the agent moves from s_t to s_{t+1} after having taken action a_t. The pay-off for the entire process is the discounted sum of per-period pay-offs where the discount factor is given by $\beta < 1$. I assume the transition process is Markovian so that I can abstract from calendar time. Let $p(s)$ be the vector of choice probabilities, so that $p(a|s)$ is the probability the agent will choose action a in state s. Then the value function is given by

$$V(s, \varepsilon) = \max_a \left[\pi(s) + \sum_{k=1}^{K} \varepsilon_k \cdot \mathbf{1}(a = k) + \beta \mathbb{E}[V(s', \varepsilon')|s, a] \right] \tag{2}$$

$$= \max_a \left[\pi(s) + \sum_{k=1}^{K} \varepsilon_k \cdot \mathbf{1}(a = k) + \beta \sum_{s'=1}^{S} g(a, s, s') V(s', \varepsilon) \right] \tag{3}$$

2.3.1 Nested Fixed Point Implementation Details

I use the NFXP algorithm outlined in Rust (1987). The objective is to estimate the model parameters θ which describe the payoff $\pi(a, s)$.

NFXP requires two sets of summary statistics from the discrete choice data: an approximation of the the probability transition matrix G and the observed choice frequency $n_{ks} \equiv \sum_t I(a_t = k, s_t = s)$.

The algorithm has an inner and outer loop, thus the name "nested". The outer loop uses maximum likelihood estimation to search for the values of θ which make the

observed n_{ks} most likely. That log-likelihood is defined as

$$LL = \sum_{k,s} n_{ks} \cdot \log(p(a = k|s)), \tag{4}$$

where the choice probabilities of taking action a in state s, $p(a|s)$, are calculated for the values θ in an inner loop.

The inner loop takes θ and uses a value iteration technique to find the value function $V(s; p)$. Then to find the choice probabilities $p(a|s)$, recall that shocks ε are assumed to be Normally distributed. So after observing the realizations of shocks ε for a period, the probability that action a_1 for example has the highest value out of all other possible actions is

$$\begin{aligned} p(a|s) = P(&V(s; a = 1) + \varepsilon_1 > V(s; a = 2) + \varepsilon_2 \quad \cap \\ &V(s; a = 1) + \varepsilon_1 > V(s; a = 3) + \varepsilon_3 \quad \cap \\ &... \quad \cap \\ &V(s; a = 1) + \varepsilon_1 > V(s; a = K) + \varepsilon_K) \end{aligned} \tag{5}$$

And rearranging

$$\begin{aligned} p(a|s) = P(&\varepsilon_1 - \varepsilon_2 > V(s; a = 2) - V(s; a = 1) \quad \cap \\ &\varepsilon_1 - \varepsilon_3 > V(s; a = 3) - V(s; a = 1) \quad \cap \\ &... \quad \cap \\ &\varepsilon_1 - \varepsilon_K > V(s; a = K) - V(s; a = 1) \end{aligned} \tag{6}$$

I assume that shocks ε are independent and normally distributed with equivalent means and variances of 1. Then the differences $\varepsilon_1 - \varepsilon_2$, $\varepsilon_1 - \varepsilon_3$, ..., and $\varepsilon_1 - \varepsilon_K$ are also normally distributed, they have mean zero, and they each have a variance of 2. I therefore find the joint probability $p(a|s)$ using a multivariate normal distribution package using the lower integration limits of
$(V(s, a = 2), V(s, a = 3), ..., V(s, a = K)) - V(s, a = 1)$. Note that
$Cov(\varepsilon_1 - \varepsilon_2, \varepsilon_1 - \varepsilon_3) = Cov(\varepsilon_1, \varepsilon_1) = Var(\varepsilon_1) = 1$, so the multivariate normal variance-covariance matrix I use is this $K \times K$ matrix.

$$\begin{bmatrix} 2 & 1 & 1 & \cdots \\ 1 & 2 & 1 & \cdots \\ 1 & 1 & 2 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{7}$$

If instead you assume that the shocks are, like a logit, distributed Extreme Value Type 1, computing the conditional probabilities is even simpler, following Rust (1987),

$$p(a|s) = \frac{exp(\pi(a,s) + \beta \sum_{s' \in S} g(a,s,s')V(s';p))}{\sum_{a \in A} exp(\pi(a,s) + \beta \sum_{s' \in S} g(a,s,s')V(s';p)}. \tag{8}$$

2.3.2 Linear Programming IRL Implementation Details

The key to the IRL algorithms from Ng and Russell (2000) and Abbeel and Ng (2004) is that the true pay-off function $\pi(s)$ might be recovered by simulating synthetic trajectories under different guesses for payoff functions and trying to match feature expectations μ.

Assume the pay-off function $\pi(s)$ is a function of only s and not a function of action a, and all preference shocks are zero. Instead of shocks, the agent is blown as if by wind a random direction w.p. 0.3. Otherwise, the agent moves in order to maximize value. Let $\{s_1, s_2, ...\}$ represent the process following policy $p(a|s)$. Then the value function evaluated at s_1 is

$$V(s_1;p) = \sum_{t=1}^{\infty} \beta^{t-1} \pi(s_t)$$

Let the *feature expectations* of $\{s_1, s_2, ..., s_T\}$ in the simplest case where each state is its own feature be $\mu(s) = \sum_{t=1}^{T} \beta^{t-1} 1(s_t = s)$. Then the key to IRL is that the $V(s_1, p)$ can be estimated by

$$\hat{V}(s_1;p) = \mu(s)^T \cdot \pi(s)$$

The expert's policy p^* should maximize $\hat{V}(s_1, p)$, so compared to any other policy p':

$$\begin{aligned}
\hat{V}(s_1|p^*) &\geq \hat{V}(s_1|p') \\
\mu^*(s)^T \pi(s) &\geq \mu'(s)^T \pi(s) \\
\mu^*(s)^T \pi(s) - \mu'(s)^T \pi(s) &\geq 0 \\
(\mu^*(s) - \mu'(s))^T \pi(s) &\geq 0
\end{aligned}$$

For Linear Programming IRL (Ng and Russell, 2000), let the feature expectation margin be $m(s) = \mu^*(s) - \mu'(s)$. The pay-off function $\pi(s)$ that best rationalizes $\mu^*(s)$ compared to $\mu'(s)$ is found by

$$\begin{aligned} \max_{\pi(s)} \quad & m(s)^T \pi(s) \\ \text{s/t} \quad & |\pi(s)| \leq 1 \quad \text{for } s = 1, ..., L \end{aligned}$$

Which is solved by

$$\pi(s) = \begin{cases} -1 & \text{if } m(s) < 0 \\ 1 & \text{if } m(s) > 0 \\ \text{any in } [-1, 1] & \text{if } m(s) = 0 \end{cases}$$

In order to estimate $\pi(s) = 0$ instead of only -1 and 1, add L1 regularization:

$$\begin{aligned} \max_{\pi(s)} \quad & m(s)^T \pi(s) - 1.2 \sum_s |\pi(s)| \\ \text{s/t} \quad & |\pi(s)| \leq 1 \quad \text{for } s = 1, ..., L \end{aligned}$$

Which makes the solution:

$$\pi(s) = \begin{cases} -1 & \text{if } m(s) < -1.2 \\ 1 & \text{if } m(s) > 1.2 \\ 0 & \text{if } -1.2 < m(s) < 1.2 \end{cases}$$

Using a constrained optimization package instead of this decision rule leads to artifacts which may estimate $\pi(s)$ falsely to be outside of the set $\{-1, 0, 1\}$. So to summarize, the Ng and Russell (2000) Linear Programming IRL algorithm is:

1. Use data $\{a_t, s_t\}_{t=1}^T$ to estimate the observed feature expectations μ^*. Generate a random pay-off function π_1 and set $i = 1$ and $m = 0$.

2. Use π_i to find the value function, generate expert trajectories, and feature expectations μ_i. Let $m = m + \mu^* - \mu_i$.

3. Set $i = i + 1$ and find π_i: $\max_{\pi_i} m\pi_i - 1.2 \sum_s |\pi_i|$. Jump to step 2, or quit after a large number of iterations (convergence is not guaranteed).

2.3.3 Projection IRL Implementation Details

Projection IRL (Abbeel and Ng, 2004) is similar to Linear Programming IRL except in the way that it updates guesses for the payoff function. It uses Projection instead of Linear Programming to try to find payoff functions which generate simulated datasets with similar feature expectations compared to the expert. This is how the Projection algorithm works:

1. Initialize the algorithm by calculating expert feature expectations μ_E the same way as in Linear Programming IRL: $\mu_E = \sum_{t=1}^{T} \beta^{t-1} 1(s_t = s)$. Let $i = 0$ and make a random guess for the payoff function and call it π_0.

2. Solve the Dynamic Programming Problem: for the current payoff function guess π_i, solve the dynamic programming problem to find the value function. Use this value function to generate a synthetic dataset for an agent optimizing under π_i. Calculate the feature expectations for this synthetic dataset, denoted as μ_i.

3. For $i = 0$, calculate the margin and update the payoff function: compute the margin $m_0 = \mu_E - \mu_0$. Update the iteration counter $i = 1$ and make the next guess for the payoff function $\pi_1 = m_0$. Find μ_1 using the same procedure as in step 2.

4. Projection Step: Update the guess towards the expert's feature expectations by projecting onto the space of possible feature expectations. Specifically, adjust $\bar{\mu}_i$ as follows: $\bar{\mu}_i = \mu_{i-1} + projection(m_{i-1}, \mu_i - \mu_{i-1})$. The next guess for the payoff function then becomes $\pi_i = \mu_E - \bar{\mu}_i$.

5. Iterate: Update the iteration counter $i = i + 1$. Calculate μ_i, the feature expectations implied by the updated payoff function guess π_i, as done in step 2. Repeat steps 4 and 5 until the magnitude of π_i becomes smaller than a predefined threshold ε. The final estimate can undergo an affine transformation between -1 and 1.

The algorithm works smoothly under the assumption of wind shocks. But using preference shocks instead introduces a complication. Specifically, as the magnitude of the update to the payoff function π_i decreases toward zero, synthetic feature expectations μ_i converge to the feature expectations under a payoff function of all zeros. Under preference shocks with mean 0 and standard deviation 1, this looks like the agent is wandering the grid completely randomly. This undermines the algorithm's ability to align the synthetic agent's feature expectations with those of the expert. To address this, I included an affine transformation of the updated payoff function guess between -1 and 1 *every iteration*, not just for the final iteration (after comparing its magnitude to ε).

I also observed an improvement in performance if, at the end, Projection IRL returns the payoff function which generated feature expectations closest to the expert, instead of returning the smallest $\mu_E - \bar{\mu}_i$. The rationale is that $\mu_E - \bar{\mu}_i$ is a corrective adjustment to the search direction rather than the most accurate estimate of the payoff function that could replicate the expert's feature expectations.

2.4 Methods

Each experiment is on a 5x5 GridWorld where features are comprised of one or more cells, but for simplicity, no cells belong to more than one feature. Every cell in a feature has the same payoff.

The first step is to generate the expert data:

1. Select the expert agent's starting position randomly out of the 25 cells on the grid.
2. The expert moves according to either "wind shocks" or "preference shocks", depending on the experiment trial. If the trial uses preference shocks, then each time step, five i.i.d. Normal (0, 1) shocks are drawn and added to the value of taking each action and the agent takes the action with the highest value. If the trial uses wind shocks with a wind coefficient of 0.7, the agent takes the action with the highest value with probability 0.7, and with probability 0.3, she selects among all actions with equal probability. The expert always has five actions available: up, down, left, right, or to stay in the same place. If the expert moves toward the outer boundary of the grid, they stay in the same place. The discount factor is 0.95.
3. Truncate the trajectory after 10 time steps.
4. Generate `nt` such trajectories and evaluate the performance of NFXP, Linear Programming IRL, and Projection IRL on each of these datasets.

Repeat the steps above for 20 trials per experiment using increasing amounts of data (let `nt` be 10, 50, and finally 100).

When the NFXP algorithm is used, I apply an affine transformation to bound the final NFXP estimate between -1 and 1. Note that the NFXP algorithm always models the expert data as having come from the preference shock process, so when the expert data is generated with the windy process, NFXP is misspecified. Linear Programming IRL and Projection IRL uses simulation in its estimation strategy, so when the expert is subject to wind shocks, both IRL algorithms simulate data subject to wind shocks. And when the expert is subject to preference shocks, both IRL algorithms simulate data subject to preference shocks.

2.5 Results

Table 1 presents summary statistics (means and standard deviations in parentheses) from the first experiment involving a 3-parameter GridWorld. For a more detailed illustration of each trial's estimate, see 9. Under preference shocks, NFXP is able to estimate the true values of the payoff function with accuracy, even with a limited dataset of only 100 points. As the size of the data increases, NFXP's estimations get even closer to the true payoff function values and estimates are more consistent. Projection IRL is almost as accurate and consistent as NFXP, and takes an average of 2-31 seconds to complete with an average of 20-51 iterations required, compared to NFXP's 55-65 seconds and 406-417 iterations required. Linear Programming IRL is not able to estimate values outside the set {-1, 0, 1}, yet under preference shocks, it is fairly accurate and its estimates improve with more data.

But under wind shocks (see the right hand side of Table 1), NFXP is no longer able to estimate the payoff function with accuracy: it estimates around (1, -.9, -.8) instead of the true values of (1, .95, -1). Projection IRL fairs a little better, estimating around (1, .6, -1), and LP IRL settles on (1, 0, -1) most frequently.

The same pattern presents in Table 2 for the 4-parameter GridWorld. Here, the true payoff function is (1, .99, .94, -1), which both NFXP and Projection IRL are able to estimate accurately and consistently under preference shocks, especially as the amount of data increases. Projection IRL takes, on average, between 10 and 55 seconds to complete, requiring 70 to 94 iterations. NFXP, on the other hand, takes 62 to 73 seconds to complete and does 479-501 iterations. Linear Programming IRL is less successful here with a tendency to estimate (0, 0, 1, -1) (see 10 for more detail). And under wind shocks, again NFXP is no longer able to make accurate estimates, and neither either of the IRL algorithms.

For the 5-parameter GridWorld, the same pattern appears again (see Table 3): under preference shocks, NFXP and Projection IRL do well and NFXP has a slight edge. Linear Programming IRL is a good estimator as well, estimating parameters rounded to the nearest integer: it estimates (1, 0, 0, 0, -1) instead of the true values of (1, .36, .36, .33, -1). And under wind shocks, Projection IRL is the only algorithm that gets close, but it still is not able to estimate the payoff function very precisely. One difference from the 3 and 4 parameter experiments is this: Projection IRL does not hold its time and number of iterations advantage over NFXP under preference shocks. NFXP takes 50-64 seconds to complete, doing 486-502 iterations, while Projection IRL takes 27-385 seconds to complete, doing 364-692 iterations.

2.6 Discussion

2.6.1 Estimates Under Preference Shocks

In the case that unobserved variables can be modeled as preference shocks, which method is recommended? Holding constant the data size and the number of parameters to estimate, NFXP has a slight advantage over Projection IRL when it comes to accuracy, across the board. That advantage decreases as the number of parameters to estimate increases or the data size increases. But in many cases, NFXP has a significant disadvantage over Projection IRL in terms of the number of iterations and the time it takes to converge.

So I recommend using Projection IRL in use cases with many parameters to estimate, and NFXP to use elsewhere. Linear Programming IRL can be used in cases where it is advantageous to assume the payoff function is limited to the set {-1, 0, 1}.

2.6.2 Identification Problem from Wind Shocks

In every experiment with wind shocks, I found that NFXP, Linear Programming IRL, and Projection IRL all appeared to be biased estimators of the payoff function when the expert data was generated with wind shocks. In order to explain why, I'll first explain a couple of differences between data generated with wind shocks and data generated with preference shocks. Then I'll explain why wind shocks create identification problems for estimators.

Consider a cliff GridWorld environment (Figure 4). The agent can collect a payoff of 50 and then 100, but only if they are willing to approach the cells with payoff -1000. When the agent is subject to preference shocks (see figure 4), there is no danger because every action that agent makes is purposeful. The agent collects 50 on their way to collecting 100 and only finds themselves in the cells with -1000 when that was their (random) starting position. Preference shocks are distributed N(0, 1), so because of the scale of payoffs on this grid, behavior here is identical to the behavior in the deterministic case where there are no shocks at all.

When the expert is instead subject to wind shocks set to 0.7 (see figure 5), then 30% of the time, the agent can not act with intention and is instead blown randomly by wind. The agent's behavior here is very different from the preference shock case. Note that this agent largely avoids collecting the payoff of 50 on their way to collecting 100 so that they don't have to risk being blown off the edge of the cliff.

When I change the wind coefficient to 0.5 (50% of the time, the agent is blown in a random direction: see figure 6), the agent's best policy shifts again. Now, the agent does not seem to have the goal of collecting 50 *or* 100, but instead, it stays near the bottom of the board, as far away as possible from the -1000 payoff cells, as if for safety. Preference shocks and wind shocks can create different behaviors from the agent. In the

preference shock case, you need to know the variance of the preference shocks in order to estimate the scale of the payoffs correctly (assuming a variance of 1 will still allow you to correctly estimate relative values of the payoffs). But in the wind shock case, different wind coefficients can yield very different behaviors and it is crucial to either estimate it correctly or give your algorithm correct values for that wind coefficient.

Consider a new GridWorld environment with the same shape as experiment 1. In all three of figures 7a, 7b, and 7c, the agent is subject to preference shocks distributed N(0, 1). In figure 7a, the payoff function is (1, 0.5, 0). The result is that the agent has a slight preference for the center of the grid. As the scale of the payoff function increases to (2, 1, 0) in figure 7b, the agent's preference for the center cell and, sometimes, left and right of center intensifies. As the scale of the payoff function again increases to (10, 5, 0) in figure 7c, the agent works to get to the center and then never wanders away from it. Although the payoff functions in these three figures are similar, under preference shocks, the behaviors they elicit are very different.

On the other hand, in figures 8a, 8b, and 8c, the agent is subject to wind shocks with a wind coefficient of 0.7. As the scale of the payoff function increases again from (1, .5, 0), to (2, 1, 0) to (10, 5, 0), the behavior of the agent *does not change*. Here, 30% of the time, the agent is blown randomy by wind, and 70% of the time, the agent is able to act according to its best policy. And the best policy for the agent is independent of the scale of the payoff: as long as the payoff function is increasing toward a single maximum, the agent can't improve on a policy that includes reaching that maximum (from the left or right if it doesn't take extra steps to do so), and once in the center, to stay there. This is the root of the identification issue that comes with wind shocks. Because the agent's behavior does not change when the scale of the payoff function changes, under wind shocks, relative values for the payoff function can never be identified precisely: ony ordinal values can sometimes be estimated.

Wind shocks make identification impossible because when the expert takes non-optimal actions randomly, there is an upper bound to the amount of information the Econometrician can possibly extract from the data. Best policies may be identifiable (letting ordinal identification sometimes be possible), but if the agent takes all other actions in a random uniform way, then it is unknown which action may have been second- or third-best, and it is unknown how much worse the second-best (or third-best) option may have been compared to the best options. When the agent is blown by wind, actions give the Econometrician no information about the relative values for $\pi(s')$, and estimates will be biased no matter the estimation strategy.

The Econometrics literature (and NFXP) solves this identification problem by assuming that for each time step, actions have preference shocks which are not observable to the econometrician, but the expert always takes the action with the highest value *combined* with the shock. In this way, the best option is taken with the

highest probability, and then the second best option is taken with the next highest probability, onward. And even more than that, the difference between the relative action frequencies gives the econometrician useful information about how much worse the second-best options are from the first, and so on. With preference shocks, the expert takes action a in a way that is proportional to $V(s';p)$, so actions give the econometrician the information required about relative values for $\pi(s')$ so that identification is possible.

2.7 Conclusion

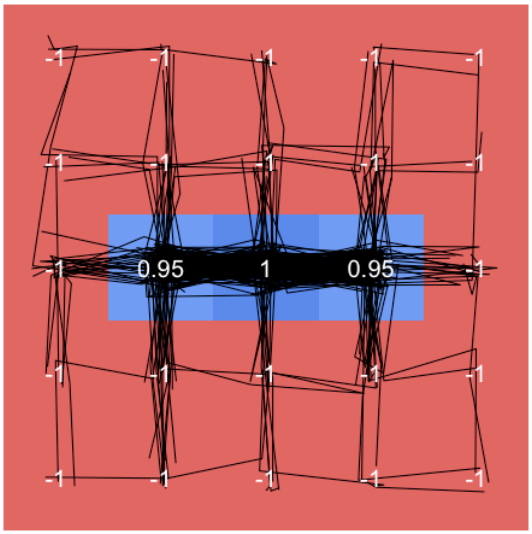
This paper explores three methods for estimating payoffs in sequential discrete choice models, focusing on their respective advantages and disadvantages. I compared Inverse Reinforcement Learning (IRL) methods—Linear Programming IRL and Projection IRL—with the Nested Fixed Point (NFXP) algorithm from the Econometrics literature.

I found that wind shocks commonly used in Machine Learning lead to partial identification of the payoff function, whereas preference shocks from Econometrics allow for full identification in both Linear Programming IRL and Projection IRL. I showed that Linear Programming IRL is limited to identifying payoffs in the set {-1, 0, 1}. Despite Projection IRL's slightly lower accuracy compared to NFXP, it offers significant computational advantages, requiring up to 20 times fewer dynamic programming problems to be solved and delivering results up to 27 times faster. These findings suggest that Projection IRL may be preferable in situations with large state spaces, while Linear Programming IRL can be useful when the payoff function is limited to a specific set of values.

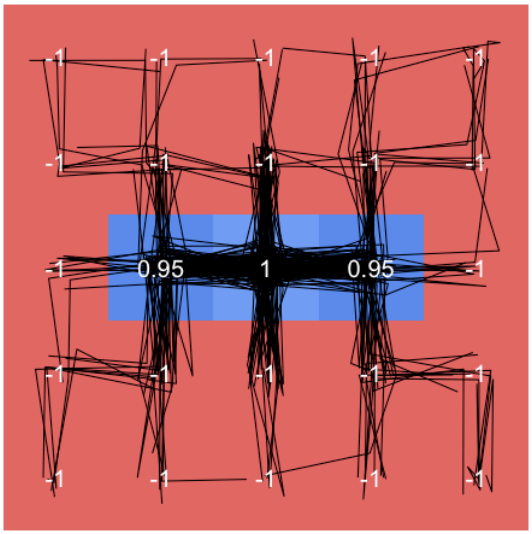
2.8 Tables and Figures

Figure 1: Simulated Data on a 3-Parameter GridWorld

(a) Preference Shock Trajectories



(b) Wind Shock Trajectories



Notes: Figures 1 (a) and (b) illustrate typical simulated paths of an agent on a 5x5 GridWorld where the agent is tasked with maximizing her cumulative discounted payoffs over time. The grid has a defined payoff structure that is known to the agent: the center cell has a payoff of 1 unit, the cells immediately to the left and right of the center have a payoff of 0.95 units, and all other cells have a payoff of -1 units. Each trajectory consists of 10 steps, taken by an agent that starts from a random location and who can move up, down, left, right, or who can stay in the same place. If the agent hits the grid's outer edge, she remains in place. Figures 1 (a) and (b) both illustrate 50 such trajectories.

The trajectories in **Figure 1 (a)** are subject to preference shocks, which means that at every time step, the agent receives a set of five independent and identically distributed shocks, drawn from a normal distribution with mean 0 and standard deviation 1, one for each potential action. The agent then chooses the action that is expected to yield the highest total discounted payoff, shock included. This process generates trajectories with the pattern that the agent is attracted toward the center cell and, once there, moves predominantly left or right of center, with fewer movements up or down from the center.

The agent in **Figure 1 (b)**, on the other hand, is subject to wind shocks with a wind coefficient of 0.7. That means that there is a 70 percent chance that the agent can move with her intention (selecting from optimal policies), and a 30 percent chance that a random gust of 'wind' will blow her in a random direction. The agent's strategy still involves moving toward the center, often via the cells to the left and right of the center. However, once in the center, she stays there unless blown by wind. The agent's movements to the left and right of the center therefore occur just as frequently as movements up and down from the center, which contrasts with Figure 1. Note that trajectory data in Figures 1 (a) and (b) seem similar on the surface, except for the activity going up and down from the center cell. Although wind shocks can sometimes create data that looks superficially similar to preference shock data, wind shocks prevent the full identification of all estimation strategies (see Table 1).

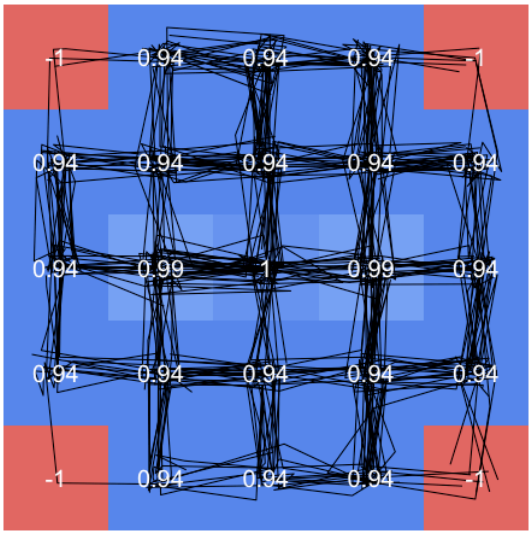
Table 1: Results for the Estimation of a 3-Parameter GridWorld

		Preference Shocks			Wind Shocks	
N = 100	True Values	NFXP	Projection IRL	LP IRL	NFXP	Projection IRL	LP IRL
Feature 1	1	0.95 (0.09)	0.89 (0.19)	0.8 (0.41)	1 (0)	0.97 (0.06)	1 (0)
Feature 2	0.95	0.9 (0.12)	0.88 (0.18)	0.9 (0.31)	-0.77 (0.25)	0.13 (0.79)	0.05 (0.69)
Feature 3	-1	-1 (0)	-1 (0)	-1 (0)	-0.82 (0.22)	-0.76 (0.52)	-0.95 (0.22)
Time elapsed (seconds)		55.2 (12.41)	2.06 (0.95)	21.53 (4.29)	57.32 (11.8)	0.66 (0.38)	6.92 (0.75)
Number of iterations		406.25 (49.04)	20.05 (8.44)	204.8 (8.37)	395.6 (88.89)	19.2 (11.51)	212.65 (16.59)
N = 500							
Feature 1	1	0.98 (0.04)	0.95 (0.08)	0.9 (0.31)	1 (0)	0.98 (0.04)	1 (0)
Feature 2	0.95	0.91 (0.11)	0.89 (0.14)	0.8 (0.41)	-0.76 (0.15)	0.64 (0.54)	0.55 (0.51)
Feature 3	-1	-1 (0)	-1 (0)	-1 (0)	-0.91 (0.17)	-1 (0)	-1 (0)
Time elapsed (seconds)		65.64 (8.77)	18.67 (18.09)	90.67 (4.06)	71.44 (5.06)	0.75 (0.75)	6.64 (0.57)
Number of iterations		408.35 (58.54)	40.65 (39.02)	201.25 (0.72)	435.25 (31.61)	22.45 (22.86)	207.05 (12.95)
N = 1000							
Feature 1	1	0.99 (0.03)	0.97 (0.05)	0.95 (0.22)	1 (0)	0.94 (0.07)	1 (0)
Feature 2	0.95	0.94 (0.05)	0.93 (0.09)	0.95 (0.22)	-0.87 (0.14)	0.58 (0.65)	0.1 (0.31)
Feature 3	-1	-1 (0)	-1 (0)	-1 (0)	-0.81 (0.2)	-1 (0)	-1 (0)
Time elapsed (seconds)		55.36 (8.26)	31.33 (30.12)	121.6 (12.15)	82.03 (3.18)	0.8 (0.96)	6.52 (0.16)
Number of iterations		417.85 (60.07)	51.05 (47.8)	201.15 (0.37)	431.2 (17.94)	24.35 (30.94)	204.75 (4.48)

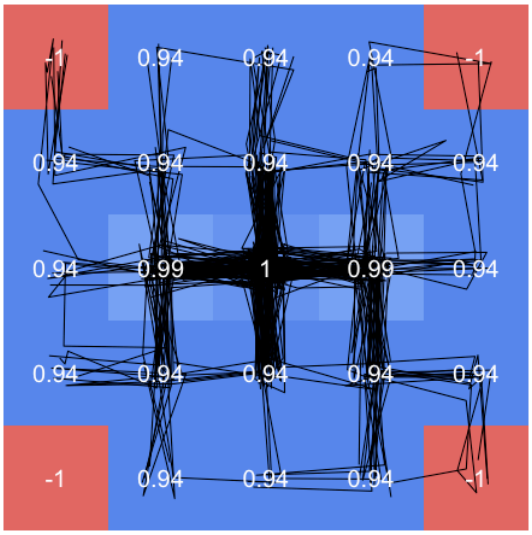
Notes: Table 1 reports the performance of three algorithms—Nested Fixed Point, Linear Programming IRL, and Projection IRL—in estimating the payoffs for cells in the GridWorld environment from Figures 1 (a) and (b). These algorithms are provided with a detailed model of the GridWorld environment, including its movement mechanics, a discount factor of 0.95, and a wind coefficient of 0.7. They also have access to data in the form of the agent's trajectories, and they are asked to use the model and the data to estimate the payoff function. The true payoff function is (1, 0.95, -1) by feature. Each algorithm's estimates undergo an affine transformation for standardization between 1 and -1. The table shows average estimates over twenty trials for each estimation method, data size (N = 100, 500, and 1000 observations), and using trajectories subject to preference shocks versus wind shocks. The sample standard deviations for estimates over the twenty trials is noted in parentheses. If the method produced an estimate of 1 for a certain feature in all twenty trials, the table will record 1 (0) because the standard deviation is zero.

Figure 2: Simulated Data on a 4-Parameter GridWorld

(a) Preference Shock Trajectories



(b) Wind Shock Trajectories



Notes: Figures 2 (a) and (b) illustrate typical paths of an agent on a 4-parameter GridWorld environment under preference shocks and then under wind shocks. The payoff function is (1, .99, .94, -1), defined on the set of features (center, left and right of center, all else except corners, corners). Under preference shocks, the agent wanders almost uniformly around the grid, avoiding only the corners. Under wind shocks, the agent focuses on the center cell unless blown by wind.

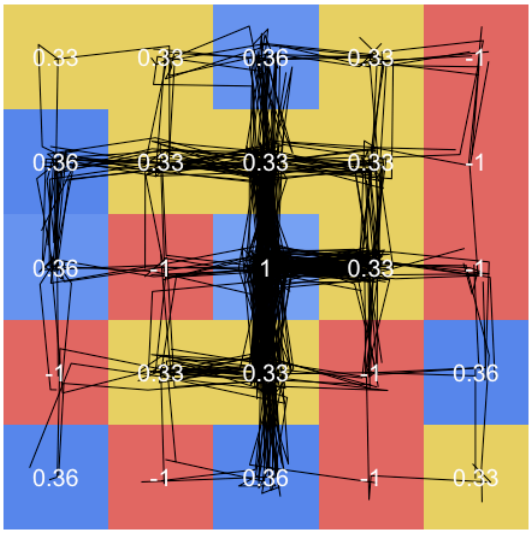
Table 2: Results for the Estimation of a 4-Parameter GridWorld

		Preference Shocks			Wind Shocks	
N = 100						
	True Values	NFXP	Projection IRL	LP IRL	NFXP	Projection IRL	LP IRL
Feature 1	1	0.8 (0.29)	0.41 (0.73)	0.2 (0.62)	1 (0)	0.99 (0.04)	1 (0)
Feature 2	0.99	0.81 (0.21)	0.57 (0.59)	0.1 (0.55)	-0.53 (0.44)	0.57 (0.52)	0.25 (0.55)
Feature 3	0.94	0.92 (0.09)	0.89 (0.14)	0.95 (0.22)	-0.69 (0.44)	0.07 (0.82)	-0.5 (0.69)
Feature 4	-1	-1 (0)	-0.8 (0.49)	-0.95 (0.22)	-0.24 (0.67)	-0.7 (0.54)	-0.55 (0.51)
Time elapsed (seconds)		68.55 (11.3)	10.32 (17.29)	22.97 (1.9)	71.29 (10.13)	2.42 (2.13)	6.72 (0.46)
Number of iterations		484.9 (76.25)	92.5 (156.52)	206.2 (8.79)	481.1 (71.28)	68 (60.25)	206 (6.85)
N = 500						
Feature 1	1	0.94 (0.08)	0.8 (0.19)	0.5 (0.51)	1 (0)	0.98 (0.03)	1 (0)
Feature 2	0.99	0.93 (0.08)	0.93 (0.08)	0.45 (0.51)	-0.73 (0.31)	0.68 (0.44)	0.35 (0.59)
Feature 3	0.94	0.93 (0.07)	0.92 (0.09)	0.9 (0.31)	-0.7 (0.22)	0.04 (0.77)	-0.7 (0.47)
Feature 4	-1	-1 (0)	-1 (0)	-1 (0)	-0.26 (0.51)	-0.84 (0.32)	-1 (0)
Time elapsed (seconds)		73.21 (11.94)	31.43 (17.39)	87.75 (8.77)	82.01 (6.84)	4.29 (4.81)	6.39 (0.23)
Number of iterations		479.3 (78.4)	70.6 (38.14)	202.35 (1.14)	490.3 (41.76)	131.6 (148.04)	201.25 (0.55)
N = 1000						
Feature 1	1	0.94 (0.09)	0.9 (0.16)	0.55 (0.51)	1 (0)	0.99 (0.02)	1 (0)
Feature 2	0.99	0.96 (0.05)	0.93 (0.09)	0.6 (0.5)	-0.8 (0.16)	0.56 (0.58)	0.35 (0.49)
Feature 3	0.94	0.96 (0.03)	0.93 (0.07)	0.9 (0.31)	-0.87 (0.19)	-0.08 (0.77)	-0.8 (0.41)
Feature 4	-1	-1 (0)	-1 (0)	-1 (0)	-0.36 (0.29)	-0.84 (0.3)	-0.6 (0.5)
Time elapsed (seconds)		62.77 (2.35)	55.47 (25.45)	116.44 (5.43)	86.09 (17.27)	1.74 (1.32)	6.12 (0.48)
Number of iterations		501.7 (3.45)	94.55 (40.33)	202.1 (1.02)	460.3 (104.61)	55.45 (41.26)	202.05 (1.73)

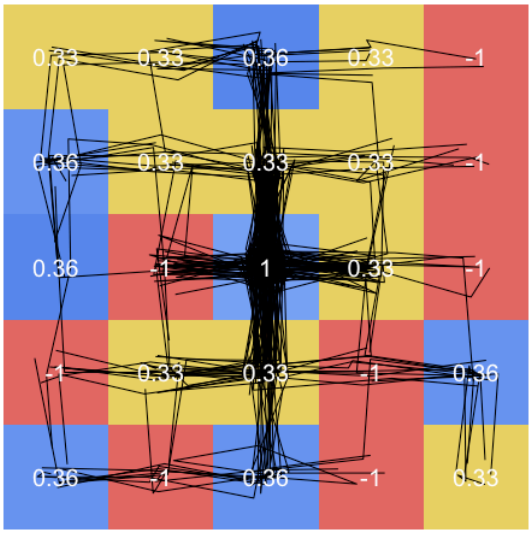
Notes for Table 2: Table 2 reports the performance of three algorithms—Nested Fixed Point, Linear Programming IRL, and Projection IRL—in estimating the payoffs for cells in the GridWorld environment from Figures 2 (a) and (b). The true payoff function is (1, 0.99, .94, -1) by feature. The table shows average estimates and standard deviations in parentheses over twenty trials for each estimation method, data size (100, 500, and 1000 observations), and using trajectories subject to preference shocks versus wind shocks.

Figure 3: Simulated Data on a 5-Parameter GridWorld

(a) Preference Shock Trajectories



(b) Wind Shock Trajectories



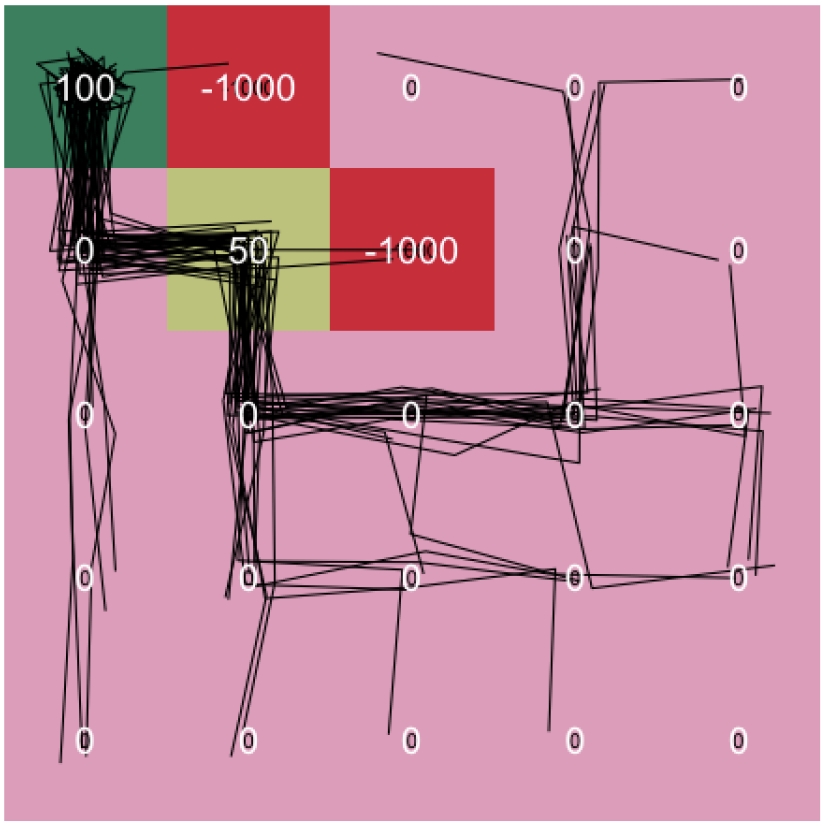
Notes: Figures 3 (a) and (b) illustrate typical paths of an agent on a 5-parameter GridWorld environment under preference shocks and then under wind shocks. The payoff function is (1, .36, .36, .33, -1). Under preference shocks, the agent wanders around the grid, often avoiding cells with a payoff of -1 and focusing slightly more on the center cell with a payoff of 1. Under wind shocks, the agent focuses only on the center cell, and once there, is blown to the left of center (receiving a payoff of -1) as many times as she is blown other directions.

Table 3: Results for the Estimation of a 5-Parameter GridWorld

		Preference Shocks			Wind Shocks		
N = 100	True Values	NFXP	Projection IRL	LP IRL	NFXP	Projection IRL	LP IRL
Feature 1	1	1 (0)	1 (0)	0.95 (0.22)	1 (0)	1 (0)	1 (0)
Feature 2	0.36	0.38 (0.36)	0.47 (0.32)	-0.1 (0.31)	0.08 (0.53)	-0.24 (0.68)	-0.35 (0.49)
Feature 3	0.36	0.43 (0.42)	0.14 (0.58)	-0.3 (0.47)	0.04 (0.58)	0.17 (0.74)	-0.35 (0.49)
Feature 4	0.33	0.38 (0.17)	0.48 (0.25)	0.3 (0.47)	-0.38 (0.42)	0.54 (0.53)	-0.35 (0.49)
Feature 5	-1	-1 (0)	-0.96 (0.1)	-1 (0)	-0.83 (0.36)	-0.26 (0.65)	-0.9 (0.31)
Time elapsed (seconds)	NA	50.81 (3.27)	27.35 (27.12)	14.68 (0.83)	75.97 (10.25)	6.19 (3.74)	6.63 (0.37)
Number of iterations	NA	502.3 (1.42)	364.05 (344.2)	204.05 (8.15)	484.85 (82.13)	183.15 (112.98)	202.3 (3.23)
N = 500							
Feature 1	1	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
Feature 2	0.36	0.3 (0.22)	0.2 (0.36)	0 (0)	0.1 (0.21)	0.24 (0.58)	-0.15 (0.37)
Feature 3	0.36	0.33 (0.16)	0.31 (0.25)	0 (0)	0.15 (0.23)	0.18 (0.4)	0 (0)
Feature 4	0.33	0.34 (0.08)	0.36 (0.07)	0 (0)	-0.61 (0.11)	0.25 (0.34)	-0.15 (0.37)
Feature 5	-1	-1 (0)	-1 (0)	-1 (0)	-1 (0)	-0.86 (0.43)	-1 (0)
Time elapsed (seconds)	NA	54.81 (7.02)	180.41 (100.35)	57.87 (1.5)	87.64 (1.26)	3.22 (2.26)	6.38 (0.16)
Number of iterations	NA	486.85 (73.44)	631.45 (352.21)	201.2 (0.52)	502.4 (1.6)	99.25 (70.78)	201.8 (1.85)
N = 1000							
Feature 1	1	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
Feature 2	0.36	0.37 (0.1)	0.35 (0.22)	0 (0)	-0.02 (0.15)	0.41 (0.37)	-0.1 (0.31)
Feature 3	0.36	0.38 (0.15)	0.41 (0.09)	0 (0)	0.18 (0.13)	0.19 (0.37)	0 (0)
Feature 4	0.33	0.32 (0.06)	0.3 (0.08)	0 (0)	-0.58 (0.09)	0.16 (0.43)	-0.1 (0.31)
Feature 5	-1	-1 (0)	-1 (0)	-1 (0)	-1 (0)	-0.96 (0.19)	-1 (0)
Time elapsed (seconds)	NA	65.98 (0.64)	385.58 (227.73)	112.59 (0.68)	92.78 (9.11)	3.68 (4.58)	5.73 (0.58)
Number of iterations	NA	501.1 (6.16)	692.15 (408.74)	201.4 (0.99)	502.95 (2.28)	128.7 (167.17)	201.15 (0.49)

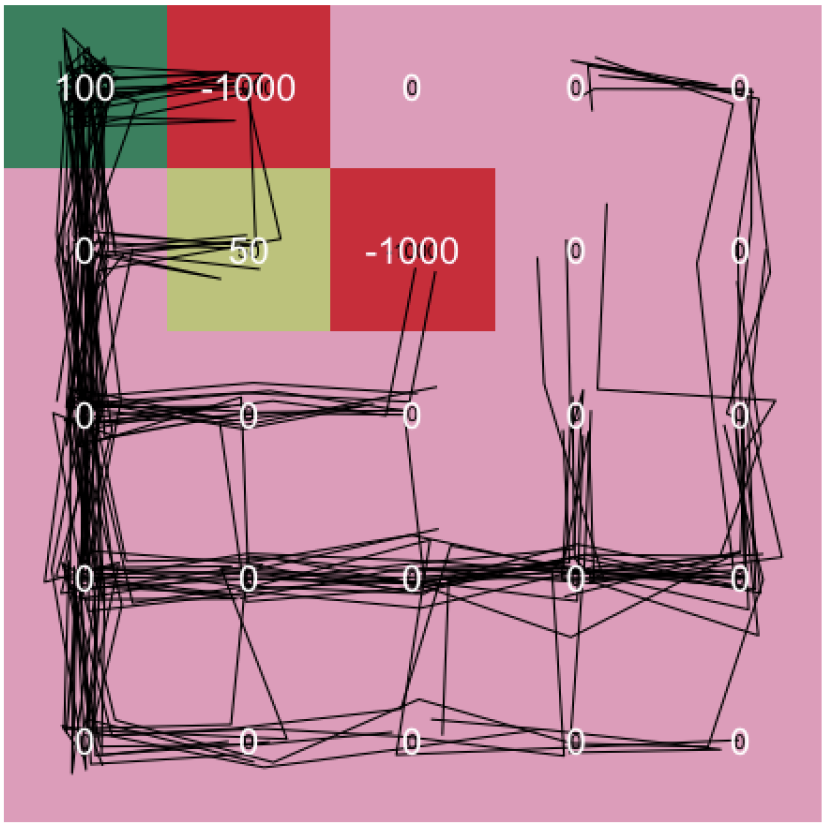
Notes for Table 3: Table 3 reports the performance of three algorithms—Nested Fixed Point, Linear Programming IRL, and Projection IRL—in estimating the payoffs for cells in the 5-parameter GridWorld environment from Figures 3 (a) and (b). The true payoff function is (1, 0.36, .36, .33, -1) by feature. The table shows average estimates and standard deviations in parentheses over twenty trials for each estimation method, data size (100, 500, and 1000 observations), and using trajectories subject to preference shocks versus wind shocks.

Figure 4: Cliff GridWorld with Preference Shocks



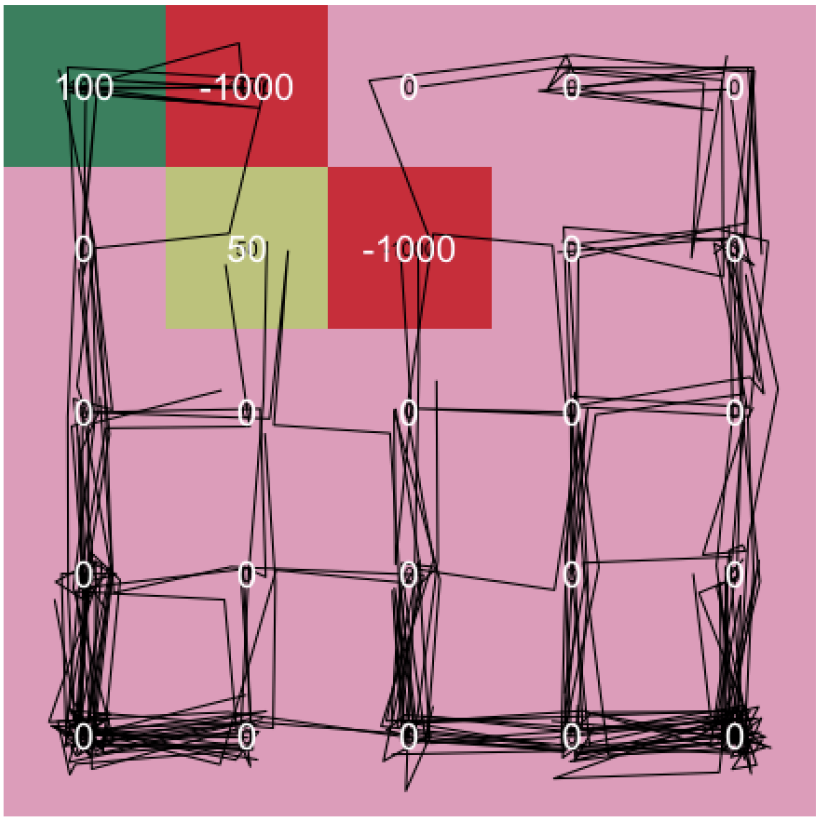
Notes: Figure 4 illustrates typical paths of an agent on a GridWorld "cliff" environment with the payoffs of (100, 50, 0, -1000), subject to N(0, 1) preference shocks each time step and a discount factor of 0.95. The large scale of the payoff function makes trajectories very close to the deterministic case with no shocks at all. Navigating the GridWorld optimally, the agent receives a random starting position and moves toward the cell with a payoff of 100, collecting 50 along the way if it doesn't take an extra time step to do so, and avoiding cells with -1000 unless that was her starting position. Once she has arrived at the cell with a payoff of 100, she stays there.

Figure 5: Cliff GridWorld with Wind Shocks (Coefficient = 0.7)



Notes: Figure 5 illustrates typical paths of an agent on a GridWorld "cliff" environment with the payoffs of (100, 50, 0, -1000), subject to wind shocks with a wind coefficient of 0.7 each time step and a discount factor of 0.95. Navigating the GridWorld optimally, the agent receives a random starting position and seeks to move toward the cell with a payoff of 100. Contrasting with Figure 4 though, she does not collect 50 along the way because she is forward thinking and knows that with a probability of 0.3, she is blown by wind and may find herself collecting the payoff of -1000. Getting to the cell with the payoff of 100 is worth it, but collecting 50 along the way is not.

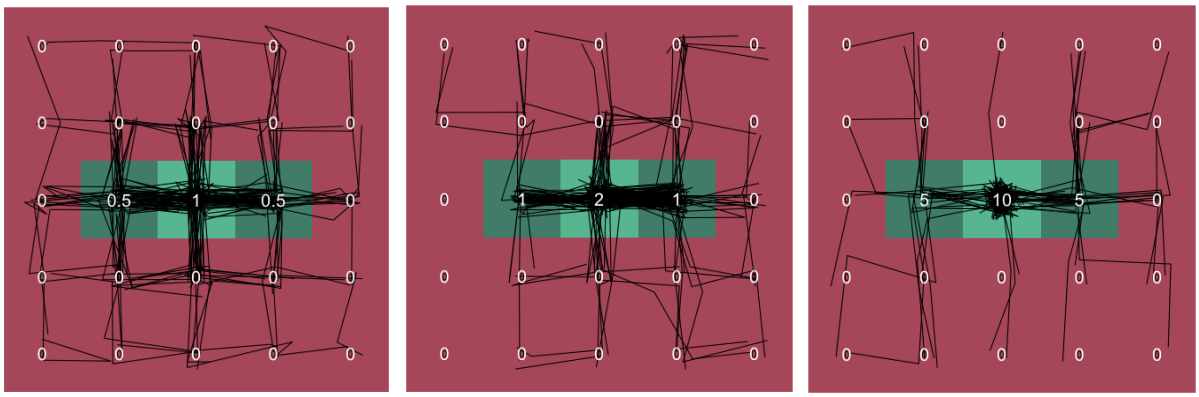
Figure 6: Cliff GridWorld with Wind Shocks (Coefficient = 0.5)



Notes: Figure 6 illustrates typical paths of an agent on a GridWorld "cliff" environment with the payoffs of (100, 50, 0, -1000), subject to wind shocks with a wind coefficient of 0.5 each time step and a discount factor of 0.95. Navigating the GridWorld optimally, the agent receives a random starting position and seeks to maximize the expected sum of discounted payoffs. Here, what is interesting is that her optimal behavior does not include moving toward cells with payoffs of 50 or 100. The wind coefficient of 0.5 means that half the time, the agent is blown by wind and can not expect to be able to select actions according to her optimal policy. Because of the high probability she will repeatedly receive payoffs of -1000 if she travels to the top of the grid, she finds it preferable to stay at the bottom of the grid, collecting zero.

Figure 7: Preference Shocks

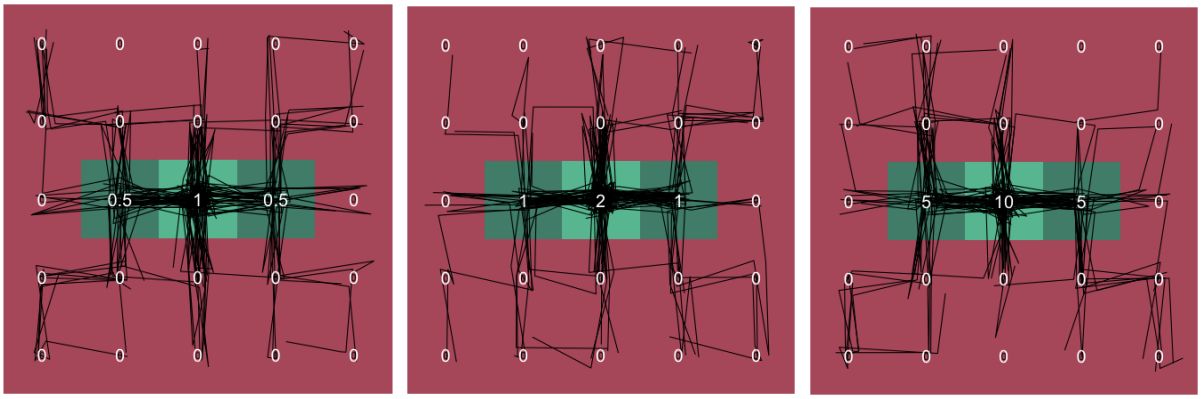
(a) Payoffs of 1, 0.5, 0 **(b)** Payoffs of 2, 1, 0 **(c)** Payoffs of 10, 5, 0



Notes: Figure 7 illustrates typical paths of an expert agent subject to N(0, 1) preference shocks on three GridWorlds with payoff functions that increase in scale: first (1, 0.5, 0), second (2, 1, 0), and then third (10, 5, 0). As the scale of the payoff function increases, the agent's behavior changes to focusing more and more on the center cell with the highest payoff because N(0, 1) shocks become less and less capable of justifying giving up the opportunity to receive the payoff in the center cell.

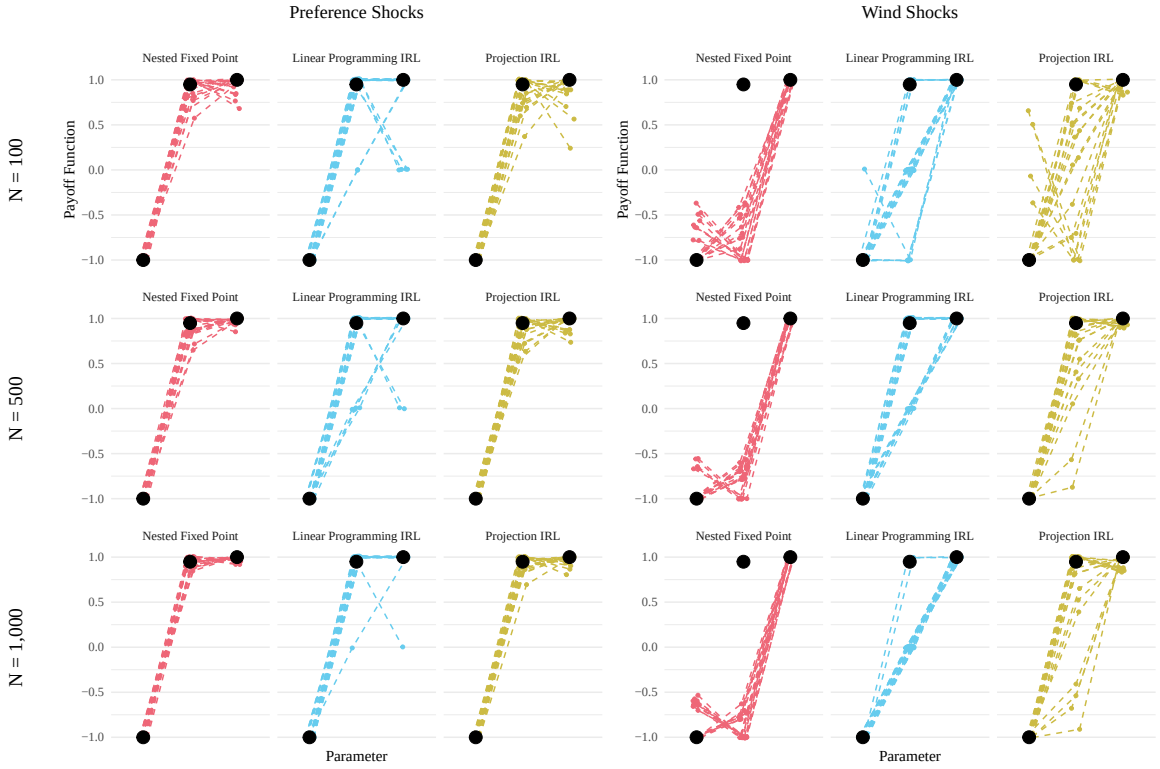
Figure 8: Wind Shocks

(a) Payoffs of 1, 0.5, 0 **(b)** Payoffs of 2, 1, 0 **(c)** Payoffs of 10, 5, 0



Notes: Figure 8 illustrates typical paths of an expert agent subject to wind shocks with a wind coefficient of 0.7 on three GridWorlds with payoff functions that increase in scale. Contrasting with Figure 7, the agent's behavior does not change as the scale of the payoff function increases because the agent's best policies do not change as the scale of the payoff function changes.

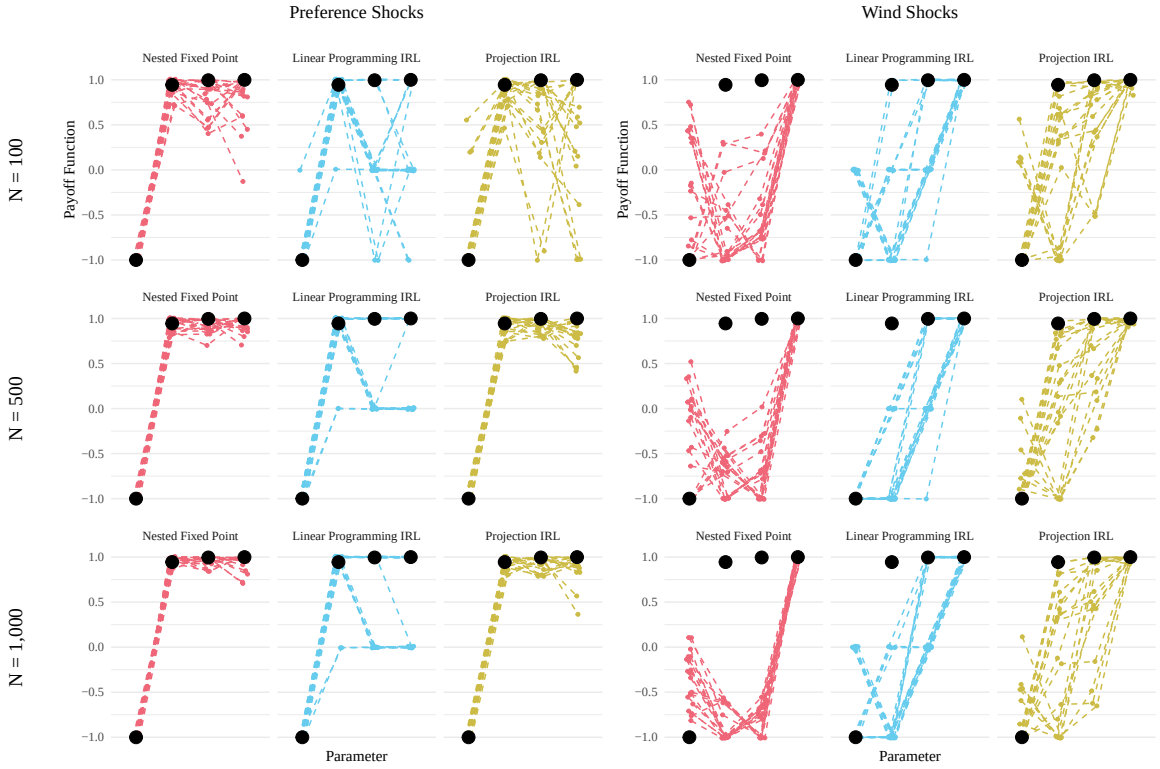
Figure 9: GridWorld Simulation Data and Results with 3 Parameters



Notes: Figure 9 shows the results for all twenty trials of each of the eighteen experiments for the 3-parameter GridWorld environment. Each dashed line represents the estimated payoff function for one trial of the experiment. Black dots represent the true payoff function of (1, .95, -1). For summaries of these results, see Table 1. Under preference shocks, Nested Fixed Point (NFXP) is able to accurately estimate the true payoff function, and its error quickly approaches zero as the data size increases from 100 to 500 to 1000. Under preference shocks, Projection IRL's accuracy is almost as good as NFXP but lags slightly behind. Linear Programming IRL has the limitation that it can only make estimates in the set (1, 0, -1), but under preference shocks, it does not perform badly.

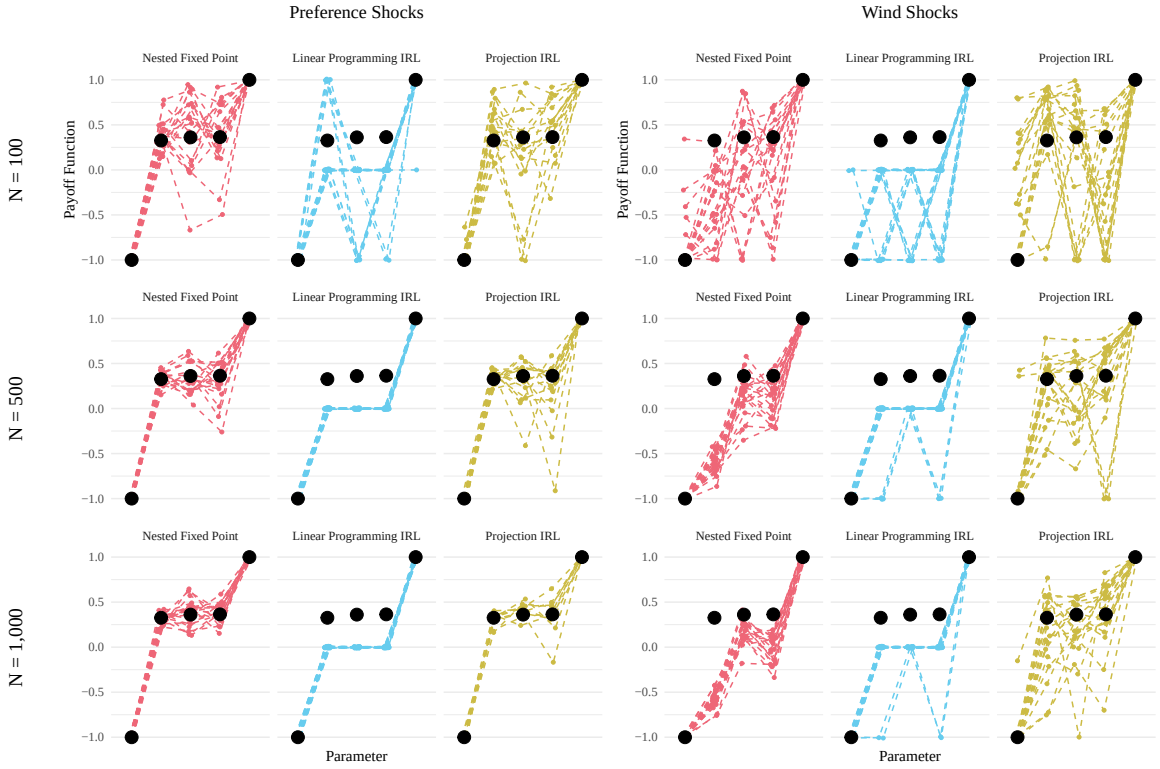
Under wind shocks however, the NFXP estimate converges toward (1, -1, -.7), which is badly biased compared to the true values of (1, .95, -1). Linear Programming IRL settles on estimating (1, 0, -1), and Projection IRL is noisy but seems to prefer (.8, 1, -1), which gets the rank incorrect but the bias is not as severe as the other methods.

Figure 10: GridWorld Simulation Data and Results with 4 Parameters



Notes: Figure 10 shows the results for all twenty trials of each of the eighteen experiments for the 4-parameter GridWorld environment. Each dashed line represents the estimated payoff function for one trial of the experiment. Black dots represent the true payoff function of (1, .99, .94, -1). For summaries of these results, see Table 2.

Figure 11: GridWorld Simulation Data and Results with 5 Parameters



Notes: Figure 11 shows the results for all twenty trials of each of the eighteen experiments for the 5-parameter GridWorld environment. Each dashed line represents the estimated payoff function for one trial of the experiment. Black dots represent the true payoff function of (1, .36, .36, .33, -1). For summaries of these results, see Table 3.

3 COMPARING INVERSE REINFORCEMENT LEARNING TO RUST'S ESTIMATOR

3.1 Introduction

In this paper, I conduct a comparative analysis of the Projection Inverse Reinforcement Learning (IRL) algorithm proposed by Abbeel and Ng (2004) against the Nested Fixed Point (NFXP) algorithm (Rust, 1987) in an empirical, spatial setting. The NFXP algorithm, originating from the Econometrics literature, uses maximum likelihood estimation, whereas the IRL algorithm (from Machine Learning) aims to match feature expectations. Though rooted in separate academic traditions and using different techniques for estimating payoff functions, both techniques are designed to fit models of dynamic discrete choice. In this paper, I evaluate whether these two methods yield comparable estimates in practical problem spaces, and what advantages exist in choosing one over the other. Given the scarcity of existing work comparing these methods, this study aims to shed light on their differences in performance in empirical settings, thereby enhancing our understanding of their respective efficacies.

To empirically evaluate the performance of the NFXP and Projection IRL algorithms, I use publicly available GPS taxi data from 2013 for New York City (Whong, 2014). This dataset includes unique identifiers for taxi drivers, which allows me to link together all the trips taken by a driver in a single day and trace their route through the city. The strategic decision a taxi driver makes is to decide which direction to start driving after a dropoff to pick up a new passenger. I model this decision as a dynamic discrete choice because the driver aims to quickly find a new passenger to maximize immediate payoffs, while also considering how to put themselves in advantageous positions to get high-payoff trips later on. Estimating such a model provides insights into the relative payoffs of different neighborhoods and their characteristics from a taxi driver's perspective.

To accurately model the transition dynamics for a taxi driver in New York City, I first estimate the transition probability function (G) from the GPS taxi data. This function predicts the likely destinations of passengers based on their pickup locations and the time of day, effectively capturing the flow of taxi passenger traffic within the city. Once G is established, I apply both the NFXP and Projection IRL algorithms to estimate the payoff functions for taxi drivers. These payoff functions reflect the relative attractiveness of different neighborhoods and characteristics from the perspective of a taxi driver.

My findings reveal two important insights regarding the capabilities of Projection Inverse Reinforcement Learning (IRL) and the Nested Fixed Point (NFXP) estimator. First, the choice of the kernel density smoothing parameter h for the transition

probability function greatly influences the estimates derived from both methods. IRL offers direct feedback about the validity of the choice for h. For instance, an excessively large value for h generates overly smooth transition probabilities, leading to a failure in IRL's ability to match feature expectations. When h is sufficiently reduced, IRL becomes capable of matching feature expectations with precision.

Second, IRL exhibits a relatively flat objective function, meaning that various payoff functions can closely align feature expectations with empirical data, resulting in divergent estimates based on the initial search value. In contrast, NFXP provides more dependable estimates regardless of the starting point. This dependability underscores the robustness of NFXP in estimating dynamic discrete choice models.

3.1.1 Related Work

My research focuses on extending the work in Inverse Reinforcement Learning (IRL) by connecting it with Dynamic Discrete Choice (DDC) methods. Only one other study has attempted this connection. Sanghvi et al. (2021) established that IRL and DDC share the same objective: estimating the structural parameters of a Markov Decision Process (MDP) using data from an agent's optimal trajectories. They demonstrated that Maximum Causal Entropy IRL (Ziebart et al., 2010) is equivalent to the Nested Fixed Point (NFXP) algorithm (Rust, 1987) when payoffs are linear in features. Their research compared this IRL method with two DDC methods (Hotz and Miller 1993 and **?**) and found that all methods performed similarly with ample data, but NFXP had an edge when data was limited. My research is unique because I compare the NFXP algorithm with an earlier IRL method developed by Abbeel and Ng, rather than Maximum Causal Entropy IRL. Moreover, I compare these two methods in the context of a real-world application, which has not been done before.

Economists have primarily used public taxi driver data to answer questions about labor supply like, "when drivers are earning more, will they work more hours as Neoclassical labor supply theory predicts, or will they work fewer hours as theories on income targeting or reference points predict?". For example, Camerer et al. (1996) found evidence, especially among inexperienced taxi drivers, in support of income targeting: the logarithm of daily hours is negatively related to earnings divided by daily hours. Farber (2005) finds evidence for the Neoclassical theory instead, and the discrepency is accounted for by "division bias": regressing hours worked on a wage measure that is computed using the reciprocal of hours worked creates a downward bias. Other studies in the Economics literature using public taxi data include Haggag et al. (2017) : studying how taxi drivers learn to optimize by accumulating neighborhood-specific experience. And Jackson (2018) use taxi data and an instrumental variables approach to study supply-side discrimination, finding that

especially among inexperienced drivers, taxi drivers seem to use statistical discrimination and linger shorter amounts of time while looking for new passengers in neighborhoods with certain demographics.

3.2 Model

In this section, I define the model used to analyze the decision-making process of taxi drivers regarding their next destination after a passenger drop-off. The model considers the drivers as agents who aim to maximize a value function by choosing the best direction to head for finding their next passenger. The choices are influenced by the transition probabilities of passenger requests between census tracts, which vary by time of day, day of week, and week of year.

The agents in my model are taxi drivers. Each driver is observed over time, making sequential choices about where to head after dropping off a passenger. The state, $s \in S$, is defined as the census tract corresponding to the GPS location (latitude and longitude coordinates) of the driver's drop-off. This state captures the geographical context within which the driver operates. The driver chooses among $K = 8$ actions, representing primary and secondary compass directions to drive while searching for their next passenger.

The marginal costs and benefits for taxi drivers can be categorized as follows: benefits come from direct income, including fares and tips. Marginal costs include fuel costs and maintenance costs, which may or may not be incurred directly by the taxi driver depending on whether the driver owns the vehicle, leases it, or works as a fleet driver. Additionally, opportunity costs play a significant role: longer trips may prevent drivers from completing multiple shorter, potentially more profitable trips. Furthermore, long trips to less populated areas comes with a risk that the driver will have to make a return trip without a passenger (otherwise known as *deadheading*: incurring unproductive time and mileage).

I define the payoff function $\pi(s)$ as the expected utility for a taxi driver in census tract s when starting their search for their next passenger. If the driver expects to find a new passenger quickly, $\pi(s)$ is higher, reflecting the fact that the driver finds the state to be favorable. If the next passenger is expected to pay a high fare and give a large tip, $\pi(s)$ is also higher. But if the next passenger is expected to request a long trip that consumes a lot of time and fuel without a large enough driver benefit to compensate, $\pi(s)$ will be lower.

In state s at time t, the driver receives K independent and identically distributed preference shocks, denoted by ε_k, which are not observable to the Econometrician. The preference shocks reflect any idiosyncratic fluctuations in the taxi driver's perceived costs or benefits for each action (driving in each of the primary and secondary compass

directions while searching for their next passenger). For instance, if a driver immediately spots a new passenger hailing them from the North, "North" would have a positive preference shock at that time. Conversely, if the driver knows about a traffic jam to the South, "South" would have a negative preference shock due to the time cost associated with driving in that direction.

I assume additive separability between expected utility and preference shocks, so the total payoff for taking action a in state s at time t is the expected utility plus the preference shock:

$$\pi(s_t) + \sum_{k=1}^{K} \varepsilon_{t,k} \cdot \mathbf{1}(a_t = k) \tag{9}$$

The action a chosen by the driver has another associated benefit or cost: the expected value of the next state s' the driver will end up in after the next drop-off. The driver may expect their next destinations s' to vary significantly depending on which direction they choose to drive in state s. The transition probability function G governs the probability that a driver in a given census tract s, having just made a dropoff, choosing action a, will pick up a passenger and drop them off in another census tract s'. G is therefore made up of elements $g(a, s, s')$. For this study, I assume the transition probabilities are identical for the morning commute (7:45am-9:45am, Monday to Friday) during the Spring months (March, April, and May) of 2013. Using this assumption, I can aggregate 2 million rides' worth of data to estimate G.

The driver's value is the discounted sum of the pay-offs for each period, with the discount factor β less than 1. By assuming a Markovian transition process, I can abstract away from calendar time. The value of being in a given census tract s and receiving the preference shocks ε is defined as follows:

$$V(s, \varepsilon) = \max_a \left[\pi(s) + \sum_{k=1}^{K} \varepsilon_k \cdot \mathbf{1}(a = k) + \beta \mathbb{E}[V(s', \varepsilon')|s, a] \right] \tag{10}$$

$$= \max_a \left[\pi(s) + \sum_{k=1}^{K} \varepsilon_k \cdot \mathbf{1}(a = k) + \beta \sum_{s'=1}^{S} g(a, s, s') V(s', \varepsilon) \right] \tag{11}$$

3.3 Data

I use three datasets in this analysis, which are all publicly available: New York City taxi cab GPS data from 2013 (Whong, 2014), U.S. Census data on Means of Transportation to Work, Table B08301 (U.S. Census Bureau, 2013), and geographic shapefiles for 2010 New York City census tracts (New York City Department of City Planning, 2010).

The taxi cab GPS data (Whong, 2014) was obtained through a FOIL (Freedom of Information Law) request submitted by Whong to the Taxi and Limousine Commission

in NYC for data covering January to December of 2013. This dataset includes detailed ride information such as medallion (taxi cab code), hack license (driver code), date and time of passenger pickup and drop-off, trip distance, and latitude and longitude coordinates of the passenger pickup and drop-off.

Additionally, I use publicly available survey data on Means of Transportation to Work (U.S. Census Bureau, 2013). This dataset provides information on the total number of workers living in each census tract in 2013, the number of workers who reported that they primarily commuted to work using a personal vehicle, and the number of workers who reported that they primarily commuted using public transportation.

Figure **??** illustrates a strong correlation between the percentage of workers commuting by personal vehicle and borough. In Manhattan, most census tracts report only 3% to 8% of workers primarily using personal vehicles. For comparison, across New York City, the percentage of workers commuting by personal vehicle typically ranges from 10% to 36%. Figure **??** presents the distribution of public transit commuters across boroughs. Staten Island has the fewest proportion of public transit commuters, and the Bronx has the highest.

3.4 Methods

3.4.1 Estimating the Transition Probability Function

First, I estimate the transition probability function (G) in order to accurately model the transition dynamics for a taxi driver in New York City. Taxi drivers make one type of strategic decision: which direction to drive after dropping off a passenger. This involves the driver evaluating their current location (state s_0) and choosing a direction (action a) that not only aims to maximize immediate earnings but also positions them advantageously for future earnings. The driver's decision of a in s_0 leads to a new passenger who requests a new destination, and finally a new location (state s_1) for the driver.

The transition probability function G contains estimates of the probabilities that a taxi driver in s_0 who chooses action a will end up in any given state s_1 in time to make their next strategic decision about which direction to head. So an interpretation of G would be that it predicts popular destinations of passengers in each neighborhood. An inaccurate estimation of G would lead to an inability to understand the driver's perceived opportunities, leading to possible overestimations or underestimations of potential earnings in different neighborhoods.

To estimate the transition probability function, I first filter the data to include only taxi trips which occurred during the peak morning rush hours from 7:45 AM to 9:45 AM, Monday through Friday, in the months of March, April, and May of 2013.

Filtering on the morning commute event in a certain season should help patterns emerge more clearly. For instance, during the morning commute, there is a strong directional movement of taxi cab passengers from suburban to downtown areas (see table 5). This suggests that taxi drivers may adopt specialized strategies for navigating these peak hours. Analyzing these strategies can reveal the drivers' payoffs for being in certain neighborhoods during these hours, which may differ significantly from their payoff function at other times of day. I limited my analysis to the Spring of 2013 because in the Summer of 2013 in New York City, green cabs made their debut. Green cabs have cheaper medallions, they accept both e-hailing and street-hailing, but they are prohibited from operating in a large section of Manhattan. The presence of green cabs certainly would have shifted the incentives for yellow cabs, so I instead focus my analysis on the Spring, before any green cabs were driving in New York City.

Next, I define the spatial boundaries based on 2010 census tract spatial data. Census tracts are intended to represent neighborhoods that are relatively homogeneous in terms of population characteristics and economic status, typically containing about 4,000 people each. New York City is divided into 2,168 such tracts: 111 in Staten Island, 288 in Manhattan, 761 in Brooklyn, 389 in the Bronx, and 669 in Queens. I join the taxi trip data to the spatial data based on longitude and latitude coordinates to determine which census tract each taxi trip began and ended in. Recognizing the possibility of trips beginning or ending outside the city limits, I introduce a 2169th category labeled "outside NYC census tracts" to accommodate these exceptions. This category accounts for less than 1.1% of the morning commute dropoffs. To build the transition probability function, I calculate the centroid latitude and longitude for each census tract, and for the "outside NYC" region, I use a set of coordinates directly to the North West of the city (noticing most trips by far outside of New York were headed toward New Jersey).

To infer driver actions from the taxi ride dataset, I analyze the data of each driver (identified by their hack license) on each date. I select the coordinates of each of their dropoffs along with their next pickups and draw the vector between those two sets of coordinates. I calculate the angle of that vector (after translating latitude and longitude differences to miles), and then round that angle to the nearest compass direction among the eight primary and secondary directions (N, S, E, W, NE, NW, SE, SW), each representing a 45-degree segment out of a full 360-degrees. This process allows me to determine the approximate directional action the driver took after each passenger dropoff. Then to estimate the transition probability function, I count how often drivers, having dropped off a passenger in a specific census tract (s_0) and then selecting a specific action (a), wind up in each census tract after their subsequent dropoff. This information summarizes the directional flows of taxi passengers during the morning commute.

Finally, I use kernel density estimation to smooth the transition probability function. The dataset for my study consists of 2 million taxi rides from the morning commute in the Spring, which is still insufficient to achieve a smooth estimate of the transition probability function G. A smoothed array G is crucial for the convergence of the value iteration algorithm, which repeatedly applies G to find the value function V from a payoff function π. The kernel density estimation method involves taking each pair of initial census tracts s_0 and actions a, and estimating a smoothed probability distribution of the driver's next dropoff location using the latitude and longitude of that dropoff. Then, to complete the array G, I perform surface interpolation based on the centroids of the census tracts to calculate the probabilities of arriving at each subsequent census tract s_1.

For the kernel density estimation process, the goal is to use the smallest possible smoothing factor to preserve as much of the rich detail of the data as possible, while still allowing the value iteration algorithm to successfully converge. I adjusted the smoothing factor incrementally, settling on a value of 0.01, which was the lowest smoothing factor I found that allowed the value iteration algorithm to converge quickly.

3.4.2 Nested Fixed Point Algorithm

I use a standard Nested Fixed Point algorithm implementation (Rust, 1987), where for each iteration of the maximum likelihood search, I use the guess for the payoff function π to find the value function V using value iteration. Then I calculate choice probabilities $P(a|s_0)$ assuming preference shocks ε are distributed Extreme Value type 1 and use them to form the likelihood function: the number of times each decision a was taken in each state s_0, multiplied by the estimated probability such an action should be taken according to $P(a|s_0)$, logged and then summed. I use simulated annealing for my maximum likelihood estimation (**?**).

3.4.3 Projection IRL

I broadly follow the Abbeel and Ng (2004) Projection IRL algorithm, with a few changes: first, using a representative subsample of the expert data, second, generating data using preference shocks, and third, using an affine transformation of the payoff guess for each iteration with a scaling parameter. In the next few sections, I will discuss each of these modifications in turn, along with why they are necessary.

Each iteration of the Projection IRL algorithm has a guess for the payoff function π, it uses value iteration to find the value function V, it simulates synthetic trajectories using V with the same starting positions for trajectories as the sample (the "expert"), aiming to match the size and shape as the sample, and calculates the synthetic feature expectations for that guess. If the synthetic feature expectations are within a tolerance

of 0.025 compared to the expert feature expectations, then the algorithm returns the payoff function which generated those synthetic feature expectations. If the synthetic feature expectations are not, new guesses for the payoff function are determined using the method of projection, where I take the line between the synthetic feature expectations for the last two guesses and I project the margin between the last guess and the expert's feature expectations down onto it. The result is geometrically guaranteed to be at least as close to the expert's feature expectations as either of the last two guesses, and the difference between the expert's feature expectations and that projection becomes the next guess for the payoff function.

1. Subsampling

   For each iteration of the IRL algorithm, I need to be able to generate synthetic data of the same size and shape as the expert dataset under different guesses for the payoff function. The expert sample has two million observations, and I found that it was computationally burdensome to generate so many synthetic trajectories in each iteration of the algorithm. So instead, just like estimating models using OLS on a subsample of observations, at the beginning of the algorithm I create a representative subsample of expert trajectories by randomly sampling trajectories, accumulating a larger and larger sample until the feature expectations match the expert's, or are within a certain small tolerance (I use 0.01). I find that using this procedure, the representative subsample usually has between 5 and 10 thousand observations.

   For a certain feature (model parameter) and a certain dataset, the feature expectation is a time-discounted measure of how often a feature is visited on average per trajectory. The expectation for feature i over a dataset with the number of observations j, containing k trajectories, where t is the time step for each trajectory starting at 1, is defined as:

$$\text{feature expectation}_i \equiv \frac{1}{k} \sum_{j=1}^{\max(j)} \text{feature}[i,t] \times \beta^{t-1}$$

   Where $feature$ is a vector of length j with each observation as the density of feature i visited in that row of the data. For example, if the feature is binary and takes 0 or 1 (as in the Staten Island feature), $feature_i$ will take 1 if the agent visited Staten Island that period, and 0 if not. If the feature is not binary (like log Total Workers), then $feature_i$ is the log of total workers in the census tract visited.

   I apply an affine transformation between 5 and -5 to the payoff function guess for each iteration of the Projection IRL algorithm to avoid the issue where, as the

margin between the expert feature expectations and synthetic feature expectations gets small, the guess for the payoff function doesn't also get small in magnitude. The choice of 5 for the scaling factor does not seem to have a significant impact on the final estimate, except in magnitude.

3.5 Results

In table 6, I present the main estimates for the taxi driver's payoff function, as estimated by the NFXP algorithm and then as estimated by the Projection IRL algorithm. For the NFXP estimates for the first model (including only parameters for the borough of the census tract and whether the census tract contains an airport), I estimate that Manhattan and Queens yield a very high payoff for taxi drivers serving the morning commute. Staten Island and Brooklyn yield the lowest payoffs, and the Bronx and New Jersey are in the middle. In this first model, airports do not seem to be lucrative for taxi drivers to serve during the morning commute.

The second model adds a parameter for the natural log of the total number of workers above age 16 living in the census tract (see table 6). When controlling for total workers, Manhattan loses its positive payoff estimate, but Staten Island and Brooklyn maintain low payoffs. NFXP estimates a very small negative impact for total workers on driver payoff. After controlling for total workers, estimates for the payoffs to each borough mainly keep the same sign compared to the first model, reflecting the fact that not much variation exists in log total workers per census tract by borough (see 12).

The third model (table 6) adds two more parameters: I control for the percentage out of the total number of workers who commute to work using personal vehicles in each census tract, as well as the percentage of total workers who commute to work using public transit. Using NFXP, I find that the taxi driver's payoffs decrease both with more car commuters and more public transit commuters, and accounting for those variables, total workers have a positive effect on driver payoff. In this model, Staten Island no longer has a negative payoff, which is reasonable because Staten Island has the most car commuters out of any borough by far (see figure **??**). I find that a 10 percentage point increase in personal vehicle commuters lowers the taxi driver's payoff by 0.03 units, and a 10 percentage point increase in the percentage of public transit commuters decreases the taxi driver's payoff by 0.16 units. Controlling for the number of workers and the ways that they commute turns the coefficient on Airports from negative to positive, which is not surprising because airport census tracts have no one living in them.

The IRL estimates are all subject to an affine transformation between -5 and 5 in every iteration of the algorithm, so the scale of the payoffs differ slightly when comparing NFXP to IRL. In all three models (see table 6), IRL estimates the highest payoff for the driver in New Jersey, and the lowest payoffs to Brooklyn, Staten Island,

and Manhattan. Just like NFXP, IRL estimates negative payoffs to airports until I control for total workers and the ways they get to work, at which point the payoff to airports becomes positive. IRL estimates slightly negative payoffs in most runs of the algorithm to more total workers, car commuters, and public transit commuters. Overall, the standard deviations for the IRL estimates are large: each run of the Projection IRL algorithm yields different estimates for the payoff function, and those estimates often vary significantly.

3.6 Discussion

My findings reveal two important insights regarding the capabilities of Projection IRL and Nested Fixed Point. The first is that the selection of the kernel density smoothing parameter for the transition probability function G has a significant impact on the estimates derived from both NFXP and IRL. In particular, if the smoothing parameter is too large, IRL will not be able to match the feature expectations from the data.

Consider figure 14, which depicts 25 simulated sets of feature expectations under a transition probability function with the smoothing parameter of 0.10 and the corresponding Projection IRL payoff function estimate. The observed pattern shows that features frequently visited in the true data (such as those in Manhattan) are not visited nearly as often in the simulated data. Conversely, features that are infrequently visited in the true data (such as those in the Bronx, Brooklyn, and Queens) are visited too often in the simulated data. This pattern persists across various Projection IRL estimates and is predictable if the smoothing parameter of 0.10 is excessively large, resulting in an overly smooth transition probability function. When the transition probability function is too smooth, no matter how high the payoff to Manhattan is or how low the payoffs to the Bronx, Brooklyn, or Queens are, simulated data will end up spending equal amounts of time in every area of the map.

When I reduce the smoothing parameter down to 0.01 (see figure 13), the issue disappears. This choice for the smoothing parameter fits the data much better because, using the same payoff function as in figure 14, the expert's feature expectations can now be matched with precision. Evaluating how well IRL can match expert feature expectations appears to offer valuable feedback on the appropriate selection of the smoothing parameter for the transition probability function.

The second takeaway is that IRL's objective function is comparatively flat, so numerous payoff functions can closely align feature expectations with the empirical data. Consequently, the starting point of the search in IRL can lead to markedly divergent estimates. In contrast, NFXP, when estimating the same model, demonstrates consistent estimation as different initial positions produce similar estimates.

3.7 Conclusions

In this study, I estimated a model of the dynamic decisions made by New York City taxi cab drivers, especially their payoffs during the morning commute. These payoffs were analyzed as a function of the borough, the presence of airports, the working population of the neighborhood, and the modes of transportation to work reported by those workers. I compared two methods for estimating this dynamic discrete choice model: the Nested Fixed Point (NFXP) algorithm (Rust, 1987) and the Projection Inverse Reinforcement Learning (IRL) algorithm (Abbeel and Ng, 2004).

My findings indicate that the NFXP likelihood function provides stable estimates regardless of the initial starting position for the search. In contrast, the IRL objective function exhibited a flatter profile, resulting in significantly different estimates depending on the initial search value. The estimates derived from IRL and NFXP frequently did not align. However, the concept of feature expectations from IRL proved valuable for selecting the smoothing parameter for the kernel density estimation of the transition probability function.

In my analysis, I show that different boroughs have varying impacts on the taxi drivers' payoffs. Using the NFXP algorithm, I estimated that, during the morning commute, taxi drivers prefer neighborhoods with a higher number of workers. And when controlling for the number of workers, taxi drivers tend to avoid neighborhoods where a larger proportion of workers commute using public transit or personal vehicles.

3.8 Tables and Figures

Table 4: Pickups and Dropoffs by Borough

Borough	Pickups	Dropoffs
Staten Island	0.005%	0.011%
Manhattan	91.4%	91.9%
Brooklyn	1.51%	1.33%
Bronx	0.066%	0.298%
Queens	4.28%	3.46%
Outside NYC (New Jersey)	2.77%	2.99%

Notes: This table presents the percentage of pickups and dropoffs that occurred in each borough during the morning commute in Spring (March, April, and May) of 2013.

Table 5: Taxi Pickup and Dropoff Boroughs as Transition Probabilites

	Staten Island	**Manhattan**	**Brooklyn**	**Bronx**	**Queens**	**Outside NYC**
Staten Island	0.901	0.064	0.023	0	0	0.012
Manhattan	<0.001	0.964	0.006	0.002	0.025	0.003
Brooklyn	<0.001	0.681	0.274	0.001	0.041	0.003
Bronx	<0.001	0.386	0.022	0.536	0.031	0.024
Queens	<0.001	0.625	0.081	0.014	0.265	0.014
Outside NYC	<0.001	0.025	<0.001	<0.001	0.003	0.971

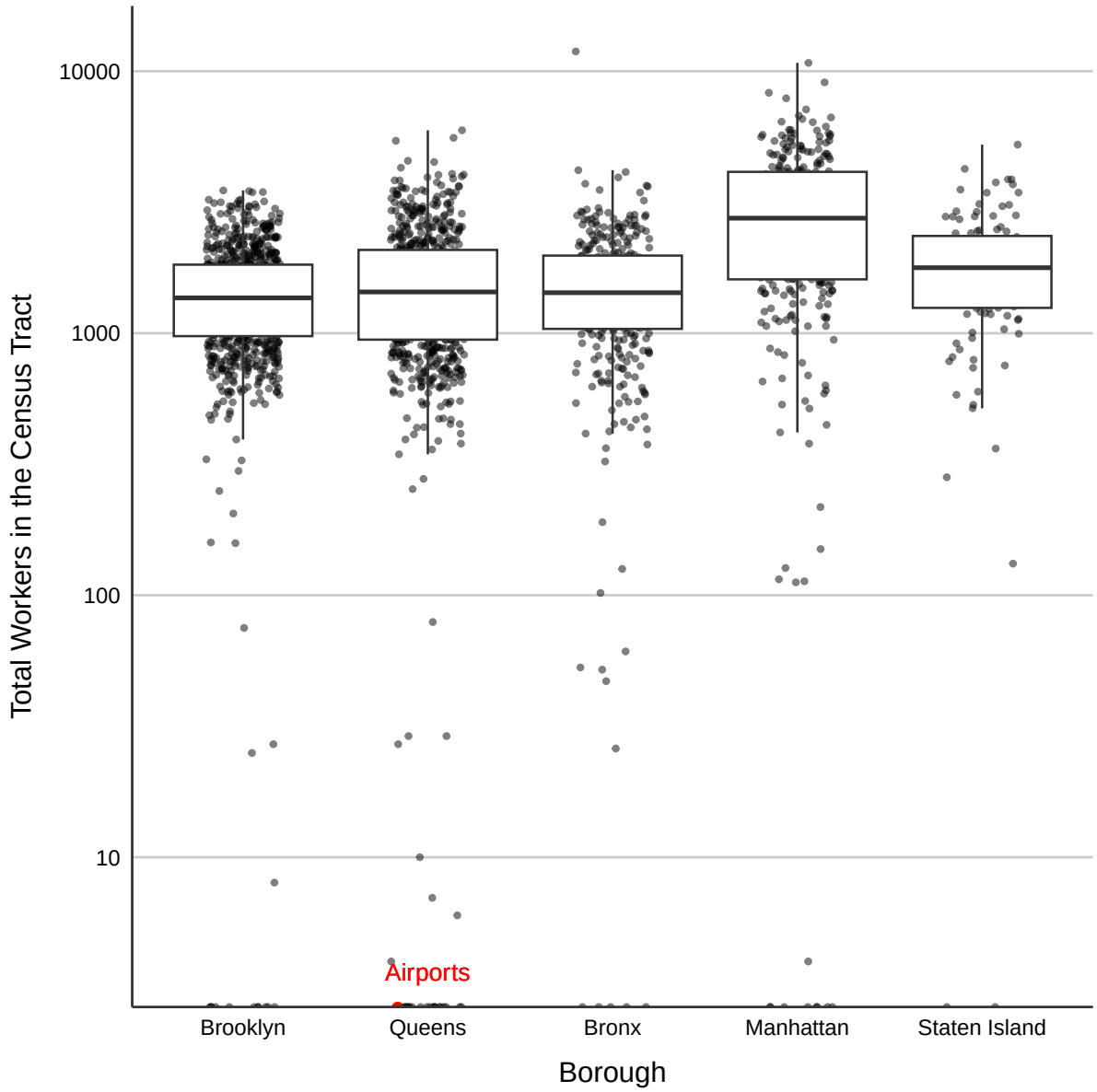
Notes: This table is a simplification of the probability transition matrix. It presents (for the morning commute in Spring, 2013), given a taxi pickup is in a certain borough (row), what the estimated probabilities that the dropoff would be in each other borough (columns), given the percentage of times such an event occurred in the taxi dataset.

Table 6: Morning Commute Payoff Function: Weekdays, 7:45am-9:45am

	NFXP	NFXP	NFXP	IRL	IRL	IRL
Staten Island	-3.760 (0.008)	-0.890 (0.006)	-0.121 (0.004)	-0.208 (2.724)	0.917 (2.757)	0.105 (1.484)
Manhattan	2.363 (0.020)	1.128 (0.007)	0.716 (0.004)	0.484 (4.685)	1.809 (3.280)	0.403 (2.537)
Brooklyn	-3.604 (0.019)	-2.970 (0.004)	-3.230 (0.005)	-1.807 (2.799)	1.096 (3.240)	0.241 (1.728)
Bronx	0.731 (0.031)	0.298 (0.003)	-0.169 (0.004)	-1.091 (3.547)	1.017 (3.031)	-0.287 (1.574)
Queens	2.143 (0.018)	2.023 (0.006)	1.664 (0.004)	0.881 (3.178)	1.074 (2.185)	0.422 (1.848)
Outside NYC (New Jersey)	2.441 (0.008)	1.698 (0.004)	-1.266 (0.007)	3.318 (3.469)	1.887 (1.866)	1.749 (0.782)
Taxi Riders (log)	0.103 (0.007)	0.011 (0.007)	0.211 (0.008)	-0.163 (3.600)	0.478 (3.120)	1.020 (3.270)
Airport	-0.140 (0.004)	0.410 (0.005)	-0.131 (0.007)	0.656 (2.661)	0.770 (2.340)	0.750 (1.113)
Rail Riders (pct)		-0.652 (0.005)	-0.254 (0.006)		2.096 (3.319)	4.033 (3.821)
Total Workers (log)		0.051 (0.001)	0.038 (0.001)		0.103 (3.580)	-3.766 (2.664)
Bicyclists (pct)			1.315 (0.004)			0.725 (1.199)
Walkers (pct)			-1.348 (0.006)			-3.245 (4.175)
Work from Home (pct)			1.356 (0.005)			0.699 (2.844)
Log Likelihood	-3776278	-3769089	-3769375	-3784992	-3776278	-3776278
Feature Expectation Margin				0.22	0.61	1.10

Note: I use Nested Fixed Point and Projection IRL to estimate three models for the payoff function for taxi cab drivers during the morning commute in NYC in March, April, and May of 2013. The parameters for the first model contain only indicators for which borough the census tract is in, plus an indicator for whether the census tract contains an airport (two census tracts in Queens have airports: John F. Kennedy International Airport and LaGuardia Airport). The second model adds a parameter for the natural log of the total number of workers above age 16 living in the census tract, measured in 2013 by the U.S. Department of Transportation. The third model adds two more parameters: one for the percent out of the total number of workers who use personal vehicles to travel to work, and also the percent of the total number of workers who use public transit to travel to work (both also measured in 2013). NFXP estimates are reported (using simulated annealing to solve the optimization problem), with standard errors in parentheses. The Projection IRL algorithm does not have a concept of standard errors, so I report the average Projection IRL estimate out of ten runs of the algorithm with different starts for the search, with the standard deviation of those ten runs in parentheses. For each estimation, I report log likelihoods (the value NFXP seeks to maximize) as well as the average feature expectation margin (the value IRL seeks to minimize).

Figure 12: Total Workers Per Census Tract



Notes: Using 2013 Means of Transportation to Work data from the U.S. Department of Transportation, I visualize the distribution of total workers above the age of 16 living in each census tract for each New York City borough. Every borough has census tracts with zero workers, which include the two census tracts with an airport, both in Queens (John F. Kennedy International Airport and LaGuardia Airport).

Table 7: Average Means of Transportation to Work by Borough (2013)

Borough	Total Workers	Taxi Riders	Rail Riders	Bicyclists	Walkers	Working from Home
Bronx	1546.41 (955.98)	13.25 (24.79)	602.17 (398.82)	4.93 (15.58)	123.41 (116.45)	48.77 (46.62)
Brooklyn	1433.13 (654.65)	4.96 (11.02)	707.98 (462.67)	19.26 (34.56)	125.03 (126.07)	58.96 (66.58)
Manhattan	2910.48 (1797.27)	86.40 (108.49)	1431.37 (988.03)	38.77 (57.64)	608.98 (536.16)	194.31 (157.32)
Queens	1560.61 (916.37)	6.64 (16.36)	583.02 (555.08)	7.78 (19.00)	89.56 (99.25)	38.42 (38.21)
Staten Island	1847.08 (942.90)	2.27 (7.79)	57.45 (64.53)	1.67 (4.50)	49.25 (51.72)	41.43 (37.21)

Notes: Using 2013 Means of Transportation to Work data from the U.S. Department of Transportation, I find, for each census tract, the total number of workers above the age of 16 living there, the number who report primarily using a taxi to get to work, the number who report primarily using a long-distance train or commuter rail to get to work, the number who report primarily biking and walking to work, and the number who report primarily working from home. The summary statistics in this table are the means (and standard deviations in parentheses) of those census tract statistics for each borough.

Table 8: Correlation Matrix: Means of Transportation to Work by Census Tract

	Total Workers	Taxi Riders	Rail Riders	Bicyclists	Walkers	Working from Home
Total Workers	1.000					
Taxi Riders	0.513	1.000				
Rail Riders	0.815	0.421	1.000			
Bicyclists	0.387	0.183	0.446	1.000		
Walkers	0.673	0.601	0.541	0.384	1.000	
Working from Home	0.673	0.570	0.645	0.397	0.603	1.000

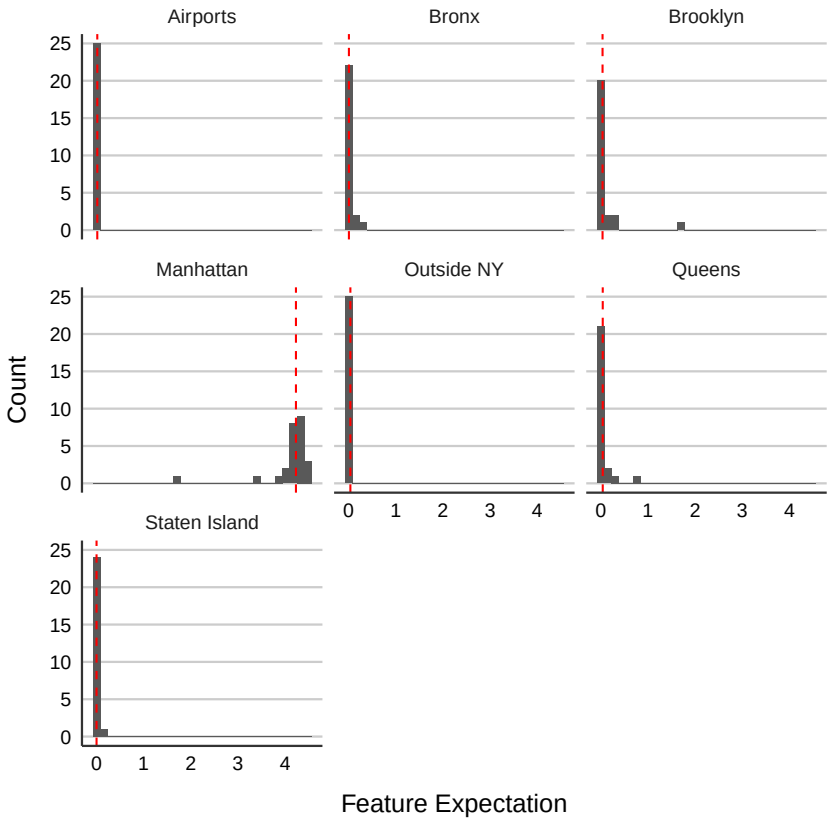
Notes: This table presents the correlation coefficient between each means of transportation to work variable by census tract. The dataset comes from the 2013 Means of Transportation to Work survey from the U.S. Department of Transportation.

Table 9: Correlation Matrix: Means of Transportation to Work by Census Tract (%)

	Total Workers	Taxi Riders	Rail Riders (%)	Bicyclists (%)	Walkers (%)	Working from Home (%)
Total Workers	1.000					
Taxi Riders	0.513	1.000				
Rail Riders (%)	0.245	0.077	1.000			
Bicyclists (%)	0.108	0.046	0.254	1.000		
Walkers (%)	0.225	0.333	0.052	0.165	1.000	
Working from Home (%)	0.130	0.214	0.100	0.206	0.255	1.000

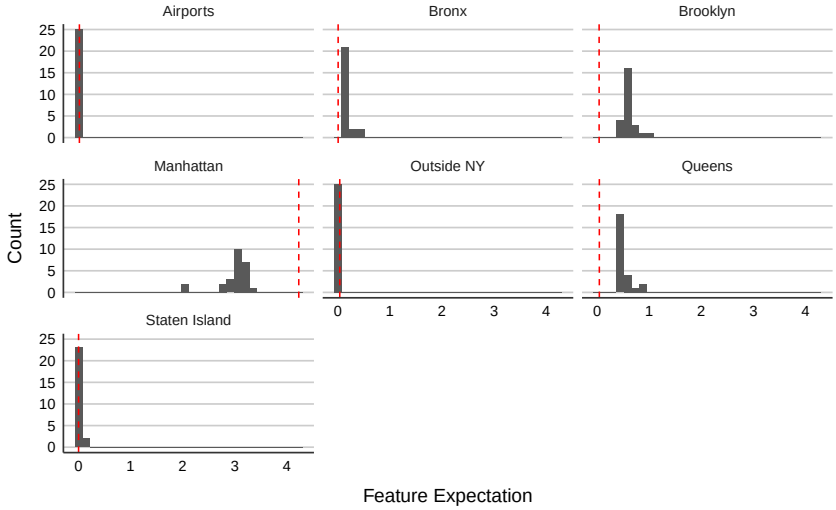
Notes: This table presents the correlation coefficient between each means of transportation to work variable by census tract, where total workers and taxi riders are presented as counts and rail riders, bicyclists, walkers, and those working from home are presented as percentages of the total number of workers in the census tract. The dataset comes from the 2013 Means of Transportation to Work survey from the U.S. Department of Transportation.

Figure 13: Feature Expectations under the Smoothing Parameter 0.01



Notes: This figure shows the distribution of feature expectations (a time-discounted sum of the number of times each feature was visited on average per trajectory) for synthetic datasets generated under a certain transition probability function and a certain payoff function. For this plot, the transition probability function was smoothed using kernel density estimation with a smoothing parameter of 0.01. The payoff function is an estimate from Projection IRL: giving a payoff of -7.31 to census tracts in Staten Island, -10 to tracts in Manhattan, -9.16 to Brooklyn, -4.27 to the Bronx, -2.22 to Queens, 10 to areas outside NYC, and -3.85 to census tracts with airports. For this plot, I generated 25 datasets of 1,000 trajectories where a simulated taxi driver optimized over such a payoff function and transition probability function, and then I calculated the feature expectations for each of those 25 datasets. The red dashed lines show the expert feature expectations: the true discounted average number of times taxi drivers made a dropoff in each feature during the morning commute time period. This payoff function and transition smoothing parameter yields feature expectations close to that of the expert.

Figure 14: Feature Expectations under the Smoothing Parameter 0.10



Notes: The only difference between this figure and the plot in figure 13 is the selection for the smoothing parameter for the transition probability function: here, it is 0.10. As a result, the simulated feature expectations are a much worse fit for the expert feature expectations. Using a smoothing factor of 0.10 or higher, this pattern persists using any reasonable payoff function values, and Projection IRL always fails to get close to the expert feature expectations.

4 LIMITED ATTENTION AND NEW PRODUCT ADOPTION FOR MARIJUANA DISPENSARIES

4.1 Introduction

Neoclassical models of firm behavior start by asserting that firms seek to maximize an objective function related to profit. They deal with uncertainty through rational expectations and Bayesian learning. But Nobel Prize Winner Daniel Kahneman describes human nature as fundamentally anti-Bayesian. Probabilistically, the more observations a person has, the more confident they can be in their inference. But psychologically, confidence works in the reverse. Confidence is a feeling having to do with cognitive ease. The less a person knows, the more simple the explanation appears, and the more confident the person can feel (Kahneman, 2011). An overconfident agent is slow to update their prior. As summarized by Malmendier and Taylor (2015), overconfident agents go out of their way to find other explanations so they can avoid admitting their mistake. And the noisier the feedback, the more room there is to misattribute.

Patterns of misattribution, limited attention, and overconfidence may help to explain why small businesses remain small. Dispensary owners in Washington are limited to holding a maximum of three licenses at a time, although that constraint is not binding for the great majority. In the presence of licensing, an owner's superior decisionmaking will not lead to market concentration like in other industries. The market does not punish poor decisions as severely, and does not incentivize better decisions as strongly.

We know that people are overconfident, their attention is limited, and they will misattribute noise for true signals. But outside of the laboratory, it is not an easy task to measure agents' mistakes or their downstream consequences. There are two main reasons for this: the lack of available data and the lack of a plausible instrument.

The dataset I use is an unusually detailed account of firm decisionmaking. The Washington marijuana traceability system traces the product from seed to sale in order to keep the product out of the black market. A researcher can observe when any given dispensary ordered a certain product, the wholesale price they paid, how much they ordered, when they sold it to consumers, and at what final price.

A plausible instrument in this case must decompose the signal a dispensary receives about the profitability of different products into two parts: one that contains relevant information about the strength of demand for the product, and one that is pure noise. If the dispensary seems to make decisions based on pure noise, that is evidence of misattribution and limited attention. And if those decisions can be linked to lower profits in the long run, that is a way to ballpark the cost of limited attention.

In this paper, I use several instruments to try to accomplish this task. First, I use weather shocks (temperature and precipitation fluctuations) that effect foot traffic into brick and mortar stores like dispensaries. If weather shocks during a new product's debut has predictive power over a dispensary's reordering decisions, that is evidence that the owner or other decisionmaker may jump to conclusions based on limited data. The owner may be overconfident in the strength of demand for the product, when in reality the extra sales were because of favorable weather.

The second set of instruments I use are weather shocks interacted with owner characteristics like experience. The crucial assumption is that dispensary owners are the ones who make decisions about new product adoption. This may or may not be the case depending on the dispensary. But if owners with no previous experience tend to make more decisions based on pure noise than owners with previous experience, that measures the value of a business owner having previous experience. Another owner characteristic I use is the accumulated experience (whether or not the dispensary is in its first year) to test whether dispensaries learn to make better decisions over time. Finally, I also use the distance from the owner's home to the dispensary to test whether owners who live farther away seem to have a harder time being attentive to the way weather shocks impact sales at the dispensary. I follow Goldfarb and Xiao (2019) in using these owner characteristics as an instrument in this way, and broadly Gabaix (2019) to model decisionmakers as using rationally sparse models where paying attention is costly.

In general, I find that temperature and precipitation weather shocks are a weak instrument in this case and they explain very little of the total variation in the sales of a new product. I find no evidence of limited attention using temperature shocks, and no evidence that inexperienced owners are less attentive than owners with previous experience. First stages are not strong enough to make a claim about whether dispensary owners learn to pay closer attention over time to market variables, or whether living closer to the dispensary can be associated with the owner paying closer attention.

The last instrument I use is daily revenue at each dispensary. Foot traffic can vary for many reasons, including weather. But across the board, demand for new products and whether or not they are reordered depends heavily on daily revenue. This could be evidence of limited attention, but it could also be the impact of dispensary promotions or events that are not observable to the econometrician. I find that daily revenue during a product's debut week has a negative correlation with the total number of times a dispensary orders the product, conditional on the product being reordered once. This seems to indicate that the dispensary misattributes high foot traffic for strength of demand for the product at first, but after ordering it a few more times, learns that demand for the product is not as high as they initially thought and stops ordering it. But daily revenue during the debut of a new product can not be associated with a lower

average profit per order, indicating that the dispensary was not making a mistake when they were reordering the product.

4.2 Data

4.2.1 Traceability Data

Washington's Liquor and Cannabis Board regulates and audits license holders through a traceability system, and the resulting dataset is publicly available. Each transaction between growers, processors, dispensers, and consumers is recorded. The strength of this dataset is that it is reliable as well as detailed. Failing to properly account for each and every gram of marijuana puts a dispensary's license at risk. For this analysis, I use the first three years after the legalization of the sale of recreational marijuana, from July 2014 to July 2017.

During this period, the market was changing quickly, so an attentive owner could be especially effective. During the first month of sales in July 2014, only 18 dispensaries across the state were active. By the end of the year, that number grew to 84. By July 2015, there were 153 active dispensaries, after two years, 261, and after three years, 355. Only 17 dispensaries exited during the three year period, although 61 changed locations at least once. The price of a gram of marijuana fell from $20 to $8 in just one year after legalization (see Figure 15). But overall, revenue increased over time for active dispensaries. From July 2014 to July 2015, the median dispensary more than tripled their monthly sales (see Figure 16). Then from 2015 to 2016, revenue grew by 40.3%. Over the next year, revenue increased again by 10.2%.

Dispensaries opened with an initial stock of an average of 52 different products. They expanded their selections over time (see Figure 17). An average dispensary ordered 5.3 more products each month compared to the previous month. By 2017, the average dispensary was ordering 158 different products every month. 51.1% of products that are ordered once are never ordered again. If the product is ordered again, the dispensary takes 32 days on average to do so. But if the product becomes part of the dispensary's regular inventory, it's reordered every 14 days on average.

4.2.2 Weather Data

I use weather data from the National Oceanic and Atmospheric Administration over 2014-2017 from 62 weather stations across Washington state. The majority of dispensary locations are matched with a weather station within 5.5 miles, although for 5 percent of dispensary locations, the closest weather station is between 20 and 27 miles away.

The weather data includes the maximum daily temperature and the amount of

precipitation received during the day. Weather in Western Washington is very different from weather in Eastern Washington because the Cascade Mountain range divides Washington down the middle. The West is temperate and wet, and the East is dry with more extreme temperatures.

In the data, daily temperatures in Western Washington dispensaries and Eastern Washington dispensaries weren't too different in the Spring and Fall. But in the Summer, while the average daily high temperature in the West is 78 degrees, it's 87 in the East. And in the Winter, while the average daily low temperature in the West is 32 degrees, it's 22 in the East. The West also gets more rain and more rainy days than the East, especially during rainy months of the year, which includes every season except for the Summer. In the Fall, Winter, and Spring, Western Washington gets an average of 0.14 inches of rain each day, with 20 days with rain out of every month. Meanwhile during those months, Eastern Washington gets an average of 0.06 inches of rain each day, with an average of 12 days of rain each month.

4.2.3 Owner characteristics data

The data on dispensary owner characteristics is on the public record from Washington's Liquor and Cannabis Board. Starting in May 2014, the board began awarding licenses to operate marijuana dispensaries across the state. Applicants provided information on their past three occupations and their current residence.

For this analysis, I match dispensaries in the traceability data to the owner characteristics data. 152 owners out of 454, or 33.5%, own more than one dispensary. Of those, six own four dispensaries, 38 own three dispensaries, and the other 108 own two dispensaries. I define a variable for "previous experience" by whether or not the owner has the words "owner", "CEO", or "president" in their occupation history. I also record for how many years the owner held that position as a continuous measure for previous experience. 26.7% of owners have previous experience owning another business (see Table 10). Owners with multiple dispensaries are not more likely to have previous experience.

Conditional on the owner having previous experience, the median length of that occupation is 11 years. Twenty-five percent of owners with previous experience held that occupation for less than 4 years.

Using the owner's home address, I measure the distance between the owner's home and their dispensary. About half of the owners lived less than 15 miles from their dispensary and 23% lived less than 5 miles from their dispensary (see Table 11). Only two owners were living out of state at the time they applied, and no owners provided their home address as the dispensary's address.

For 23% of locations, the owner lived closer to the dispensary than the distance to

the nearest weather station. These dispensaries are removed from the dataset during the analysis of the effect of the owner's distance on their attention. For most locations however, the distance from the dispensary to the owner's home is more than three times as large as the distance from the dispensary to the closest weather station.

An owner's years of previous experience does not have a statistically significant effect on revenue, and neither does the distance from the owner's home to the dispensary. But on average, a dispensary made $60 more each day after they opened, and that effect is signficant at the .1% level (standard errors clustered at the dispensary level).

4.3 Model

Limited Attention Overall: I use an instrumental variables approach to identify limited attention by dispensary owners. I instrument debut sales using weather shocks. The dependent variable is whether or not the product was reordered by the dispensary. If the owner is inattentive to the impact of the weather shock on debut sales, they will misattribute the apparent popularity of the product to strength of demand for the product's characteristics and they will be more likely to reorder the product. The instrumental variables estimate for the relationship between debut popularity (instrumented by weather shocks) and the probability a product is reordered will be positive in the presence of limited attention.

The first stage of the model in equation 12 is a linear regression of weather shocks at time t at dispensary j on the logarithm of the percent of the initial order of the product that was sold during the first seven days after the dispensary started selling the product. I include fixed effects for the strain, the product (a strain and processor combination), the product combined with the month, the processor, the dispensary, the week of year, the month of year, the fiscal year, every Federal holiday, and April 20th. I include fixed effects for product-month combinations to at least partially account for lapses in the product's availability.

$$\log(\text{Debut Sales}_{ijt}) = \beta_0^1 + \beta_1^1 \text{Weather}_{jt} + \text{Product}_{it} + \text{Dispensary}_j + \text{Time} + u_{ijt} \quad (12)$$

The reduced form in equation 13 is a linear probability model of weather shocks at time t at dispensary j on whether or not the dispensary reordered the product. I use the same fixed effects as the first stage. If the weather shock increases debut sales and there is limited attention, the weather shock will increase the probability that the product is reordered and β_1^2 will be greater than zero. But if the weather shock decreases debut sales and there is limited attention, the weather shock will decrease the probability that the product is reordered and β_1^2 will be less than zero.

$$\text{Reorder}_{ijt} = \beta_0^2 + \beta_1^2 \text{Weather}_{jt} + \text{Product}_{it} + \text{Dispensary}_j + \text{Time} + u_{ijt} \quad (13)$$

Heterogeneity in Limited Attention: The next instruments I use are weather shocks interacted with three different owner characteristics that could be associated with inattentiveness: the distance from the owner's house to the dispensary, the years of previous experience the owner has, and the days since the dispensary opened. Owners who live farther away from their dispensaries may have a harder time mentally separating the impact of weather shocks from signals about the true strength of demand for a product. Also, owners may get better at that task the more experience they accumulate. Using the weather shock-owner characteristic interactions as instruments for debut sales lets me explore the possibility that inattentiveness is heterogeneous or that inattentiveness is present initially but disappears with experience.

The first stage of this model in equation 14 is a linear regression of weather shocks interacted with the owner characteristic (a proxy for attentiveness) at time t at dispensary j on the logarithm of the percent of the initial order of the product that was sold during the first seven days after the dispensary started selling the product. I include all fixed effects from the previous model.

$$\begin{aligned}\log(\text{Debut Sales}_{ijt}) = \beta_0^3 + \beta_1^3 \text{Weather}_{jt} + \beta_2^3 \text{Inattentive}_{jt} + \beta_3^3 \text{Weather} * \text{Inattentive}_{jt} \\ + \text{Product}_{it} + \text{Dispensary}_j + \text{Time} + u_{ijt}\end{aligned} \quad (14)$$

If the first stage is strong, weather shocks change the debut sales for a new product, and β_1^3 will not be zero. It will be important to make sure β_3^3 is zero. If it is not, then weather shocks could effect debut sales differently for different types of owners, which would cast doubt on the reduced form interpretations.

The reduced form in equation 15 is a linear probability model of weather shocks interacted with the owner characteristic (again, a proxy for attentiveness) on whether the product was reordered at dispensary j. The same fixed effects are included as before.

$$\begin{aligned}\text{Reorder}_{ijt} = \beta_0^4 + \beta_1^4 \text{Inattentive}_{jt} + \beta_2^4 \text{Weather}_{jt} + \beta_3^4 \text{Inattentive} * \text{Weather}_{jt} \\ + \text{Product}_{it} + \text{Dispensary}_j + \text{Time} + u_{ijt}\end{aligned} \quad (15)$$

The reduced form in equation 15 allows attentive and inattentive owners to have different average rates of reordering with β_1. For instance, if inexperienced owners reorder a smaller fraction of the products they debut compared to experienced owners, that would not point toward their inattention. Inattentive is a dummy variable

identifying potentially less attentive owners (owners without previous experience, owners in their first year of business, and owners who live farther from the dispensary). So β_2^4 is the impact of weather shocks on reording for the attentive owners and $\beta_2^4 + \beta_3^4$ is the impact of weather shocks on reordering for inattentive owners. If the weather shock can be associated with higher debut sales and a higher probability of reordering for the inattentive, $\beta_4^3 > 0$ is evidence that the inattentive group makes reordering decisions more frequently based on weather shocks than the attentive group. On the other hand, if the weather shock can be associated with lower debut sales and a lower probability of reordering for the inattentive, $\beta_4^3 < 0$ is evidence that the inattentive group makes reordering decisions more frequently based on weather shocks than the attentive group.

Validity of Weather Instrument: Temperature fluctuations account for only .0015% of the total variation in the debut sales for a product, and precipitation shocks account for .023%, so the strength of the first stage is a concern. Nevertheless, I select the weather shock-season combinations that have the strongest first stage. The exclusion restriction is very plausible because the decision to reorder a product should depend only on the owner's estimation of its relative profitability and certainly not directly on the weather. Weather shocks are exogenous as long as dispensaries do not have a tendency to time the debut of certain new products to align with heat waves, cold fronts, and storms.

4.4 Results

The first stage results for the impact of precipitation-related weather shocks on a product's debut sales is summarized in Table 12 and visualized in Figures 26 and 27. Overall the first stage for precipitation-related weather shocks are weaker than the temperature-related weather shocks. Over the entire year, if a product debuts during a week that is wetter than average, I estimate that the dispensary will sell 0.14% less out of the initial order, but that estimate is not significant at the 10% level for the entire year or during any season in particular. I estimate that during wetter than usual weeks, the quantity of precipitation cannot be associated with a product's debut sales during the entire year or any season in particular. I also estimate that during drier than usual weeks, debut sales increase with the quantity of precipitation by 0.51% per tenth of a milimeter of rain, and that effect is signficant at the 5% level. That effect is driven by precipitation in the Summer and Fall. Given the fact that precipitation and temperature are strongly correlated and other estimates for the effect of precipitation on debut sales were insignificant, I do not move forward with using precipitation shocks as an instrument for debut sales and I instead use temperature shocks.

The first stage results for the impact of temperature-related weather shocks on a product's debut sales is summarized in Table 13 and visualized in Figures 28 and 29. If

a product makes its debut during a week that is warmer than average, it tends to sell .2% less out of the initial order, but that estimate is not statistically significant. During weeks that are colder than average, every degree farenheit colder predicts .1% fewer sales for the new product, and the effect is signficant at the 5% level during the Summer and the Fall. The strongest first stage patterns for weather shocks on debut sales are that cool temperatures outside of the Spring can be associated with lower sales for the new product. I focus on this weather shock pattern moving forward.

The popularity of a new product after its debut (as measured by the amount sold during the first seven days out of the total order) is a strong predictor of whether the product will be reordered by a dispensary. Including various fixed effects, a one percent increase in the amount sold out of the total order during the first seven days can be associated with the product being .260% (.007%) more likely to be reordered.

Table 14 reports the reduced form estimates for the effect of temperature fluctuations on whether the product was reordered. I find that when temperatures are colder than average, an extra degree colder can be associated with a .05% lower probability that the product will be reordered, although the effect is not statistically signficiant. Table 15 reports the two stage least squares estimates for the impact of debut sales on whether the product was reordered, instrumented by the temperature shocks. I find no evidence of overconfidence or limited attention across the entire year or any season in particular.

Table 16 reports the first stage estimates for weather shocks on debut sales for a new product, by owner experience. Across the entire year, during colder than usual weeks both for experienced and inexperienced owners, every degree farenheit colder the week is on average, a new product tends to sell .15% less, and the effect is significant at the 5% level. That pattern is strongest in the Fall, when an extra degree colder can be associated with .6% fewer sales for a new product for experienced owners, but no such result for inexperienced owners. The first stage is weak in the Spring and in the Summer, but in the Winter, extra degrees farenheit during warmer than usual weeks can be associated with .48% fewer sales both for experienced and inexperienced owners.

Table 17 reports the reduced form estimates for weather shocks on the probability the product is reordered, by owner experience. Given the first stage results from Table 16, I look for three patterns: cold temperatures reducing the probability a product is reordered during the entire year for experienced and inexperienced owners both, cold temperatures reducing the probability a product is reordered during the Fall for experienced owners, and warm temperatures in the Winter reducing the probability a product is reordered for both inexperienced and experienced owners.

The first effect does not hold. Although cold temperatures lower debut sales, there is not sufficient evidence that they decrease the probability a product is reordered for experienced or inexperienced owners. The second effect also fails to hold. Cold

temperatures lower debut sales in the Fall for experienced owners, but they have no effect on the probability a product is reordered for that group. Finally, there's some evidence supporting the last effect: warm temperatures during the Winter lower debut sales and lower the probability that a product will be reordered, but inexperienced owners don't seem to be more or less aware of the impact of weather than experienced owners.

In tables 18 and 19, I repeat the analysis from table 16, substituting the owner's previous experience with the distance from the owner's home to the dispensary in Table 18 and then the owner's accumulated experience at the dispensary in Table 19. I do not find a strong first stage for temperature shocks in either case.

The last instrument I use is the daily revenue at the dispensary (summed over the course of the debut week) as a proxy for foot traffic. Table 20 shows that this instrument has a much stronger first stage than the weather shock instruments. A 1% increase in weekly revenue can be associated with a .2% increase in the debut sales for a new product, and the effect is signficant at the .1% level. Table 20 also shows that foot traffic does not effect debut sales differently for owners with or without previous experiene, nor for dispensaries inside or outside of their first year.

I report the two-stage least squares results for the weekly revenue instrument in table 21. When weekly revenue is 1% higher, a new product is .05% more likely to be reordered, and that effect is significant at the .1% level. This could be because of limited attention, but also it could be because dispensaries may time the debut of certain new product to coincide with special promotions or events. When I filter on the owner's previous experience, I find that for owners with previous experience, when weekly revenue is 1% higher, a new product is .11% more likely to be reordered, compared to just .03% more likely for owners without previous experience. When I filter on the dispensary's number of years being open, I find a similar pattern: dispensaries in their second or third years see an 11% increase in the probability a new product will be reordered following 1% higher weekly sales, compared to .04% for dispensaries in their first year.

4.5 Conclusion

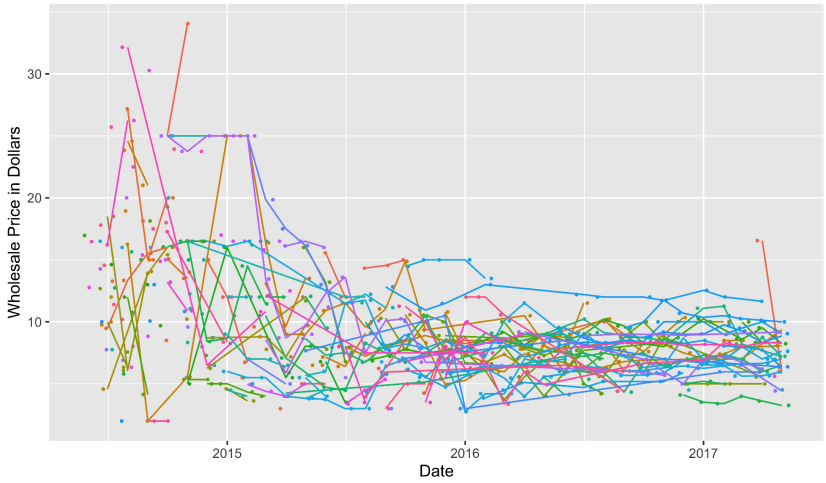
When small business owners select a product to sell, the decisionmaking process is not likely to be scientific. That's because these types of decisions are made every day, trying to predict which products will be the most profitable is next to impossible, and after an agent makes a selection, they only receive noisy signals about the quality of those selections. These types of tasks encourage agents to be overconfident and focus their attention on just a few memorable events and neglect the rest.

However, it is not a simple task to identify limited attention in inventory selection

decisions. I find that temperature fluctuations only account for .0015% of the total variation in the debut sales of a new product and first stages are weak. Weekly revenue has a much stronger first stage, but a potential problem with endogeneity. Dispensaries may time the debut of certain new products to coincide with special promotions or events. I find that the dispensary's revenue has predictive power over whether a new product will be reordered, but counterintuitively, that effect is stronger for owners with previous experience and dispensaries in their second or third years as opposed to their first years. This is likely not because owners make more mistakes as they become more experienced. Instead, owners may become more strategic about debuting promising new products during promotional events.

4.6 Tables and Figures

Figure 15: Median Wholesale Price for a Gram of the Twenty Most Popular Strains Each Month



Note: I took Traceability data on the orders dispensaries placed with processors, filtered on the twenty most frequently ordered strains each month, and calculated the median wholesale price each dispensary paid for a gram of that strain. Wholesale prices were erratic during the first year of business, but were falling and stabilized during the second and third years.
Note: I summed the monthly sales for each dispensary and drew fifty randomly to avoid too much visual noise. Drawing any forty dispensaries will yield this same general pattern.

Figure 16: Monthly Revenue for Forty Dispensaries

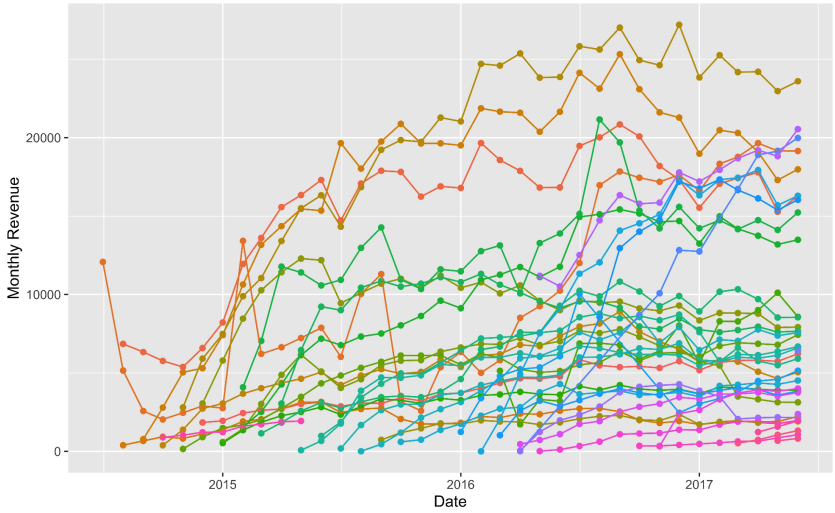
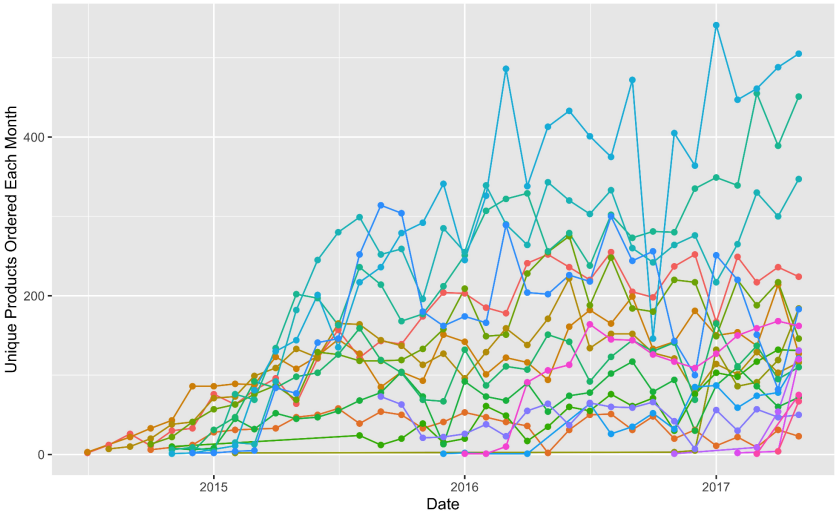


Figure 17: Number of Unique Products Ordered by Dispensaries Each Month



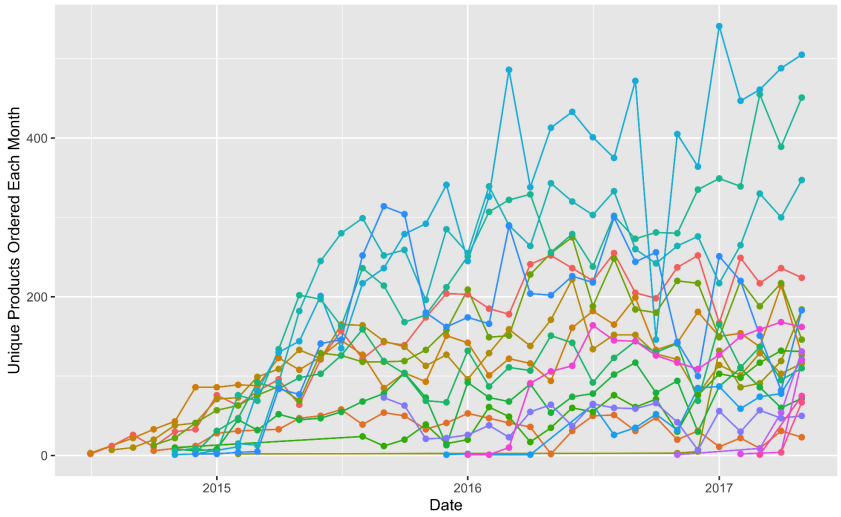
Note: I counted the number of unique products ordered by each dispensary, each month in the Traceability data. This shows data from thirty random dispensaries, but any random draw of dispensaries yields the same pattern.

Table 10: Distribution of Dispensaries by Years of Previous Experience

Years of Previous Experience	Number of Dispensaries
0	199
1–5	34
6–10	9
11–15	21
16–20	7
21–25	11
26–30	6
31–40	4

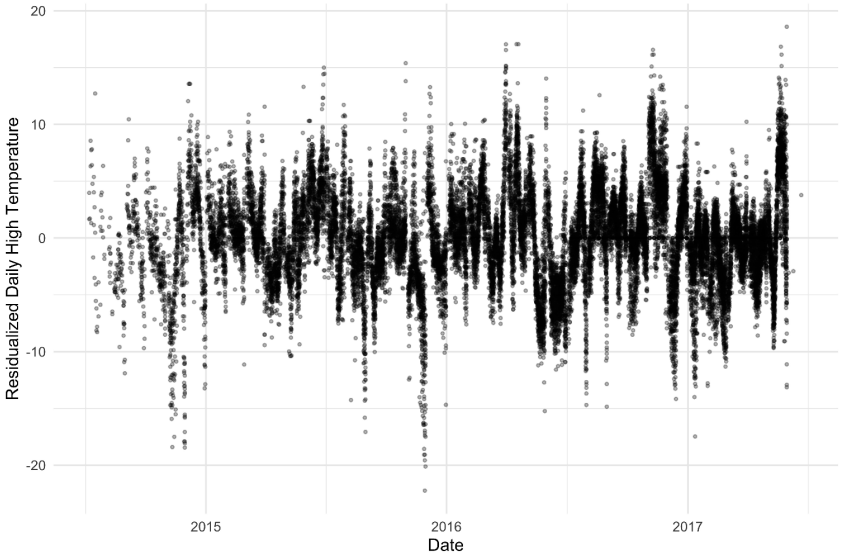
Note:I analyze license applications to determine the owner or sole proprietor of the dispensary. If their employment history mentioned "owner", "president", or "CEO", I record that the owner has previous experience and I record the number of years they held that job as the years of previous experience attained. When there is more than one owner listed, I record the years of previous experience as the maximum out of all owners in the application.

Figure 18: Seasonality of Daily High Temperatures



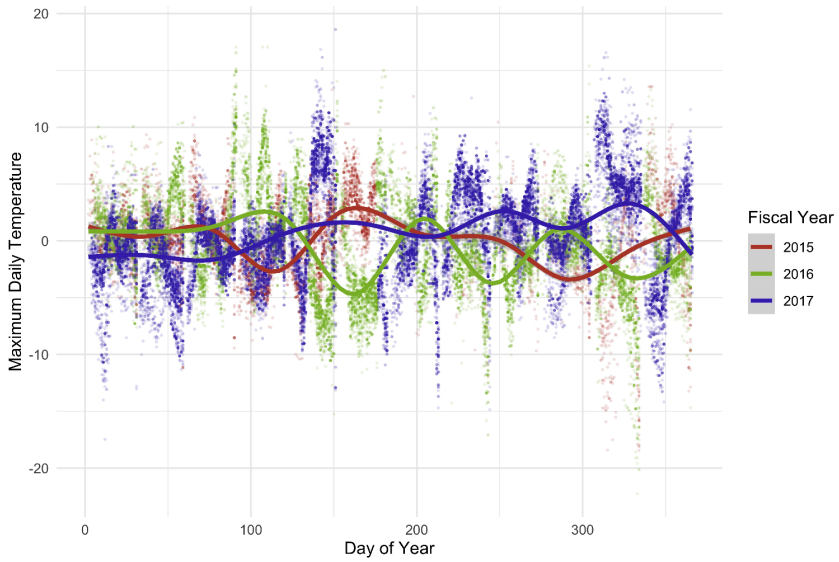
Note: The data, collected online from NOAA, measures the daily maximum temperature in degrees farenheit at the weather station closest to each dispensary from July 2014 to July 2017.

Figure 19: Residual Daily High Temperatures



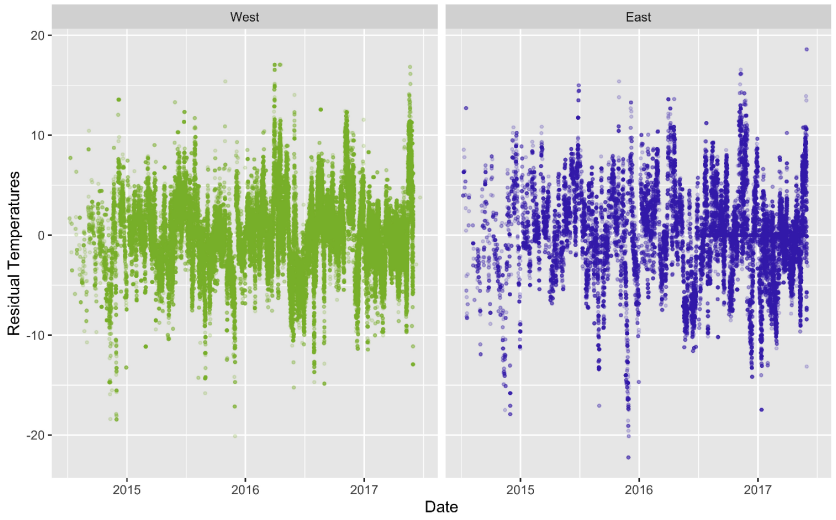
Note: Residualized daily NOAA temperatures based on the closest weather station to each dispensary. The residuals were constructed by running a linear regression and controlling for year, month of year, week of year, weather station, and the interaction of weather station and week of year. The result is roughly consistent with white noise.

Figure 20: Another View of Residual Daily High Temperatures



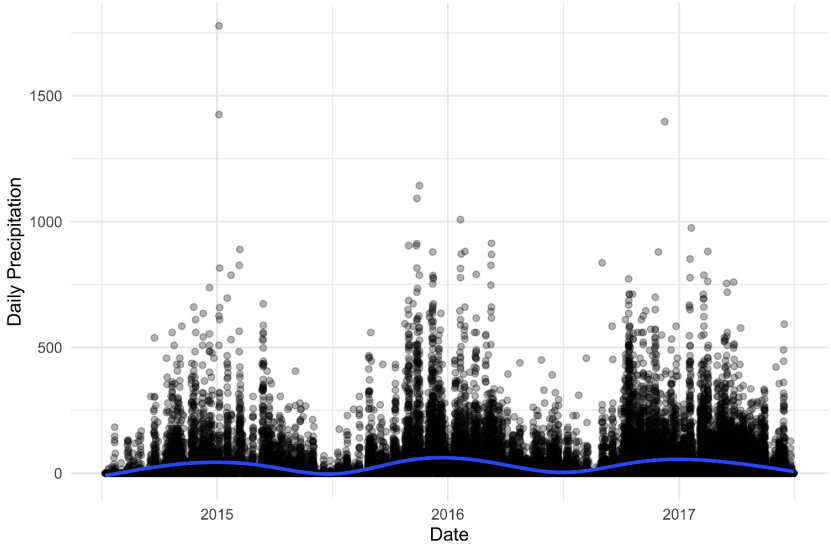
Note: Residualized daily NOAA temperatures based on the closest weather station to each dispensary, constructed by running a linear regression and controlling for year, month of year, week of year, weather station, and the interaction of weather station and week of year. Visual inspection points toward the existence of autocorrelation, but not seasonality, because as temperatures increase in one year, they decrease in another.

Figure 21: Residual Daily High Temperatures In Eastern versus Western Washington



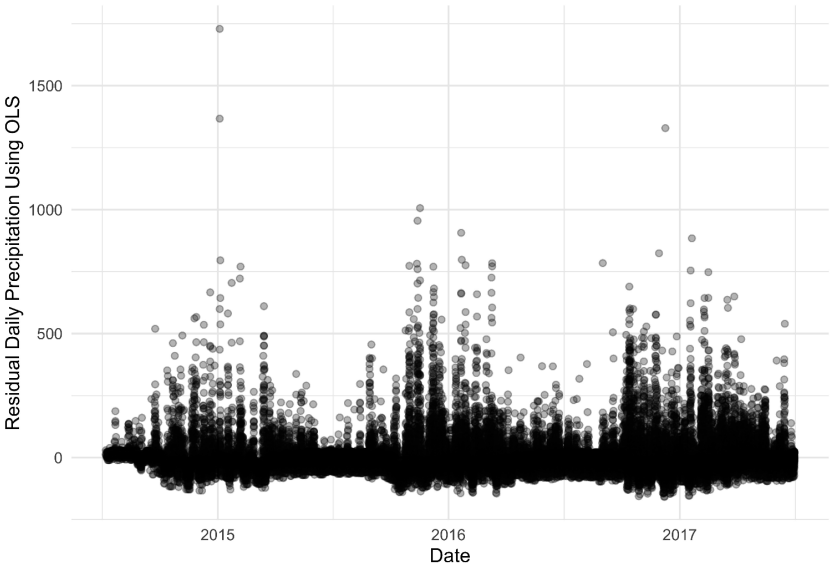
Note: Residualized daily NOAA temperatures based on the closest weather station to each dispensary, constructed by running a linear regression and controlling for year, month of year, week of year, weather station, and the interaction of weather station and week of year. Dispensaries to the East of -121 degrees are Eastern and the rest are Western.

Figure 22: Seasonality of Precipitation



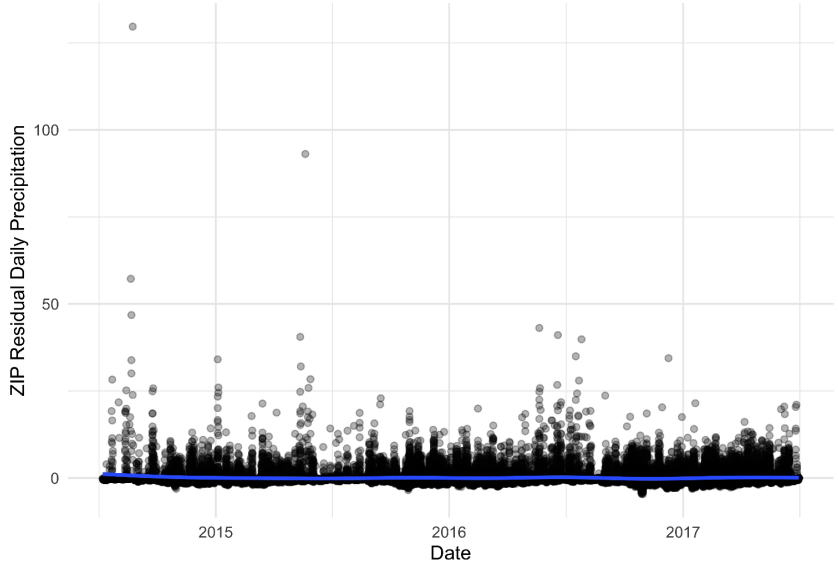
Note: Daily NOAA precipitation in tenths of milimeters at each dispensary, based on measurement at the closest weather station.

Figure 23: OLS Precipitation Residuals



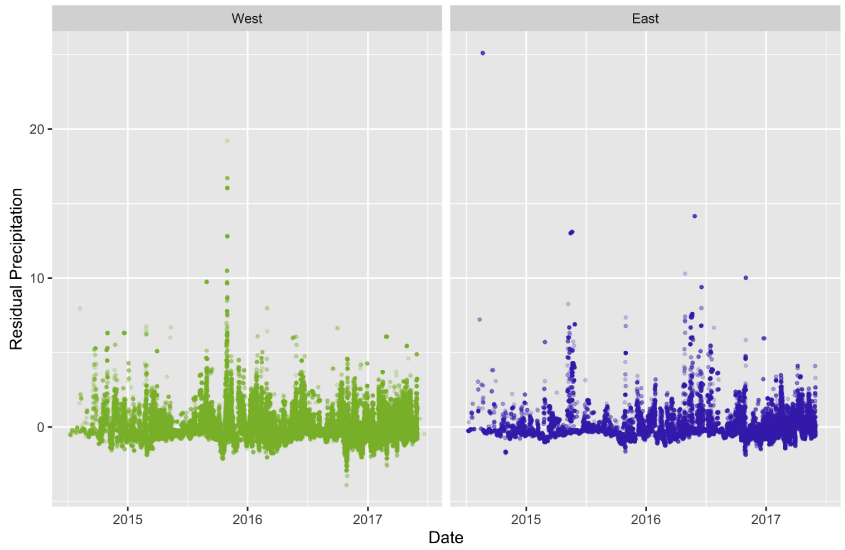
Note: Residualized daily NOAA precipitation in tenths of milimeters at each dispensary, based on measurement at the closest weather station. The residuals were constructed by running a linear regression and controlling for year, month of year, week of year, and location. OLS residualization does not help remove seasonality in this case.

Figure 24: ZIP Precipitation Residuals



Note: Residualized daily NOAA precipitation in tenths of milimeters at each dispensary, based on measurement at the closest weather station. The residuals were constructed by fitting a Zero-Inflated Poisson (ZIP) regression and controlling for year, month of year, week of year, and location in both the count and the inflation portions of the model. ZIP residualization helps remove the seasonality of daily precipitation.

Figure 25: ZIP Precipitation in Western versus Eastern Washington



Note: Residualized daily NOAA precipitation in tenths of milimeters at each dispensary, based on measurement at the closest weather station. The residuals were constructed by fitting a Zero-Inflated Poisson (ZIP) regression and controlling for year, month of year, week of year, and location in both the count and the inflation portions of the model. Dispensaries to the East of -121 degrees are Eastern and the rest are Western.

Table 11: Distance From Owner's Home Address to Their Dispensary

Distance from Home to Dispensary	Number of Dispensaries
0–2 miles	33
2–5 miles	48
5–10 miles	63
10–15 miles	30
15–20 miles	17
20–30 miles	26
30–40 miles	11
40–50 miles	12
50–100 miles	18
100–200 miles	21
>200 miles	12

Note: I analyze license applications to determine the owner or sole proprietor of the dispensary. I measure the distance between their home address and the dispensary's location. When there is more than one owner listed, I use the minimum of the distances between home addresses and the dispensary location.

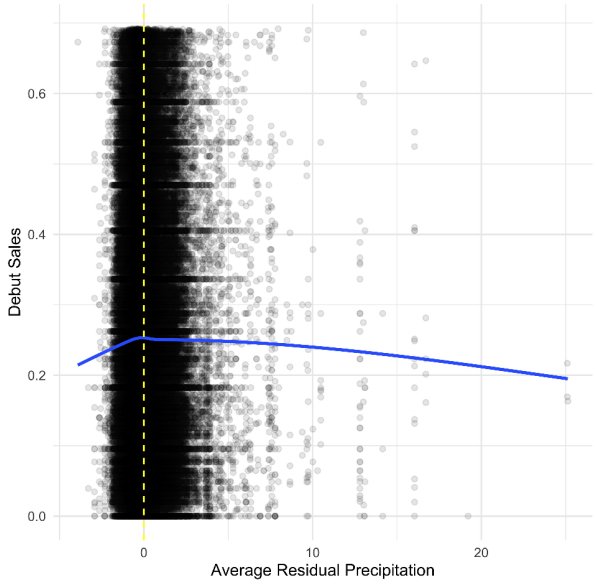
Table 12: First Stage Results: Effect of Precipitation Shocks on a Product's Debut Sales

	Entire Year	Spring	Summer	Fall	Winter
Precipitation > 0	−0.00144	−0.00332	−0.00734	−0.00100	0.00458
	(0.00183)	(0.00330)	(0.00485)	(0.00365)	(0.00339)
Precipitation	0.00051**	0.00040	0.01702*	0.00903***	−0.00243
	(0.00020)	(0.0041)	(0.00994)	(0.00340)	(0.00352)
(Precip > 0) * Precip	−0.00459**	−0.00366	−0.01570	−0.00887**	0.00403
	(0.00214)	(0.00417)	(0.01045)	(0.00340)	(0.00433)
N	186,115	63,449	36,665	40,420	45,581

*p < 0.1; **p < 0.05; ***p < 0.01

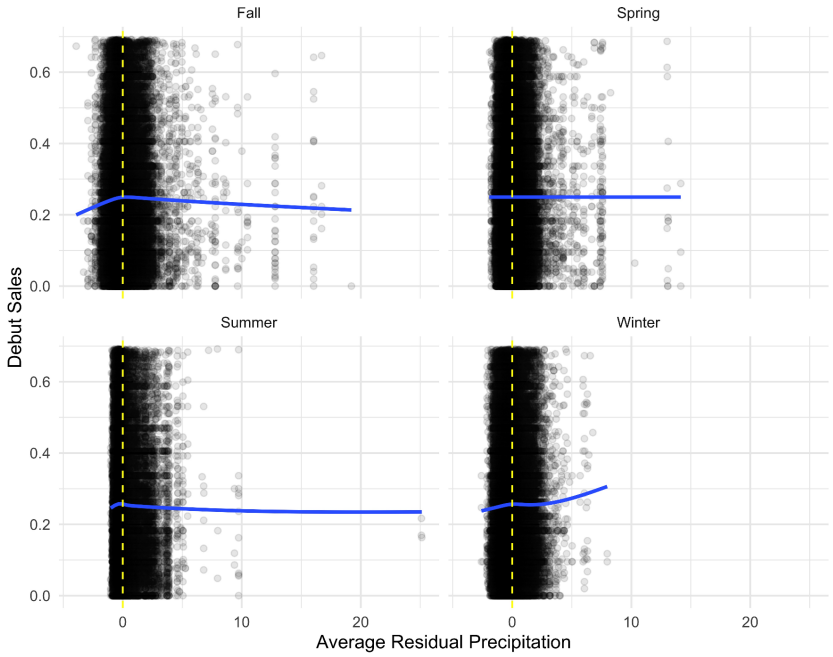
Note: I estimate the relationship between weekly average precipitation at the dispensary and debut sales. The model is fit using data across the entire year, and then for each season. Precipitation is residualized by estimating a zero-inflated poission model and controlling for year, month of year, week of year, and location in both the count and the inflation portions of the model. Debut sales at a dispensary during the debut week is measured by the percent out of the total order that was sold during the first seven days after the product was debuted, in log terms. The explanatory variables allow the relationship between precipitation and debut sales to have different intercepts and slopes for weeks that are wetter than average versus weeks that are drier than average. I include fixed effects for the strain, the product (a strain and processor combination), the product combined with the month, the processor, the dispensary, the week, the month, the fiscal year, every federal holiday, and April 20th. Standard errors that are clustered by dispensary and robust to heteroskedasticity are in parentheses.

Figure 26: ZIP Precipitation in Western versus Eastern Washington



Note: Residualized daily NOAA precipitation in tenths of milimeters at each dispensary, based on measurement at the closest weather station. The residuals were constructed by fitting a Zero-Inflated Poisson (ZIP) regression and controlling for year, month of year, week of year, and location in both the count and the inflation portions of the model. Dispensaries to the East of -121 degrees are Eastern and the rest are Western.

Figure 27: Effect of Residual Precipitation on Debut Sales by Season



Note: Precipitation is residualized by fitting a zero-inflated poisson model and controlling for year, month of year, week of year, and location in both the count and the inflation portions of the model. Debut sales at a dispensary during the debut week is measured by the percent out of the total order that was sold during the first seven days after the product was debuted, in log terms. Blue lines are fitted using a gamma smoothing function.

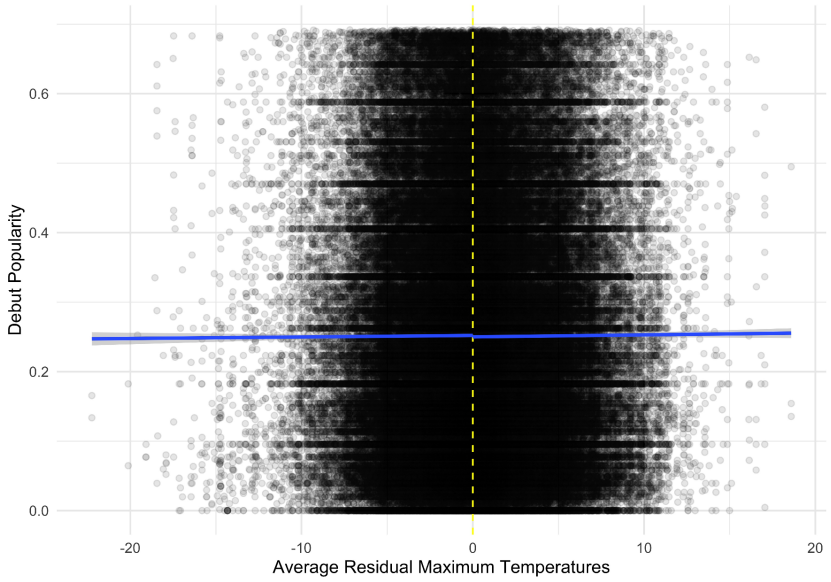
Table 13: Effect of Residual Precipitation on Debut Sales by Season

	Entire Year	Spring	Summer	Fall	Winter
Temp > 0	−0.00211	0.00074	−0.00513	−0.00184	−0.00258
	(0.00167)	(0.00273)	(0.00421)	(0.00362)	(0.00350)
Temperature	0.00109***	−0.00056	0.00181**	0.00183**	0.00122
	(0.00041)	(0.00071)	(0.00077)	(0.00091)	(0.00085)
(Temp > 0) * Temp	−0.00110*	−0.00093	−0.00085	−0.00207	−0.00296**
	(0.00057)	(0.00095)	(0.00119)	(0.00132)	(0.00149)
N	186,115	63,449	36,665	40,420	45,581

*p < 0.1; **p < 0.05; ***p < 0.01

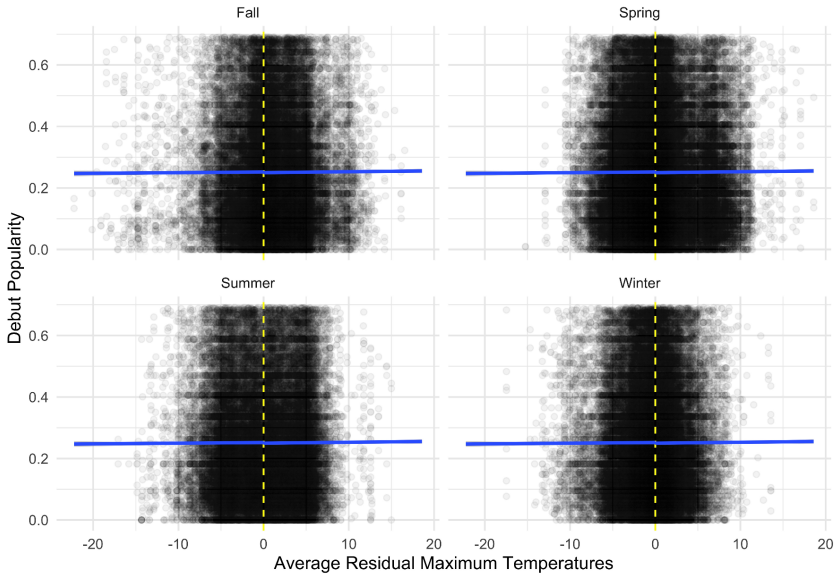
Note: Precipitation is residualized by fitting a zero-inflated poisson model and controlling for year, month of year, week of year, and location in both the count and the inflation portions of the model. Debut sales at a dispensary during the debut week is measured by the percent out of the total order that was sold during the first seven days after the product was debuted, in log terms. Blue lines are fitted using a gamma smoothing function.

Figure 28: Effect of Temperatures on Debut Popularity



Note: On the horizontal axis I plot the weekly average of the residual daily maximum temperatures at each dispensary, constructed by running a linear regression and including controls for week of year, month of year, year, and location. On the vertical axis I plot the percent of the total order sold during the first seven days after a product's debut, in log terms. Cold temperatures don't seem to effect debut popularity, but for warmer than average weeks, as temperatures increase, debut popularity falls.

Figure 29: Effect of Temperatures on Debut Popularity by Season



Note: On the horizontal axis I plot the weekly average of the residual daily maximum temperatures at each dispensary, constructed by running a linear regression and including controls for week of year, month of year, year, and location. On the vertical axis I plot the percent of the total order sold during the first seven days after a product's debut, in log terms. Debut popularity seems to decrease with warmer temperatures in every season.

Table 14: Reduced Form Results: Effect of Temperature Shocks on a Product's Probability of Reorder

	Entire Year	Spring	Summer	Fall	Winter
Temp > 0	0.00388	0.01468**	0.00794	−0.01677**	−0.00258
	(0.00418)	(0.00661)	(0.01002)	(0.00836)	(0.00350)
Temperature	0.00051	−0.00366**	0.00130	−0.00108	0.00122
	(0.00095)	(0.00179)	(0.00194)	(0.00229)	(0.00085)
(Temp > 0) * Temp	−0.00266*	0.00242	−0.00076	−0.00524*	−0.00296**
	(0.00144)	(0.00251)	(0.00270)	(0.00318)	(0.00149)
N	186,115	63,449	36,665	40,420	45,581

*p < 0.1; **p < 0.05; ***p < 0.01

Note: I estimate the linear probability model for residualized temperatures, the weekly temperature being higher than normal, and the interaction of those two variables on whether the product is reordered. Fixed effects for strain, the product, the product and the month, the dispensary, processor, month, week, all Federal holidays, and April 20th are included. Heteroskedasticity robust standard errors that are clustered at the dispensary level are reported in parentheses. The model is fit using data across the entire year, and then for each season.

Table 15: 2SLS Results

	Entire Year	Spring	Summer	Fall	Winter
Debut Sales (fitted)	0.2417	−1.372	1.1439	−1.273	1.804
	(0.8436)	(2.783)	(0.9559)	(1.470)	(1.172)
N	186,115	63,449	36,665	40,420	45,581

*p < 0.1; **p < 0.05; ***p < 0.01

Note: I instrument debut sales by residualized temperature, the residualized temperatures being greater than zero, and the interaction of those two variables. I estimate the linear probability model of the fitted vales for debut sales on whether the product was reordered. Fixed effects for strain, the product, the product and the month, the dispensary, processor, month, week, all Federal holidays, and April 20th are included. Heteroskedasticity robust standard errors that are clustered at the dispensary level are reported in parentheses. The model is fit using data across the entire year, and then for each season.

Table 16: First Stage Results: Effect of Temperature Shocks on a Product's Debut Sales By Owner Previous Experience

	Entire Year	Spring	Summer	Fall	Winter
Temperature	0.00150**	−0.00164	0.00161	0.00609***	0.00053
	(0.00065)	(0.00137)	(0.00117)	(0.00149)	(0.00136)
Temp > 0	−0.00589**	0.00169	−0.01306	−0.01006	−0.00021
	(0.00289)	(0.00586)	(0.00792)	(0.00709)	(0.00588)
(Temp > 0) * Temp	−0.00145	0.00185	−0.00022	−0.00579***	−0.00480*
	(0.00099)	(0.00168)	(0.00190)	(0.00203)	(0.00259)
Temperature * Inexperienced	−0.00079	0.00252*	−0.00003	−0.00637***	0.00161
	(0.00075)	(0.00146)	(0.00140)	(0.00170)	(0.00157)
(Temp > 0) * Inexperienced	0.00438	−0.00215	0.01051	0.01162	−0.00405
	(0.00349)	(0.00676)	(0.00965)	(0.00812)	(0.00700)
Temp * (Temp > 0) * Inexperienced	0.00095	−0.00307	0.00023	0.00625***	0.00043
	(0.00104)	(0.00176)	(0.00224)	(0.00204)	(0.00297)
N	155,854	53,284	30,640	33,728	38,202

*p < 0.1; **p < 0.05; ***p < 0.01

Note: I estimate the linear regression for residualized temperatures, the weekly temperature being higher than normal, the owner having no previous experience, and the interaction of those three variables on the percentage of the initial order of a product that is sold during the first seven days after the order is received. Fixed effects for strain, the product, the product and the month, the dispensary, processor, month, week, all Federal holidays, and April 20th are included. Heteroskedasticity robust standard errors that are clustered at the dispensary level are reported in parentheses. The model is fit using data across the entire year, and then for each season.

Table 17: Reduced Form Results: Effect of Temperature Shocks on a Product's Probability of Reorder By Owner Previous Experience

	Entire Year	Spring	Summer	Fall	Winter
Temperature	0.001888	−0.002809	0.00008	−0.005219	0.002971
	(0.001483)	(0.003251)	(0.00294)	(0.003567)	(0.003662)
Temp > 0	−0.001483	0.002941	−0.006954	−0.01969	0.014313
	(0.00727)	(0.013291)	(0.01649)	(0.016473)	(0.01420)
(Temp > 0) * Temp	−0.003785	0.003064	0.005317	−0.01257***	−0.01113*
	(0.002466)	(0.004299)	(0.004851)	(0.00455)	(0.00608)
Temperature * Inexperienced	−0.001588	−0.000500	0.000977	−0.00845**	0.002399
	(0.001755)	(0.003666)	(0.00352)	(0.004206)	(0.003921)
(Temp > 0) * Inexperienced	0.010520	0.014973	0.02161	0.005386	0.001522
	(0.009111)	(0.016110)	(0.02021)	(0.01913)	(0.01709)
Temp * (Temp > 0) * Inexperienced	0.001094	−0.001001	−0.00659	0.010128**	−0.00246
	(0.002517)	(0.004445)	(0.005455)	(0.005081)	(0.00664)
N	155,854	53,284	30,640	33,728	38,202

*p < 0.1; **p < 0.05; ***p < 0.01

Note: I estimate the linear probability model for residualized temperatures, the weekly temperature being higher than normal, the owner having no previous experience, and the interaction of those three variables on whether the product is reordered. Fixed effects for strain, the product, the product and the month, the dispensary, processor, month, week, all Federal holidays, and April 20th are included. Heteroskedasticity robust standard errors that are clustered at the dispensary level are reported in parentheses. The model is fit using data across the entire year, and then for each season.

Table 18: First Stage Results: Effect of Temperature Shocks on a Product's Debut Sales By Distance to Owner's Home

	Entire Year	Spring	Summer	Fall	Winter
Temperature	0.00060	−0.00070	0.00134	0.00179	0.00073
	(0.00052)	(0.00095)	(0.00104)	(0.00113)	(0.00114)
Temp > 0	−0.00234	0.00035	−0.00797	−0.00186	−0.00460
	(0.00220)	(0.00361)	(0.00549)	(0.00494)	(0.00442)
(Temp > 0) * Temp	−0.00048	0.00029	0.000196	−0.00192	−0.00299
	(0.00073)	(0.00118)	(0.00153)	(0.00157)	(0.00210)
Temperature * (Distance > 20m)	0.00117	0.00269*	0.00091	−0.00123	0.00309
	(0.00083)	(0.00149)	(0.00158)	(0.00199)	(0.00172)
(Temp > 0) * (Distance > 20m)	−0.00176	−0.01093	0.00698	0.00058	0.00476
	(0.00411)	(0.00663)	(0.00969)	(0.00913)	(0.00735)
Temp * (Temp > 0) * (Distance > 20m)	−0.00104	−0.00198	0.00077	0.00236	−0.00486
	(0.00107)	(0.00174)	(0.00264)	(0.00250)	(0.00305)
N	155,854	53,284	30,640	33,728	38,202

*p < 0.1; **p < 0.05; ***p < 0.01

Note: I estimate the linear regression for residualized temperatures, the weekly temperature being higher than normal, the owner living more than 20 miles from the dispensary, and the interaction of those three variables on the percentage of the initial order of a product that is sold during the first seven days after the order is received. Fixed effects for strain, the product, the product and the month, the dispensary, processor, month, week, all Federal holidays, and April 20th are included. Heteroskedasticity robust standard errors that are clustered at the dispensary level are reported in parentheses. The model is fit using data across the entire year, and then for each season.

Table 19: First Stage Results: Effect of Temperature Shocks on a Product's Debut Sales By Accumulated Experience at the Dispensary

	Entire Year	Spring	Summer	Fall	Winter
Temperature	0.00074	0.00042	0.00115	0.00229	0.00149
	(0.00059)	(0.00108)	(0.00118)	(0.00166)	(0.00122)
Temperature > 0	−0.00188	−0.00229	0.00259	−0.00338	0.00292
	(0.00265)	(0.00443)	(0.00745)	(0.00599)	(0.00532)
Days Open < 366	0.00452	−0.00331	0.00653	0.00314	0.01623**
	(0.00384)	(0.00613)	(0.00820)	(0.00815)	(0.00731)
(Temp > 0) * Temp	−0.00064	−0.00069	−0.00042	−0.00186	0.00447*
	(0.00087)	(0.00135)	(0.00211)	(0.00209)	(0.00245)
Temp * (Days Open < 366)	0.00036	−0.00057	0.00074	−0.00107	−0.00002
	(0.00076)	(0.00148)	(0.00147)	(0.00169)	(0.00169)
(Temp > 0) * (Days Open < 366)	−0.00187	0.00477	−0.01272	0.00306	−0.01057
	(0.00320)	(0.00637)	(0.00937)	(0.00746)	(0.00695)
Temp * (Temp > 0) * (Days Open < 366)	−0.00019	0.00081	0.00050	0.00068	0.00041
	(0.00105)	(0.00170)	(0.00244)	(0.00213)	(0.00294)
N	155,854	53,284	30,640	33,728	38,202

*p < 0.1; **p < 0.05; ***p < 0.01

Note: I estimate the linear regression for residualized temperatures, the weekly temperature being higher than normal, the dispensary being in its first year, and the interaction of those three variables on the percentage of the initial order of a product that is sold during the first seven days after the order is received. Fixed effects for strain, the product, the product and the month, the dispensary, processor, month, week, all Federal holidays, and April 20th are included. Heteroskedasticity robust standard errors that are clustered at the dispensary level are reported in parentheses. The model is fit using data across the entire year, and then for each season.

Table 20: First Stage Results: Effect of Weekly Revenue on a Product's Debut Sales

	(1)	(2)	(3)
Weekly Revenue	0.20619***	0.21740***	0.204625***
	(0.00942)	(0.01282)	(0.010939)
Weekly Revenue*Experienced		−0.03077	
		(0.02001)	
Days Open > 365			−0.04601
			(0.19264)
Weekly Revenue*(Days Open > 365)			0.00924
			(0.01793)
N	279,815	235,771	235,771

*p < 0.1; **p < 0.05; ***p < 0.01

Note: I estimate the relationship between log weekly revenue at a dispensary and debut sales for a new product at that dispensary, also in log terms. Model (1) reports the simple regression estimate. Model (2) includes an interaction term between log weekly revenue and whether the owner has previous experience. Model (3) includes an interaction term between log weekly revenue and whether the dispensary is in its second or third year as opposed to its first year. Debut sales at a dispensary during the debut week is measured by the percent out of the total order that was sold during the first seven days after the product was debuted. I include fixed effects for the strain, the product (a strain and processor combination), the product combined with the month, the processor, the dispensary, the week, the month, the fiscal year, every federal holiday, and April 20th. Standard errors that are clustered by dispensary and robust to heteroskedasticity are in parentheses.

Table 21: 2SLS Results: Effect of Weekly Revenue on a Product's Probability of Being Reordered, Through Debut Sales

	2SLS Estimate	Standard Error	N
Weekly Revenue	0.05348***	(0.01189)	279,815
Weekly Revenue for the Experienced	0.11187***	(0.03415)	74,526
Weekly Revenue for the Inexperienced	0.02907*	(0.01629)	161,245
Weekly Revenue for (Days Open > 365)	0.11314***	(0.03619)	108,656
Weekly Revenue for (Days Open ≤ 365)	0.03718**	(0.01792)	127,115

*p < 0.1; **p < 0.05; ***p < 0.01

Note: I instrument debut sales by weekly revenue in log terms to estimate a 2SLS linear probability model of the probability a product is reordered. Fixed effects for strain, the product, the product and the month, the dispensary, processor, month, week, all Federal holidays, and April 20th are included. Heteroskedasticity robust standard errors that are clustered at the dispensary level are reported in parentheses. The model is fit using data on every dispensary, then only dispensaries where the owner had previous experience, dispensaries where the owner had no previous experience, dispensaries in their second or third years, and finally dispensaries in their first years.

5 REFERENCES CITED

P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 1, New York, NY, USA, 2004. ACM. ISBN 1-58113-828-5.

P. Abbeel, A. Coates, M. Quigley, and A. Ng. An application of reinforcement learning to aerobatic helicopter flight. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. URL https://proceedings.neurips.cc/paper_files/paper/2006/file/98c39996bf1543e974747a2549b3107c-Paper.pdf.

C. Camerer, L. Babcock, G. Loewenstein, and R. Thaler. Labor Supply of New York City Cab Drivers: One Day At A time. Working Papers 960, California Institute of Technology, Division of the Humanities and Social Sciences, Apr. 1996. URL https://ideas.repec.org/p/clt/sswopa/960.html.

H. Farber. Is tomorrow another day? the labor supply of new york city cabdrivers. *Journal of Political Economy*, 113(1):46–82, 2005. ISSN 00223808, 1537534X. URL http://www.jstor.org/stable/10.1086/426040.

X. Gabaix. Chapter 4 - behavioral inattention. In B. D. Bernheim, S. DellaVigna, and D. Laibson, editors, *Handbook of Behavioral Economics - Foundations and Applications 2*, volume 2 of *Handbook of Behavioral Economics: Applications and Foundations 1*, pages 261–343. North-Holland, 2019. doi: https://doi.org/10.1016/bs.hesbe.2018.11.001. URL https://www.sciencedirect.com/science/article/pii/S2352239918300216.

Gartner. Gartner Press Release gartner identifies top trends shaping the future of data science and machine learning, 2023. URL https://www.gartner.com/en/newsroom/press-releases/2023-08-01-gartner-identifies-top-trends-shaping-future-of-data-science-and-machin

A. Goldfarb and M. Xiao. Transitory shocks, limited attention, and a firm's decision to exit. 2019. URL http://www-2.rotman.utoronto.ca/agoldfarb/limitedattention.pdf.

K. Haggag, B. McManus, and G. Paci. Learning by driving: Productivity improvements by new york city taxi drivers. *American Economic Journal: Applied Economics*, 9(1): 70–95, January 2017. doi: 10.1257/app.20150059. URL https://www.aeaweb.org/articles?id=10.1257/app.20150059.

V. Hotz and R. Miller. Conditional choice probabilities and the estimation of dynamic discrete choice models. *Review of Economic Studies*, 60:497–529, 07 1993. doi: 10.2307/2298122.

K. Hu. Chatgpt sets record for fastest-growing user base: analyst note. February 2023. URL https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/. Accessed: 2024-06-13.

O. Jackson. The supply side of discrimination: evidence from the labor supply of Boston taxi drivers. Working Papers 18-2, Federal Reserve Bank of Boston, June 2018. URL https://ideas.repec.org/p/fip/fedbwp/18-2.html.

D. Kahneman. *Thinking, fast and slow*. Farrar, Straus and Giroux, New York, 2011. ISBN 9780374275631 0374275637. URL https://www.amazon.de/Thinking-Fast-Slow-Daniel-Kahneman/dp/0374275637/ref=wl_it_dp_o_pdT1_nS_nC?ie=UTF8&colid=151193SNGKJT9&coliid=I3OCESLZCVDFL7.

B. Kim and J. Pineau. Socially adaptive path planning in human environments using inverse reinforcement learning. *International Journal of Social Robotics*, 8:51–66, 2016.

H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard. Socially compliant mobile robot navigation via inverse reinforcement learning. *The International Journal of Robotics Research*, 35, 01 2016. doi: 10.1177/0278364915619772.

U. Malmendier and T. Taylor. On the verges of overconfidence. *Journal of Economic Perspectives*, 29(4):3–8, November 2015. doi: 10.1257/jep.29.4.3. URL https://www.aeaweb.org/articles?id=10.1257/jep.29.4.3.

D. McFadden. Conditional logit analysis of qualitative choice behaviour. In P. Zarembka, editor, *Frontiers in Econometrics*, pages 105–142. Academic Press New York, New York, NY, USA, 1973.

G. Neu and C. Szepesvari. Apprenticeship learning using inverse reinforcement learning and gradient methods, 2012.

New York City Department of City Planning. Nyc census tracts for 2010 us census, water included, 2010. URL https://services5.arcgis.com/GfwWNkhOj9bNBqoJ/arcgis/rest/services/NYC_Census_Tracts_for_2010_US_Census_Water_Included/FeatureServer/0/query?where=1=1&outFields=*&outSR=4326&f=pgeojson. Accessed on January 10, 2024.

A. Y. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *in Proc. 17th International Conf. on Machine Learning*, pages 663–670. Morgan Kaufmann, 2000.

A. S. Pakes. Patents as Options: Some Estimates of the Value of Holding European Patent Stocks. *Econometrica*, 54(4):755–784, July 1986. URL https://ideas.repec.org/a/ecm/emetrp/v54y1986i4p755-84.html.

N. D. Ratliff, D. Silver, and J. A. Bagnell. Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots*, 27:25–53, 2009.

J. Rust. Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher. *Econometrica*, 55(5):999–1033, September 1987. URL https://ideas.repec.org/a/ecm/emetrp/v55y1987i5p999-1033.html.

J. Rust. Chapter 51 structural estimation of markov decision processes. volume 4 of *Handbook of Econometrics*, pages 3081–3143. Elsevier, 1994. doi: https://doi.org/10.1016/S1573-4412(05)80020-0. URL https://www.sciencedirect.com/science/article/pii/S1573441205800200.

N. Sanghvi, S. Usami, M. Sharma, J. Groeger, and K. Kitani. Inverse reinforcement learning with explicit policy estimates, 2021.

A. Tucker, A. Gleave, and S. Russell. Inverse reinforcement learning for video games. In *Proceedings of the Workshop on Deep Reinforcement Learning at NeurIPS*, 2018.

A. Tyson and E. Kikuchi. Growing public concern about the role of artificial intelligence in daily life. August 2023. URL https://www.pewresearch.org/short-reads/2023/08/28/growing-public-concern-about-the-role-of-artificial-intelligence-in-daily-life/. Accessed: 2024-06-13.

U.S. Census Bureau. American community survey 5-year estimates, 2013: Means of transportation to work, table b08301, 2013. URL https://www.census.gov/programs-surveys/acs/data.html. Accessed via tidycensus in R on May 14, 2024.

A. Vogel, D. Ramachandran, R. Gupta, and A. Raux. Improving hybrid vehicle fuel efficiency using inverse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 384–390, 2012.

C. Whong. FOILing NYC's Taxi Trip Data, 2014. URL https://chriswhong.com/open-data/foil_nyc_taxi/. Accessed: 2024-01-10.

World Economic Forum. *The Future of Jobs Report 2023*. April 2023. ISBN 978-2-940631-96-4. URL https://www.weforum.org/publications/the-future-of-jobs-report-2023/digest/. Accessed: 2024-06-13.

B. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. Bagnell, M. Hebert, A. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. pages 3931–3936, 12 2009. doi: 10.1109/IROS.2009.5354147.

B. D. Ziebart, J. A. Bagnell, and A. K. Dey. Modeling interaction via the principle of maximum causal entropy. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 1255–1262, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.