Uncovering Utility Functions from Observed Outcomes

Marta Grześkiewicz[1]

[1]University of Cambridge
mg2192@cam.ac.uk

Abstract

Determining consumer preferences and utility is a foundational challenge in economics. They are central in determining consumer behaviour through the utility-maximising consumer decision-making process. However, preferences and utilities are not observable and may not even be known to the individual making the choice; only the outcome is observed in the form of demand. Without the ability to observe the decision-making mechanism, demand estimation becomes a challenging task and current methods fall short due to lack of scalability or ability to identify causal effects. Estimating these effects is critical when considering changes in policy, such as pricing, the impact of taxes and subsidies, and the effect of a tariff. To address the shortcomings of existing methods, we combine revealed preference theory and inverse reinforcement learning to present a novel algorithm, Preference Extraction and Reward Learning (PEARL) which, to the best of our knowledge, is the only algorithm that can uncover a representation of the utility function that best rationalises observed consumer choice data given a specified functional form. We introduce a flexible utility function, the Input-Concave Neural Network which captures complex relationships across goods, including cross-price elasticities. Results show PEARL outperforms the benchmark on both noise-free and noisy synthetic data.

1

1 Introduction

Preferences and utility are central concepts in economics and play a key role in determining consumer behaviour and market outcomes. Preferences refer to an ordering over possible choices based on an individual's needs and desires and under certain conditions, these preferences can be represented by a utility function. In micro-economic theory, the consumer is utility maximising. Utility is an intrinsic 'reward' a consumer obtains when purchasing goods or services given their budget constraint. The outcome of this constrained optimisation problem is optimal consumption, or demand. Knowing the response of demand to changes in price and income is essential for estimating the effects of changes in policies which affect those variables. These responses can be used directly by firms to optimise their objectives, such as profit maximisation, sales maximisation, or indirectly by computing properties such as consumer surplus or elasticities. These properties are often used to determine economic policies such as the impact of taxes and subsidies, the effect of a tariff on prices, the valuation of non-market goods, and measuring the value of innovation [9, 13].

Demand estimation from observational data is a difficult problem. The estimation of interest is the response of demand to changes in price or other factors, *ceteris paribus*. The distinctive challenge appears when the presence of unobservable variables that affect both price and demand are acknowledged. The effects of such unobserved 'demand shocks' arise from changes in tastes or preferences and lead to the problem of price endogeneity, where the effect of change in price on demand cannot be identified as it is confounded by the unobserved shocks. To measure demand or its elasticities, these demand shocks must be held fixed. This condition rules out common approaches for estimation with endogeneity, even in the hypothetical case of a single-good economy [9]. This is additionally complicated by the fact that demand for any one good will, in general, depend on the prices, observed characteristics and demand shocks of that good and all related goods. Allowing endogeneity of additional characteristics leads to more restrictions on econometric models.

Demand is often modeled with a discrete choice model [36, 37], where a consumer selects exactly one of the options available to them and the option can be one good, or more generally, a basket of goods. Discrete choice demand is often represented with a random utility model, where the consumer's utility vector corresponds to each available good and an 'outside good' (choice of 'none of the above') is drawn from a joint distribution, conditional on their characteristics and goods available. These models are frequently formulated using a parametric random utility specification [37, 20, 38, 7, 8, 9]. Such a formulation may allow the correlation of prices with the error, thus allowing for endogeneity but still requires specification of IVs, the requirements for which are heavy as an IV has to be specified for every product and every combination of products, which rapidly faces scalability issues.

An alternative approach to uncovering preferences is using RP theory, a branch of micro-economics introduced by [42]. It uses observational data to obtain pairwise comparisons between chosen bundles of goods, which provides a ranking over all possible choices that can be represented by an ordinal utility function and is scale invariant. It also provides a method to verify that a dataset is consistent with the utility-maximising consumer with the condition that the ranking over choices inferred from the observations follows the axioms of revealed preference and there are no contradictions. For example, by the condition of transitivity, if a is strictly preferred to b and b is strictly preferred to c, it cannot be that c is strictly preferred to a. If the data is consistent, then a utility function which rationalises the observations can be inferred; with inconsistencies, the observed data can be adjusted by an efficiency index, such as Afriat's Index [2], to ensure it is consistent under the revealed preference axioms with adjustment.

The proposed algorithm, Preference Extraction and Reward Learning (PEARL), is an iterative process based on revealed preferences and works in two steps: firstly, with the current estimate of the parameters of the specified utility function, we find the consumption which minimises expenditure while keeping utility at least as good as what is obtained under the observed consumption (the dual of the utility maximisation problem). This is computed by projected gradient optimisation which iteratively computes the optimal value given an initial starting point. Secondly, we compute the difference between the expenditure under the estimated optimal consumption and that of the observed consumption. This is the loss that is to be minimised by updating the parameters of the utility function which generated the estimated optimal consumption with gradient descent. Intuitively, the algorithm should return the observed consumption as the optimal consumption under the observed budget constraint, given the fitted utility function.

The explicit treatment of prices as an input to the constraint of the utility maximisation problem means that, with changes in price, optimal consumption bundles that are consistent with the previous

observations and the resulting utility function can be computed, thus isolating the effect of price on demand. Any deviations from this optimal consumption bundle can be viewed as the demand 'shocks' and thus addressing the endogeneity problem found in demand estimation with regression. With subsequent observations, changing tastes and preferences can be shown by inconsistencies with the observed dataset under revealed preferences or by significant deviations from the estimated optimal output by the algorithm.

This approach is similar to the procedure in Inverse Reinforcement Learning (IRL) algorithms, where for the given estimate of the reward, the optimal action under a reward-maximising agent is found, and then the outcome of the optimal action with the observed outcomes is compared, often done by feature matching [40]. However, unlike IRL methods, PEARL is centered around the constrained optimisation problem of utility maximisation under a budget constraint, and given that the data generating process is that of the utility-maximiser, revealed preference theory provides foundations for the guarantees of the existence of a utility function which rationalises the observed data.

To address the requirement to specify a functional form of the utility function, we introduce multiple concave activation functions: *concave-tanh*, *concave-sigmoid* and *concave-log*. The architecture of this neural network, the Input-Concave Neural Network (ICNN), is based on [4]. ICNN retains flexibility of the function while satisfying the properties of a utility function: it is monotonically increasing and concave with respect to the input. It allows own-price and cross-price elasticities to vary with prices and incomes, which can be estimated numerically.

The performance of PEARL is shown on simulated data and real-world data. We investigate two settings: simulated data with noise-free observations and simulated data with noisy observations. In the first setting, we generate observations and show that PEARL can recover the true parameter values when the functional form of the utility function is known and we show that the ICNN is able to recover the utility function and predict demand and estimate elasticities with near-zero error. In the second setting, we perturb the generated observations with noise. We consider two types of noise: random noise on consumption and noise arising from endogeneity, both of which result in deviations from the optimal utility-maximising bundle. We compare the performance of PEARL to machine learning demand prediction methods, as presented in [6], and show that our methodology significantly out-performs the benchmark.

2   Related Work

RP was first introduced by [41] and further developed by [27, 1, 16]; see [47] and [45] for an overview. Testing of RP axioms in observational data is discussed in [43], and causes of violations in [45].

Demand estimation is related to policy analysis, for which the standard econometric identification method is instrumental variables [7, 25, 24, 5, 39, 21]. Specification of instrumental variables is necessary as they are sources of exogenous variation which is required to overcome the effect of demand shocks on endogenous variables of price and demand. To be identified, at least $2k$ instrumental varialbes are required for k goods, and often more are necessary to check for validity of instruments. PEARL overcomes these problems by implicitly holding the demand shocks across all goods fixed by keeping preferences constant under RP. When there are deviations from the utility-maximising optimal consumption, these are treated as demand shocks and de-coupled from the effect of price on demand.

Other commonly used methods in the economics literature are using the matching assumption [22]; the control function [23, 44, 10]; and panel data approaches, such as separability and difference-in-difference. For demand prediction without explicit treatment of endogeneity, linear regression is still a popular benchmark model in demand forecasting [46]. Machine learning methods are shown as alternatives to the econometric model in [11], neural networks in [18] and random forests have been applied in the context of load forecasting on the electricity market in [35]. Various methods for demand estimation for supermarket data are benchmarked in [6]. All these methods fall short because they do not allow for identification for demand due to the lack of explicit treatment of endogeneity. See [9] for a detailed discussion. PEARL addresses these shortcomings by explicitly de-coupling the effect of changes in price on demand and demand shocks through solving the consumer's utility-maximisation problem where price is an input and demand is an output, thus addressing endogeneity.

The structure of the proposed algorithm, PEARL, is similar to that found in the IRL literature; [40] use a reward function linear in features and feature matching to evaluate the fitted reward function against the observations, while [52] extend the method and fit a deep neural network as the reward function, while still evaluating with feature matching under the Maximum Entropy framework [53]. These methods lack theoretical foundations of choice and decision-making and often have to solve for e.g. maximum entropy

3

such that the reward function does not become degenerate and collapse to 0 everywhere. PEARL uses economic theory to underpin the mechanism of consumer choice, which informs the form of the utility function and avoids the problem of degeneration.

Finally, [33] use Afriat's piece-wise linear utility function [1] to solve the IRL problem in a probe and drone setting, where a drone/UAV or electromagnetic signal probes an 'adversary' cognitive radar system which then estimates the kinematic coordinates of the drone/UAV using a Bayesian tracker and then adapts its mode (waveform, aperture, revisit time) dynamically using feedback control based on sensing the kinematic state of the drone. They apply RP to determine whether the radar is 'rational' and use Afriat's piece-wise linear utility to find a utility function to predict the future behaviour of the cognitive radar. They use pairwise comparisons from RP under a utility-maximising and rational agent in a non-linear constraint setting (first presented by [19]) to fit the piece-wise linear utility function given observations. In contrast, PEARL does not rely on constructing the ordered bundles and finding Afriat's numbers that result in an order-preserving piece-wise linear function. Instead, it introduces a model-agnostic algorithm that can fit any utility function given observations from a single time period, which either are or can be adjusted to be consistent under RP. This provides more accurate representations of the utility due to the input-concave nature of the function.

3   Background

Consider an agent choosing between two bundles of k goods denoted by the $k \times 1$ vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, $\mathbf{x}_i, \mathbf{x}_j \in X \subseteq \mathbb{R}^k_+$, which denote the quantities that the agent chooses over the k goods. If they choose $\mathbf{x}_i$ over $\mathbf{x}_j$, we say that $\mathbf{x}_i$ is weakly preferred to $\mathbf{x}_j$ and this is denoted as $\mathbf{x}_i \succeq \mathbf{x}_j$. In other words, $\mathbf{x}_i$ is at least as good as $\mathbf{x}_j$. Strict preference is denoted by $\mathbf{x}_i \succ \mathbf{x}_j$, while indifference is denoted by $\mathbf{x}_i \sim \mathbf{x}_j$. Following standard theory (see, for example, [49]), the following three assumptions about preferences are made: firstly, they are complete, meaning that any two bundles can be compared; secondly, they are reflexive, meaning that any bundle is at least as good as itself ($\mathbf{x}_i \succeq \mathbf{x}_i$); and thirdly, they are transitive, meaning that if $\mathbf{x}_i \succeq \mathbf{x}_j$ and $\mathbf{x}_j \succeq \mathbf{x}_k$, then $\mathbf{x}_i \succeq \mathbf{x}_k$. While the first two assumptions are straightforward, transitivity is a strong assumption as it ensures that for any sets of options that are to be compared, there will be at least one best choice (the agent may be indifferent between multiple best choices). Further technical assumptions for the preferences to be 'well-behaved' are made, such as monotonicity, convexity and local non-satiation. Note that convexity in preferences implies that averages (combinations of many goods) are preferred to extremes (consuming lots of one good).

Given such a preference ordering, by the representation theorem in [15], this can be described by a utility function indexed by d-dimensional parameter vector $\boldsymbol{\theta}$, denoted by $U_{\boldsymbol{\theta}} : \mathbb{R}^k_+ \to \mathbb{R}_+$, where if $\mathbf{x}_i \succeq \mathbf{x}_j$, then $U_{\boldsymbol{\theta}}(\mathbf{x}_i) \geq U_{\boldsymbol{\theta}}(\mathbf{x}_j)$.[1] This utility function has to be monotonically increasing, (quasi-)concave. The agent is then faced with the following objective

$$\max_{\mathbf{x}} U_{\boldsymbol{\theta}}(\mathbf{x}) \text{ s.t. } \mathbf{p}_i^T \mathbf{x} \leq m_i \tag{1}$$

where the state is defined as $\mathbf{s}_i = (\mathbf{p}_i, m_i)$, where $\mathbf{p}_i \in \mathbb{R}^k_+$ denotes the corresponding $k \times 1$ vector of prices. This corresponds to the 'budget constraint', the total amount m_i an agent can afford, given the current prices $\mathbf{p}_i$.

Suppose an agent is acting optimally under this objective and the observations $D = \{\mathbf{x}_i, \mathbf{p}_i, m_i\}_{i=1}^N$ are obtained. Neither the utility function nor its values are observed as they are intrinsic to the agent. The feasibility region, $O(\mathbf{s}_i)$, is described as the alternative consumption choices that satisfy the condition $\mathbf{x} \in O(\mathbf{s}_i) \in X$ if $\mathbf{p}_i^T \mathbf{x} \leq m_i$. If an agent is observed to be acting optimally and always chooses what they most prefer, then it must be the case that $\mathbf{x}_i \succeq \mathbf{x}_j, \forall \mathbf{x}_j \in O(\mathbf{s}_i)$.

Under the linear constraint, for every observation, $\mathbf{x}_i$ is *strictly directly revealed preferred* to $\mathbf{x}_j$, denoted $\mathbf{x}_i P^D \mathbf{x}_j$ if $\mathbf{p}_i^T \mathbf{x}_i > \mathbf{p}_i^T \mathbf{x}_j$. Then, the transitive closure is computed and $\mathbf{x}_i$ is *revealed preferred* to $\mathbf{x}_j$, denoted $\mathbf{x}_i R \mathbf{x}_j$ if $\mathbf{p}_i^T \mathbf{x}_i \geq \mathbf{p}_i^T \mathbf{x}_k, \mathbf{p}_k^T \mathbf{x}_k \geq \mathbf{p}_k^T \mathbf{x}_l, \ldots, \mathbf{p}_m^T \mathbf{x}_m \geq \mathbf{p}_m^T \mathbf{x}_j$ for some sequence of observations $(\mathbf{x}_i, \mathbf{x}_j, \ldots, \mathbf{x}_m)$. Given these revealed preference relations, a condition for consistency with revealed preferences called the Generalised Axiom of Revealed Preference (GARP) [1] is defined:

Definition 1 (GARP). If $\mathbf{x}_i R \mathbf{x}_j$, then $\mathbf{x}_j P^D \mathbf{x}_i$ cannot be true and $\mathbf{x}_i R \mathbf{x}_j$ implies $\mathbf{p}_j^T \mathbf{x}_i \leq \mathbf{p}_j^T \mathbf{x}_j$.

Moreover, under Afriat's Theorem [1], if the data is consistent under GARP, then there is an increasing, monotonic, continuous and concave utility function that rationalises this data. GARP-consistency

---

[1]In the lottery space, for some probability attributed to the utility outcome, this is represented by the utility theorem in [50]: $\mathbb{E}[U(\mathbf{x}_i)] \geq \mathbb{E}[U(\mathbf{x}_j)]$. Here, only the deterministic case is considered.

is a testable property of the data set and we can check whether a data set is consistent under GARP with algorithms of transitive closure, such as that presented in [51] which has computational complexity $O(N^3)$, combined with an algorithm that checks for cycles in a directed graph, such as Depth-First-Search [14]. [33] introduce a statistical test to check whether a data set is GARP-consistent even in the presence of noise from measurement error. [45] review various reasons for the violation of GARP and their remedies.

In the case when the data set is not GARP-consistent, 'goodness-of-fit' indices can be obtained, for example Afriat's index [2], Varian's index [48] and the Houtman-Maks index [28]; the computational complexity of these indicies is discussed in [43]. Of these, Afriat's index is the simplest, where a value $0 \leq \varepsilon \leq 1$ is introduced for all observations. It relaxes the RP relations with the following adjustment:

$$
\begin{aligned}
&\text{if } \varepsilon \mathbf{p}_i^T \mathbf{x}_i \geq \mathbf{p}_i^T \mathbf{x}_j, \text{ then } \mathbf{x}_i R^D(\varepsilon) \mathbf{x}_j \\
&\text{if } \varepsilon \mathbf{p}_i^T \mathbf{x}_i > \mathbf{p}_i^T \mathbf{x}_j, \text{ then } \mathbf{x}_i P^D(\varepsilon) \mathbf{x}_j
\end{aligned}
\tag{2}
$$

where $R(\varepsilon)$ and $P(\varepsilon)$ represent the transitive closures of $R^D(\varepsilon)$ and $P^D(\varepsilon)$, respectively.

Definition 2. For a given ε, with $0 \leq \varepsilon \leq 1$, a dataset $D = \{\mathbf{p}_i, \mathbf{x}_i\}_{i=1}^N$ satisfies GARP(ε) if and only if, for each pair of distinct bundles, $\mathbf{x}_i, \mathbf{x}_j, i, j = 1, \ldots, N; i \neq j$, if $\mathbf{x}_i R(\varepsilon) \mathbf{x}_j$, then it cannot be that $\mathbf{x}_j P^D(\varepsilon) \mathbf{x}_i$.

4 Methodology

In this section, we present the PEARL algorithm. Under RP theory, the observed data has to be GARP-consistent for a concave, increasing, monotonic utility function to exists. As such, we first check for this consistency, and if it is not consistent, we adjust the data. We then present the algorithm for recovering the utility function, given a specified functional form. Finally, we present a numerical method for maximising the recovered utility given state variables.

4.1 GARP-consistency

We verify that the set of observations D is consistent with optimality conditions under RP theory. The $N \times N$ binary relation matrix $\mathbf{R}$ is constructed with elements $i, j = 1, \ldots, N$,

$$
\mathbf{R}[i, j] = \begin{cases} 1, & \text{if } \mathbf{p}_i^T \mathbf{x}_i \geq \mathbf{p}_i^T \mathbf{x}_j, \text{ and } i \neq j \\ 0, & \text{otherwise.} \end{cases}
\tag{3}
$$

The matrix $\mathbf{R}$ defines a directed graph such that every node is an observation and every edge represents whether the observation is preferred to the other. GARP inconsistencies are found by constructing the transitive closure graph [51] which is then checked for cycles using e.g. the Depth-First-Search algorithm; see, for example, [14]. If inconsistencies are present, then the data is adjusted by Afriat's efficiency index, $0 < \varepsilon \leq 1$, which is defined as the maximum value of ε such that the adjusted dataset $D' = \{\varepsilon \mathbf{x}_i, \mathbf{p}_i, m_i\}_{i=1}^N$ has no inconsistencies. This ensures the binary relation under transitive closure is acyclic. The value of ε is approximated by binary search.

4.2 Recovering Utility

With GARP-consistent observations and under Afriat's theorem, there exists an increasing, monotonic, concave utility function that rationalises the data. We introduce our algorithm, Preference Extraction and Reward Learning (PEARL), which recovers a representation of the underlying utility function given a set of observations and a specified functional form which satisfies the properties of a utility function.

Since we observe only the outcome of consumer choice in terms of consumption and expenditure, $\mathbf{x}_i$ and m_i, respectively, PEARL compares the observed values m_i to computed optimal m-values, $\hat{m}(\mathbf{p}_i, u_i)$ by a loss function. The m-values are solutions to the dual problem of the utility maximisation objective, also known as the money metric utility [42]. It is defined as

$$
\hat{m}(\mathbf{p}_i, u_i) = \min_{\mathbf{x}} \mathbf{p}_i^T \mathbf{x} \text{ s.t. } U_\theta(\mathbf{x}) \geq u_i
\tag{4}
$$

where $\hat{m}(\mathbf{p}_i, u_i)$ is the minimum cost payable in order to achieve the utility level u_i for some parameterised utility function U_θ. Here and in the following sections, the utility function is indexed by the unknown

5

parameter vector $\boldsymbol{\theta}$, which is object of interest in recovering the utility function. These values are compared by the L1 loss, given by

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{N} L_i(\boldsymbol{\theta}) \text{ where } L_i(\boldsymbol{\theta}) = |(\hat{m}(\mathbf{p}_i, u_i) - m_i)| \tag{5}$$

PEARL estimates the parameters $\boldsymbol{\theta}$ in two iterative stages. First, for every observation $i = 1, \ldots, N$ we find the action $\hat{\mathbf{h}}_i$ which, under the current utility function $U_{\hat{\theta}_t}$, is at least as good as the observed $\mathbf{x}_i$ and which minimises expenditure, given by

$$\hat{\mathbf{h}}_i = \arg \min_{\mathbf{x}} \mathbf{p}_i^T \mathbf{x} \text{ s.t. } U_{\hat{\boldsymbol{\theta}}}(\mathbf{x}) \geq u_i. \tag{6}$$

where the result $\hat{\mathbf{h}}_i$ corresponds to the optimal (Hicksian) demand (see, for example, [49]) where the utility level is at least u_i. The second stage uses these values to compute the loss and its gradients, and updates the parameters with gradient descent.

We now present the algorithm in more detail. The first stage computes the corresponding $\hat{\mathbf{h}}_i$ for every observation $i = 1, \ldots, N$ using a numerical method with projections. The method of projected gradients is used as with the constraint set of equation 6 defined to be $C_i = \{\mathbf{x} : U_{\hat{\boldsymbol{\theta}}}(\mathbf{x}) \geq u_i\}$, it is known to be a convex set because $U_{\hat{\boldsymbol{\theta}}}$ is by definition concave in $\mathbf{x}$. The objective to find the money metric utility can be written, for $i = 1, \ldots, N$, as $\hat{\mathbf{h}}_i = \arg \min_{\mathbf{x} \in C_i} \mathbf{p}_i^T \mathbf{x}$. The projected gradient calculations are then, with learning rate α, firstly to perform a gradient descent step

$$\hat{\mathbf{h}}_{i,t+\frac{1}{2}} = \hat{\mathbf{h}}_{i,t} + \alpha \frac{\mathbf{p}_i}{|\mathbf{p}_i|} \tag{7}$$

where $\hat{\mathbf{h}}_{i,t+\frac{1}{2}}$ denotes the intermediate step between $\hat{\mathbf{h}}_{i,t}$ and $\hat{\mathbf{h}}_{i,t+1}$. Then we compute the projection back onto the constraint set by finding the minimum distance from the intermediate point onto the constraint set

$$\hat{\mathbf{h}}_{i,t+1} = \arg \min_{\mathbf{y} \in C_i} ||\mathbf{y} - \hat{\mathbf{h}}_{i,t+\frac{1}{2}}|| \tag{8}$$

Setting up equation 6 as a Lagrangian, by the Karush-Kuhn-Tucker (KKT) conditions [29, 34], the constraint will bind such that $U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{x}}) = u_i$ and

$$\frac{\mathbf{p}_i}{|\mathbf{p}_i|} = \frac{\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_{i,t+\frac{1}{2}})/\partial \hat{\mathbf{h}}_{i,t+\frac{1}{2}}}{|\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_{i,t+\frac{1}{2}})/\partial \hat{\mathbf{h}}_{i,t+\frac{1}{2}}|} \tag{9}$$

As such, we take $\mathbf{y}$ as the point after a step is taken in the direction of normalised gradient of the utility function with respect to $\hat{\mathbf{h}}_{i,t+\frac{1}{2}}$ by the amount $U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{x}}) - u_i$, as an approximation of the shortest distance for small enough α. Hence,

$$\mathbf{y} = \hat{\mathbf{h}}_{i,t+\frac{1}{2}} + (u_i - U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_{i,t+\frac{1}{2}})) \frac{\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_{i,t+\frac{1}{2}})/\partial \hat{\mathbf{h}}_{i,t+\frac{1}{2}}}{|\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_{i,t+\frac{1}{2}})/\partial \hat{\mathbf{h}}_{i,t+\frac{1}{2}}|} \tag{10}$$

and so

$$\hat{\mathbf{h}}_{i,t+1} = \hat{\mathbf{h}}_{i,t} + \alpha \frac{\mathbf{p}_i}{|\mathbf{p}_i|} + (u_i - U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_{i,t+\frac{1}{2}})) \frac{\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_{i,t+\frac{1}{2}})/\partial \hat{\mathbf{h}}_{i,t+\frac{1}{2}}}{|\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_{i,t+\frac{1}{2}})/\partial \hat{\mathbf{h}}_{i,t+\frac{1}{2}}|} \tag{11}$$

This is summarised in Algorithm 1.

6

Algorithm 1: Computing Money Metric Utility

Input: $\mathbf{p}_i, u_i, i = 1, \ldots, N, U_{\hat{\boldsymbol{\theta}}}$, learning rate α, initial values $\hat{\mathbf{x}}_{i,0}$, number of iterations T_m.

Output: $\hat{\mathbf{h}}_i, \hat{m}_i, i = 1, \ldots, N.$

for $t = 0, \ldots, T_m - 1$ **do**

Compute intermediate step

$$\hat{\mathbf{h}}_{i,t+\frac{1}{2}} = \hat{\mathbf{h}}_{i,t} + \alpha \frac{\mathbf{p}_i}{|\mathbf{p}_i|} \tag{12}$$

Compute the projected gradient and update

$$\hat{\mathbf{h}}_{i,t+1} = \hat{\mathbf{h}}_{i,t} + \alpha \frac{\mathbf{p}_i}{|\mathbf{p}_i|} + (u_i - U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_{i,t+\frac{1}{2}})) \frac{\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_{i,t+\frac{1}{2}})/\partial \hat{\mathbf{h}}_{i,t+\frac{1}{2}}}{|\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_{i,t+\frac{1}{2}})/\partial \hat{\mathbf{h}}_{i,t+\frac{1}{2}}|}$$

end for

Set $\hat{\mathbf{h}}_i = \hat{\mathbf{h}}_{i,T_m}$; $\hat{m}_i = \mathbf{p}_i^T \hat{\mathbf{h}}_i.$

Given these estimates of $\hat{m}_i$ and $\hat{\mathbf{h}}_i$, the loss of how far away the cardinal value of the m-value is from the estimated and the observed is computed, given the observed prices $\mathbf{p}_i$. The parameters of the utility function are updated by gradient descent, with the gradient computation given below. Notice that with an assumed strictly concave utility function, the values $\hat{m}_i$ and $\hat{\mathbf{h}}_i$ are uniquely determined. The L1 loss is given by equation 5 and the parameters of the utility function, $\boldsymbol{\theta}$, are optimised by gradient descent. By the chain rule,

$$\frac{\partial L_i(\hat{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} = \left(\frac{\partial L_i(\boldsymbol{\theta})}{\partial \hat{m}_i} \right) \left(\frac{\partial \hat{m}_i}{\partial \hat{\mathbf{h}}_i} \right)^T \left(\frac{\partial \hat{\mathbf{h}}_i}{\partial \boldsymbol{\theta}} \right) \tag{13}$$

where $\frac{\partial \hat{\mathbf{h}}_i}{\partial \boldsymbol{\theta}}$ denotes the $k \times d$ matrix with elements as derivatives of $\hat{\mathbf{h}}_i$ with respect to the elements of $\boldsymbol{\theta}$. With the abuse of notation, the derivative of the loss function given in equation 5 is given by

$$\frac{\partial L_i(\boldsymbol{\theta})}{\partial \hat{m}_i} = \frac{|(\hat{m}_i - m_i)|}{(\hat{m}_i - m_i)} \tag{14}$$

for $\hat{m}_i \neq m_i$, and we set the value at 0 otherwise. Additionally, from equation 4,

$$\left(\frac{\partial m_i}{\partial \hat{\mathbf{h}}_i} \right)^T \left(\frac{\partial \hat{\mathbf{h}}_i}{\partial \boldsymbol{\theta}} \right) = \mathbf{p}_i \frac{\partial \hat{\mathbf{h}}_i}{\partial \boldsymbol{\theta}}. \tag{15}$$

To find an expression for $\frac{\partial \hat{\mathbf{h}}_i}{\partial \hat{\boldsymbol{\theta}}}$, the Lagrangian is constructed for the money-metric utility for the PEARL algorithm, denoted as $\mathcal{L}$, as

$$\mathcal{L}(\hat{\mathbf{h}}_i, \hat{\boldsymbol{\theta}}; \mathbf{p}_i, \mathbf{x}_i) = \mathbf{p}_i^T \hat{\mathbf{h}}_i - \lambda(U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_i) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{x}_i)) \tag{16}$$

where λ is the Lagrange multiplier. The first order conditions are given by

$$\frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{\theta}}} = \mathbf{p}_i \frac{\partial \hat{\mathbf{h}}_i}{\partial \hat{\boldsymbol{\theta}}} - \lambda \left(\frac{\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_i)}{\partial \hat{\boldsymbol{\theta}}} - \frac{\partial U_{\hat{\boldsymbol{\theta}}}(\mathbf{x}_i)}{\partial \hat{\boldsymbol{\theta}}} \right) - \frac{\partial \lambda}{\partial \hat{\boldsymbol{\theta}}} \left(U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_i) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{x}_i) \right) \tag{17}$$

$$\frac{\partial \mathcal{L}}{\partial \hat{h}_{i,j}} = p_{i,j} - \lambda \frac{\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_i)}{\partial \hat{h}_{i,j}} \tag{18}$$

where j refers to the jth element of $\hat{\mathbf{h}}_i$ and $\mathbf{p}_i$, $j = 1, \ldots, k$. At optimum, from Equation 18, the set of expressions is obtained for $j = 1, \ldots, k$,

$$\hat{\lambda} = \frac{p_{i,j}}{\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_i)/\partial \hat{h}_{i,j}} \tag{19}$$

which should be equal at optimum for every j. As an approximation during training it is set as

$$\hat{\lambda} = k^{-1} \mathbf{p}_i^T \cdot \left(\frac{\partial U_{\hat{\boldsymbol{\theta}}}}{\partial \hat{\mathbf{h}}_i} \right)^{-1} \tag{20}$$

7

Also from Equation 17, at optimum, since $\hat{\mathbf{x}}_i$ is the argument which maximises the utility maximisation objective for $i = 1, \ldots, N$ observations, $(U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{x}}_i) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{x}_i)) = 0$ is satisfied. So,

$$\mathbf{p}\frac{\partial \hat{\mathbf{h}}_i}{\partial \hat{\boldsymbol{\theta}}} = \hat{\lambda}\left(\frac{\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_i)}{\partial \hat{\boldsymbol{\theta}}} - \frac{\partial U_{\hat{\boldsymbol{\theta}}}(\mathbf{x})}{\partial \hat{\boldsymbol{\theta}}}\right). \tag{21}$$

Therefore, for $\hat{m}_i \neq m_i$ and 0 otherwise,

$$\frac{\partial L_i}{\partial \hat{\boldsymbol{\theta}}} = \frac{|(\hat{m}_i - m_i)|}{(\hat{m}_i - m_i)}k^{-1}\mathbf{p}_i^T \cdot \left(\frac{\partial U_{\hat{\boldsymbol{\theta}}}}{\partial \hat{\mathbf{h}}_i}\right)^{-1}\left(\frac{\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{h}}_i)}{\partial \hat{\boldsymbol{\theta}}} - \frac{\partial U_{\hat{\boldsymbol{\theta}}}(\mathbf{x}_i)}{\partial \hat{\boldsymbol{\theta}}}\right) \tag{22}$$

The methodology is summarised by Algorithm 2.

Algorithm 2: PEARL

Input: $D = \{(\mathbf{x}_i, \mathbf{p}_i, m_i)\}_{i=1}^N$, initial parameters $\hat{\boldsymbol{\theta}}_0$, number of iterations T
Output: Utility function $U_{\hat{\boldsymbol{\theta}}_t}(\cdot)$
Construct revealed preference graph by binary preference relation and transitive closure and
  check for GARP-consistency (acyclicality);
for $t = 1, \ldots, T$ **do**
$\quad$ Set $u_i = U_{\hat{\boldsymbol{\theta}}_t}(\mathbf{x}_i)$, for $i = 1, \ldots, N$;
$\quad$ Compute $\hat{\mathbf{h}}_i$, for $i = 1, \ldots, N$ by Algorithm 1 and set $\hat{m}_i = \mathbf{p}_i^T\hat{\mathbf{h}}_i$;
$\quad$ Compute loss $L(\hat{\boldsymbol{\theta}}_t) = \sum_{i=1}^N L_i(\hat{\boldsymbol{\theta}}_t)$. ;
$\quad$ Update parameters $\hat{\boldsymbol{\theta}}_t$ by gradient descent, where for $\hat{m}_i \neq m_i$ and 0 otherwise,

$$\frac{\partial L_i}{\partial \hat{\boldsymbol{\theta}}_t} = \frac{|(\hat{m}_i - m_i)|}{(\hat{m}_i - m_i)}k^{-1}\mathbf{p}_i^T \cdot \left(\frac{\partial U_{\hat{\boldsymbol{\theta}}_t}}{\partial \hat{\mathbf{h}}_i}\right)^{-1}\left(\frac{\partial U_{\hat{\boldsymbol{\theta}}_t}(\hat{\mathbf{h}}_i)}{\partial \hat{\boldsymbol{\theta}}_t} - \frac{\partial U_{\hat{\boldsymbol{\theta}}_t}(\mathbf{x}_i)}{\partial \hat{\boldsymbol{\theta}}_t}\right)$$

end for
Set $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}_T$;

4.3 Maximising Utility

Given a representation of the utility function, $U_{\hat{\boldsymbol{\theta}}}$, obtained by PEARL, the counterfactual prediction for $\hat{\mathbf{x}}$ under some given environmental variables $(\mathbf{p}, m)$ is obtained by solving the consumer's utility-maximising objective ($\hat{\mathbf{x}}^M = \arg\max_{\mathbf{x}} U_{\hat{\boldsymbol{\theta}}}(\mathbf{x})$ s.t. $\mathbf{p}^T\mathbf{x} \leq m$), which provides an estimate of (Marshallian) demand (see, for example, [49]). We present a method to compute this numerically by iteration. An initial point is chosen and projected onto the feasibility set, which, since the constraint is linear, is a simple scaling such that the constraint holds with equality $\hat{\mathbf{z}}_{i,t} = \hat{\mathbf{x}}_i^M \frac{m_i}{\mathbf{p}_i \cdot \hat{\mathbf{x}}_i^M}$. The point is then moved along the constraint by the update rule

$$\hat{\mathbf{z}}_{i,t+1} = \hat{\mathbf{z}}_{i,t} + \frac{\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{z}}_{i,t})/\partial \hat{\mathbf{z}}_{i,t}}{|\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{z}}_{i,t})/\partial \hat{\mathbf{z}}_{i,t}|} - |\mathbf{p}|^T \cdot |\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{z}}_{i,t})/\partial \hat{\mathbf{z}}_{i,t}||\mathbf{p}| \tag{23}$$

Notice that at optimum, under the KKT conditions, $\mathbf{p}$ and $\frac{\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{z}}_{i,t})}{\partial \hat{\mathbf{z}}_{i,t}}$ will be equivalent. The procedure is summarised in Algorithm 3.

4.3.1 Adjusting for GARP-inconsistencies

When the dataset is adjusted with ε to achieve GARP-consistency, the inverse of the adjustment is applied to the outcomes of the utility maximisation problem. To achieve this, the epsilon is included in the utility condition when minimising the expenditure, $u_i = U_{\hat{\boldsymbol{\theta}}}(\varepsilon\mathbf{x}_i)$ and in the transformation of the optimal value under the current utility function

$$\hat{\mathbf{x}}_i = \frac{1}{\varepsilon}\arg\min_{\mathbf{x}} \mathbf{p}_i^T\mathbf{x} \text{ s.t. } U_{\hat{\boldsymbol{\theta}}}(\mathbf{x}) \geq u_i \tag{24}$$

Algorithm 3: Maximise Utility

Input: $U_{\hat{\boldsymbol{\theta}}}, (\mathbf{p}_i, m_i)_{i=1}^N$, number of iterations T
Output: $\{\hat{\mathbf{x}}\}_{i=1}^N, \{\hat{m}_i\}_{i=1}^N$
Initialise $\hat{\mathbf{x}}_i^M$, for $i = 1, \ldots, N$;
Project $\hat{\mathbf{x}}_i^M$ onto the constraint set by setting $\mathbf{z}_i = \hat{\mathbf{x}}_i^M \frac{m_i}{\mathbf{p}_i \cdot \hat{\mathbf{x}}_i^M}$

for $t = 1, \ldots, T$ **do**
  $\hat{\mathbf{z}}_{i,t+\frac{1}{2}} = \hat{\mathbf{z}}_{i,t} + \frac{\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{z}}_{i,t})/\partial \hat{\mathbf{z}}_{i,t}}{|\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{z}}_{i,t})/\partial \hat{\mathbf{z}}_{i,t}|} - |\mathbf{p}_i|^T \cdot |\partial U_{\hat{\boldsymbol{\theta}}}(\hat{\mathbf{z}}_{i,t})/\partial \hat{\mathbf{z}}_{i,t}||\mathbf{p}_i|$
  Project $\hat{\mathbf{z}}_{i,t+\frac{1}{2}}$ onto the constraint set by setting $\hat{\mathbf{z}}_{i,t+1} = \hat{\mathbf{z}}_{i,t+\frac{1}{2}} \frac{m_i}{\mathbf{p}_i \cdot \hat{\mathbf{z}}_{i,t+\frac{1}{2}}}$
end for
Set $\hat{\mathbf{x}}_i^M = \hat{\mathbf{z}}_{i,T}$

4.3.2 Numerical Estimation of Price Elasticities

Elasticities are defined as the relative change in demand due to a change in price (see, for example [49]). The expression in equation 25, based on [31], gives a numerical approximation for (uncompensated) price elasticity of demand for good i with respect to the change in price of good j, when the partial derivative cannot be computed directly.

$$e_{i,j} = \frac{\partial x^i}{\partial p^j} \approx \frac{p^j}{x^i} \frac{\Delta x^i}{\Delta p^j} \tag{25}$$

where when $i = j$ we compute own-price elasticity, and $i \neq j$ relating to cross-price for $i, j = 1, \ldots, k$. The price vector is perturbed around the original vector $\mathbf{p}$ in both plus and minus, per element j. n Notice that these elasticities are not partitioned; the Hicks-Slutsky partition [17] can also be computed with the corresponding constrained optimisation problems to compute both compensated and uncompensated price elasticity of demand.

4.4 Utility Functions

To illustrate the performance of the method, two functional forms of utility functions are considered: Cobb-Douglas (CD), and Input-Concave Neural Network (ICNN). The first is used in both simulating data and fitting, whilst the last, the ICNN is considered for only fitting the utility function using PEARL and for computing elasticties.

4.4.1 Cobb-Douglas

The Cobb-Douglas utility function [12] is a widely used mathematical representation to describe consumer utility with constant elasticities of substitution. The k-variable case is described as

$$U_{\boldsymbol{\theta}}^{CD}(\mathbf{x}) = \prod_j^k x_j^{\theta_j} \text{ s.t. } \sum_j^k \theta_j = 1 \tag{26}$$

where x_j refers to the j-th good (element) of a bundle of goods $\mathbf{x}$, $j = 1, \ldots, k$. It is one of the simplest representations of utility and we use this as the basic case to show the effectiveness of PEARL at recovering utility functions.

4.4.2 Input-Concave Neural Network

The performance of the algorithm is investigated by assuming that the utility function is a complex parametric function, expressed by a neural network. As per the requirements of the algorithm, the function has to be input-concave and increasing, i.e. the derivative with respect to the input is positive and monotonically decreasing, and the second derivative with respect to the input is negative.

The implementation of the Input-Concave Neural Network (ICNN) is based on the Input-Convex Neural Network, presented in [4]. A neural network is defined over inputs $\{\mathbf{x}\}_{i=1}^N$ for $\ell = 0, \ldots, L$ fully connected (dense) layers. The output of each layer is

$$z_{\ell+1}(\mathbf{x}_i) = h_\ell \left(W_\ell^{(z)} z_\ell + W_\ell^{(\mathbf{x})} \mathbf{x}_i + b_\ell \right); U_\theta^{ICNN}(\mathbf{x}_i) = z_L(\mathbf{x}_i) \tag{27}$$

where, with z_0, $W_0^{(z)} = 0$, z_ℓ denotes the layer activation functions, h_ℓ are the non-linear activation functions and $\boldsymbol{\theta} = \{W_{0:\ell-1}^{(\mathbf{x})}, W_{1:\ell-1}^{(z)}, b_{0:\ell-1}\}$ are the parameters. The function $U_{\boldsymbol{\theta}}^{ICNN}(\mathbf{x}_i)$ is concave in $\mathbf{x}_i$ provided that all $W_{1:L-1}^{(z)}$ are non-negative and all functions h are concave and non-decreasing, following the convex proposition of the same nature presented in [4]. The proof follows from the fact that non-negative sums of concave functions are also concave and that the composition of two concave functions are also concave. The layer activation function weights are enforced to be non-negative by constructing an activation function to be concave. Multiple candidates for concave activation functions are considered:

Concave-tanh Adapting the hyperbolic tangent, we obtain the 'concave-tanh' activation function, given by

$$h(\mathbf{x}) = \begin{cases} \tanh(\mathbf{x}), & \mathbf{x} \geq \mathbf{0} \\ \mathbf{x}, & \mathbf{x} < \mathbf{0}. \end{cases} \tag{28}$$

Concave-sigmoid Adapting the sigmoid function, we obtain the 'concave-sigmoid' activation function, given by

$$h(\mathbf{x}) = \begin{cases} \frac{1}{1+e^{-\mathbf{x}}}, & \mathbf{x} \geq \mathbf{0} \\ \frac{1}{4}\mathbf{x} + \frac{1}{2}, & \mathbf{x} < \mathbf{0}. \end{cases} \tag{29}$$

Concave-log Adapting the natural log, we obtain the 'concave-log' activation function, where $\delta > 0$ is an offset necessary to avoid the asymptote at $\mathbf{x} = 0$, given by

$$h(\mathbf{x}) = \begin{cases} \ln(\mathbf{x} + \delta), & \mathbf{x} > \mathbf{0} \\ \frac{1}{\delta}\mathbf{x} + \ln\delta, & \mathbf{x} \leq \mathbf{0}. \end{cases} \tag{30}$$

4.5 Pre-training on the Afriat numbers

We can pre-train the network as a supervised learning problem given utility values obtained from Afriat's piecewise-linear utility function. Theorem says that if the data is GARP-consistent, then we can get a piece-wise linear function, such that there exist numbers $U_i, \lambda_i > 0$ such that $U_i \leq U_j + \lambda_j p_j(x_i - x_j)$ for $i, j = 1, \ldots, n$. These numbers are found in the iterative procedures detailed in [47].

4.6 Implementation Details

The training and model parameters have been optimised across multiple options and the resulting values are presented. The ICNN model is trained using the Adam optimiser [30] with an epsilon value of $1.0e-5$ and weight decay of $5.0e-4$. The initial learning rate is set at $1.0e-3$, decreasing over time with a Exponential Decay scheduler with a final learning rate of $1.0e-8$ and a decay rate of . We set the batch size at 128. We train for $1,000$ steps for Algorithm 1 at every iteration with a learning rate of $1.0e-3$. When the utility function is set to CD, it is trained for $1,000$ epochs. Unless otherwise specified, when the utility function is set as ICNN, we construct the network with 3 layers and k hidden units in each layer, where k is the number of dimensions of the input $\mathbf{x}$. This ensures that there is no dimensionality reduction whilst the number of parameters in the network is kept as small as possible. The architecture of the network is as described in section 4.4.2, with the addition of batch normalisation before the first dense layer and dropout after the first and before the last dense layers, neither of which affect the input-concavity of the network. Unless otherwise stated, the activation function is taken to be *concave-log* with $\delta = 0.01$. The model is trained for $10,000$ epochs due to the larger number of parameters.

5 Simulations

In this section, we illustrate the performance of PEARL on simulated data in two cases: firstly, with Cobb-Douglas as ground-truth utility with no noise; secondly, with Cobb-Douglas as ground-truth utility but the observations are affected by noise. We compare prediction error to machine learning regression methods benchmarked in [6].

5.1 Simulations without Noise

To asses the performance of PEARL on observations without noise, the observations are simulated as consumption choices made by a consumer maximising their utility function given a budget constraint. Prices and incomes are uniformly sampled in the ranges $[1, 10]$ and $[50, 150]$, respectively, and we assume that the ground truth utility function is of Cobb-Douglas form, as given in equation 26, with k parameters. The optimal observations $\mathbf{x}_i$ are computed by Sequential Least Squares Programming (SLSQP) [32] given the known values for prices $\mathbf{p}_i$ and incomes m_i, for $i = 1, \ldots, N$. The dataset is then denoted by $D = \{\mathbf{x}_i, \mathbf{p}_i, m_i\}_{i=1}^N$. Generating the observations in this way ensures that the dataset is GARP-consistent as the agent is strictly utility maximising without deviations.

5.1.1 Utility Estimated by Cobb-Douglas

If the true utility which generated the observations is of CD form and a CD function is being fitted, it is shown that PEARL successfully recovers the true parameters. For this illustration, we generate a small sample size of 160 observations, with bundles made up of 2 goods ($k = 2$) with CD parameters $\boldsymbol{\theta} = [0.4, 0.6]$. We chose $N = 160$ because this corresponds to one batch of size 128 after a $80 - 20$ train-test split, which removes any noise across batches. We have chosen these parameter values as an illustration; the same result applies across all valid parameter values. The loss and the gradients computed by the algorithm at various initial parameters $\hat{\boldsymbol{\theta}}$ are inspected. With the constraint that $\sum_i^k \hat{\theta}_i = 1$, there is only 1 parameter we need to estimate. We show that the loss and gradients differ with respect to different initial values for θ_1. This is illustrated in Figure 1. Here, we see that for any initial value, the gradient update (by negative gradient) will ensure that the parameter values will converge at the true value, $\theta_1 = 0.4$, which is where the loss is equal to 0.
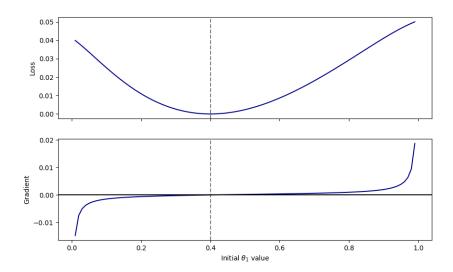

Figure 1: Loss (top) and gradient (bottom) computed at varying initial values of $\hat{\theta}_1$, with true value at $\theta_1 = 0.4$ (dashed, grey). The loss is minimised and gradients are zero at the true value. Since the gradients cross the x-axis only once, the parameter is guaranteed to converge approximately at the true value with PEARL.

5.1.2 Utility Estimated by ICNN

The ICNN, as described in section 4.4.2, is shown to be able to conform to the functional form of CD given a generated dataset. $N = 160, k = 2$ data points are generated with $\theta_1 = 0.4$ and an ICNN is fit with 3 layers and 2 hidden units at each layer. The fit of the ICNN utility function recovered via PEARL is illustrated by plotting the indifference curves (contours) of the utility function in Figure 2. The observations are indicated in grey, the ground truth contours in dashed black and the fitted model contours in blue. For a good fit, the fitted level sets should be parallel to the ground truth level sets

11

since the utility function is ordinal and the shape of the level set is most important to obtain the optimal solution, given a budget constraint.
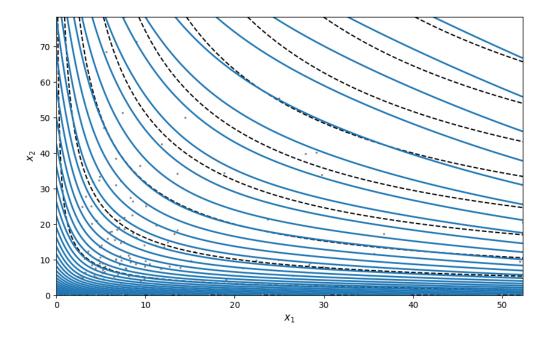

Figure 2: Comparison of contours of fitted ICNN utility function (solid, blue lines) by PEARL and the ground truth CD utility function (black, dashed lines) with $\boldsymbol{\theta} = [0.4, 0.6]$. Observations shown in grey with $N = 160$, $k = 2$.

5.1.3 Computing Elasticities

For the case where the Cobb-Douglas functional form is known and we get back the parameters, the elasticity computation is trivial, but the accuracy of the Algorithm 3 presented in section 4.3 can be tested. As such, the elasticities are computed by the method in section 4.3.2. For k products, the ground truth price elasticities are $e = -\mathbf{I}$, where $\mathbf{I}$ represents the $k \times k$ identity matrix. To illustrate, choose $k = 5$ For the same data as described in section 5, with $k = 5$, the elasticities estimated by the fitted utility function of CD and ICNN, $\hat{e}_{CD}$ and $\hat{e}_{ICNN}$, are presented in Figure 3.
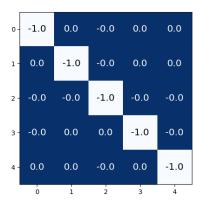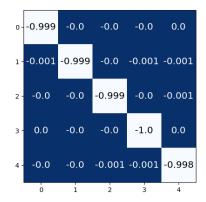
5.1.4 Estimating Demand from Utility

Given the fitted utility functions using the PEARL algorithm, a counterfactual prediction of demand given a set of prices and income which form the budget constraint can be obtained by maximising the utility function, as in 4.3. The RMSE is computed, given by

$$RMSE(\hat{\mathbf{x}}_i, \mathbf{x}_i) = \sqrt{\frac{1}{N_s} \sum_{i}^{N_s} \sum_{j}^{k} (\hat{x}_i^j - x_i^j)^2} \tag{31}$$

which refers to the difference in the actual predicted demand for each good in $\hat{\mathbf{x}}_i$, where $\mathbf{x}_i^j$ denotes the j-th element (good) of the observed bundle of goods for $i = 1, \ldots, N_s$ observations in the test set. A total of N observations are sampled with an $80\% - 20\%$ train-test split.

We investigate the performance of each component of PEARL with an ablation study. We keep $k = 2$ and $N = 160$ and the network architecture is fixed as described in section 4.4.2. We compare RMSE scores for different activation functions in the ICNN fitted with PEARL, CD fitted with PEARL. We replace the concave activation functions with commonly used activation functions, ReLU [3] and sigmoid [26], and remove the postivive constraints imposed on the weights. The activation functions we

12

(a) Price elasticities for $k = 5$ from CD.

(b) Price elasticities for $k = 5$ from ICNN.

Figure 3: Own and cross-price elasticities numerically estimated after CD (left) and ICNN (right) utility functions are trained by PEARL on $N = 1600, k = 5$.

consider for the ICNN are the three concave activation functions, presented in section 4.4.2, *concave-tanh*, *concave-sigmoid* and *concave-log*. The results are presented in Table 1.

Table 1: Demand prediction errors (RMSE) for simulated data generated with k goods and N observations using Cobb-Douglas. We compare a neural network regression with ReLU and sigmoid activation functions, PEARL fitted with non-input-concave neural networks and PEARL with ICNN and CD.
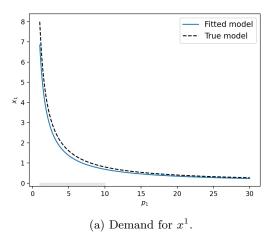
k	N	PEARL (NN) *ReLU*	PEARL (NN) *sigmoid*	PEARL (ICNN) *concave-tanh*	PEARL (ICNN) *concave-sigmoid*	PEARL (ICNN) *concave-log*	PEARL (CD) *-*
2	160	24.329	15.127	0.197	0.340	0.009	**0.002**

The performance is compared to that of machine learning regression methods for demand prediction, as shown in [6], who state that these machine learning methods outperform other forms of demand estimation. These include linear regression, LASSO, Random forest (RF), support vector machine (SVM), bagging, and k-nearest neighbours regression and XGBoost. The input is the normalised price (price divided by income) and the output is the amount consumed. The results are shown in Table 2 where the root mean squared error (RMSE) for the train and test data is reported. We compare $k = 2, N = 160$ as before. We also now increase the number of goods with values to 5 and 10, with the number of observations set at $N = 1600$. With a $80 - 20$ train-test split, we get 10 batches of size 128, which corresponds to approximately 3.5 years of daily consumption data.

Table 2: Demand prediction errors (RMSE) for simulated test data generated with k goods and N observations using Cobb-Douglas, fitted with PEARL ICNN and CD, compared to various machine learning methods (linear regression, LASSO, random forest (RF), support vector machines (SVM), Bagging, K-nearest neighbours (KNN), and XGBoost).

k	N	Linear	LASSO	RF	SVM	Bagging	KNN	XGBoost	PEARL (ICNN)	PEARL (CD)
2	160	14.118	19.613	4.647	14.855	6.821	10.655	1.984	0.009	**0.002**
5	1600	6.789	9.971	1.837	4.647	1.992	4.254	0.291	0.013	**0.012**
10	1600	4.984	7.383	2.764	3.899	2.945	4.310	0.299	0.052	**0.027**

The demand function for one good is visualised, keeping the prices of the other goods fixed, which is compared to the analytical result from the ground truth. Notice that in the simulation, the price range
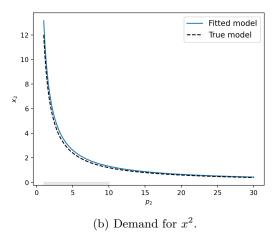
(a) Demand for x^1.   (b) Demand for x^2.

Figure 4: Illustration of comparison of demand functions for (a) x^1 and (b) x^2 between ground truth (black, dashed) and the demand obtained from maximising the fitted ICNN (blue). Prices for all the other goods have been fixed at 5 and income at 20. Observations are generated for prices between 1 and 10, as indicated by the grey bars.

is between 1 and 10; the plots illustrate that the function can extrapolate beyond the observed range. This is shown in Figure 4.

5.2 Simulations with Noise

Next, perturbations are applied to the utility function or to the observations to simulate deviations from the utility-maximising consumer with constant preferences. Two cases are considered, inspired by [45]. These are random utility, resulting from e.g. measurement error; and endogeneity, resulting in e.g. correlation between the parameters of the utility function and the prices observed. Using PEARL, the two specified utility functions of CD and ICNN form are fitted with the ε-adjustment, as described in section 4.3.1, and compare them to machine learning regression methods. The results show that in each of the cases described, the PEARL algorithm outperforms the benchmark.

5.2.1 Random Utility

Inconsistencies arising from measurement error or irrational behaviour leading to variation in the optimal consumption bundle denoted by $\mathbf{x}$ can be simulated by introducing a random error η as a perturbation. As such, the observed consumption choices are now

$$\hat{\mathbf{x}}_i = \max_{\mathbf{x}}\{U_\theta(\mathbf{x}) \text{ s.t. } \mathbf{p}_i^T\mathbf{x} \leq m_i\} - |\eta|, \tag{32}$$

where $\eta \sim \mathcal{N}(\mathbf{0}, \Sigma)$ with Σ as the $k \times k$ positive semi-definite correlation matrix across the components of the consumption bundle. The observed income is then updated to $m_i = \mathbf{p}_i^T\hat{\mathbf{x}}_i$. This means that the observed consumption bundles may not be the ones that would have been obtained under the updated income value, thus resulting in deviations from the optimal and rational consumer.

The RMSE values are reported for the counterfactual prediction of optimal consumption bundles resulting from the PEARL algorithm with ε-adjustment using CD and ICNN utility functions and their comparisons to machine learning regression method in Table 3. We report the RMSE values for settings of $k = 2, N = 160$ and $k = 5, N = 1,600$.

5.2.2 Price Endogeneity

In a system where prices are set dynamically as a response to (expected) demand, it is difficult to determine whether changes in consumption are due to changes in preferences or as a response to prices, or both. Such an environment is simulated by incorporating an error term which varies the parameters of the CD utility function, $\boldsymbol{\theta}$, and also varies the generated prices. As such, the prices are correlated with the parameters, which simulates the endogenous nature of prices. The system is described by the $k \times 1$

14

Table 3: Demand prediction errors (RMSE) for simulated test data generated with k goods and N observations using Cobb-Douglas with random noise, for fitted CD and ICNN utility functions using PEARL with ε-adjustment, compared to various machine learning methods (linear regression, LASSO, random forest (RF), support vector machines (SVM), Bagging, K-nearest neighbours (KNN), and XGBoost).

k	N	ε	Linear	LASSO	RF	SVM	Bagging	KNN	XGBoost	PEARL (ICNN)	PEARL (CD)
2	160	0.992	11.459	14.473	3.446	10.065	3.810	5.358	1.857	**0.238**	1.679
5	1600	0.844	7.370	9.926	2.571	5.435	2.829	5.018	2.107	**0.830**	2.664

vector of prices $\mathbf{p}$ being perturbed by a random error, η, such that the observed prices are now $\mathbf{p}' = \mathbf{p} + \eta$, where η is a $k \times 1$ vector of random errors, with each element $j = 1, \ldots, k$ drawn from independently from $\eta^j \sim \mathcal{N}(0, 0.01)$ and with each element $j = 1, \ldots, k$ of the price vector drawn independently from $p^j \sim \mathcal{U}[1, 10]$. These errors are also applied to the $k \times 1$ vector of true parameters of the utility function, $\boldsymbol{\theta}$, giving $\boldsymbol{\theta}' = \boldsymbol{\theta} + \eta$. Applying the same perturbation to both the parameters and prices acts as a 'demand shock' and leads to correlation between the price and the parameters of the utility function, thus simulating endogeneity. The RMSE values are reported for the counterfactual prediction of optimal

Table 4: Demand prediction errors (RMSE) for simulated test data with k goods and N observations, generated using Cobb-Douglas with endogeneity, for fitted CD and ICNN utility functions using PEARL with ε-adjustment, compared to various machine learning methods (linear regression, LASSO, random forest (RF), support vector machines (SVM), Bagging, K-nearest neighbours (KNN), and XGBoost).

k	N	ε	Linear	LASSO	RF	SVM	Bagging	KNN	XGBoost	PEARL (ICNN)	PEARL (CD)
2	160	0.999	9.785	14.470	3.771	10.026	4.365	5.604	3.234	2.197	**2.018**
5	1600	0.909	7.223	10.267	3.987	5.628	4.193	5.537	3.846	**3.065**	4.530

consumption bundles resulting from the PEARL algorithm with ε-adjustment using CD and ICNN utility functions and their comparisons to machine learning regression method in Table 4. We report the RMSE values for settings of $k = 2, N = 160$ and $k = 5, N = 1,600$.

6 Conclusion

Demand estimation is a challenging task due to the endogenous nature of price and demand. Existing methods which seek to identify the effect of price on demand have stringent IV requirements to determine exogenous variation and isolate the relationship between the two variables while accounting for the effect of demand shocks. These methods face difficulties in specifying IVs and the number required scales rapidly with the number of endogenous variables. Many commonly used methods which bypass this problem and do not rely on IVs are not able to identify the effect of price on demand because they do not address the problem of endogeneity.

To address these challenges, we introduced PEARL, an algorithm that de-couples the effect of price on demand by explicitly modelling the consumer's decision-making process. PEARL is an algorithm that successfully recovers a representation of the underlying utility function of a utility-maximising consumer under a budget constraint, which generated observed consumption data as a result of the utility-maximising process. It uses economic theory from revealed preferences to obtain conditions for ensuring that the data set is consistent with a rational utility-maximising consumer. Under these conditions, a utility function that rationalises the observed data set is guaranteed to exist. In such a case, the algorithm obtains a representation of such a function by an iterative process. We also introduced a neural network architecture, the Input-Concave Neural Network (ICNN) that is concave with respect to the input with the use of one of the three concave activation functions: *concave-tanh*, *concave-sigmoid* and *concave-log*.

PEARL was tested on simulated data with and without noise, and real-world data. Given that the

functional form is known, it has been shown that the true parameters which generated the data set can be recovered. We presented a neural network, the Input-Concave Neural Network (ICNN), which satisfies the conditions of a utility function through concave activation functions and constraints on the weights of the network. It can be used as a flexible model to recover the utility function. It has been shown to obtain small errors on very small sample sizes. These cases illustrated the performance on small sample sizes and a small number of goods. Further analysis is required to investigate its performance on larger sizes of bundles and samples, as well as more fine-tuned training and model optimisation per setting.

A key limitation is that PEARL does not account for changing preferences or parameters of utility functions. The algorithm ensures that the data on which the model is trained are GARP consistent and does not utilise the calculated error, ε, aside from adjusting the observations. As such, the predictions resulting from the methodology will only ever be GARP-consistent (possibly with adjustment) with previous preferences as the change over time is not explicitly treated.

Moreover, PEARL is built upon assumptions of well-behaved preferences, especially monotonicity and convexity, which means the consumer always wants to consume more goods and a mix of both goods is better. This is a strong assumption on consumer behaviour and while it may be reasonable to assume this in consumption settings such as in supermarkets, it may not be applicable to different scenarios, especially when decisions are more sparse over time. The result of these assumptions is the existence of a concave utility function under GARP-consistent observations. These assumptions may not hold in other scenarios and, as such, the utility function may not be concave. We have also assumed that the observations come from the same deterministic utility maximisation objective, as either the representative agent or homogenous agents; accounting for heterogeneity across individuals is an avenue for future work.

References

[1] S. N. Afriat. The construction of utility functions from expenditure data. *International Economic Review*, 8(1):67–77, 1967.

[2] S. N. Afriat. On a system of inequalities in demand analysis: An extension of the classical method. *International Economic Review*, 14(2):460–472, 1973.

[3] A. F. Agarap. Deep learning using rectified linear units (relu), 2019.

[4] B. Amos, L. Xu, and J. Z. Kolter. Input convex neural networks. In *International Conference on Machine Learning*, pages 146–155. PMLR, 2017.

[5] J. Angrist and G. Imbens. Identification and estimation of local average treatment effects, 1995.

[6] P. Bajari, D. Nekipelov, S. P. Ryan, and M. Yang. Machine learning methods for demand estimation. *The American Economic Review*, 105(5):481–485, 2015.

[7] S. T. Berry. Estimating discrete-choice models of product differentiation. *The RAND Journal of Economics*, pages 242–262, 1994.

[8] S. T. Berry, M. Carnall, and P. T. Spiller. Airline hubs: costs, markups and the implications of customer heterogeneity, 1996.

[9] S. T. Berry and P. A. Haile. Chapter 1 - foundations of demand estimation for the handbook of industrial organization. In K. Ho, A. Hortaçsu, and A. Lizzeri, editors, *Handbook of Industrial Organization, Volume 4*, volume 4 of *Handbook of Industrial Organization*, pages 1–62. Elsevier, 2021.

[10] R. Blundell and J. L. Powell. Endogeneity in nonparametric and semiparametric regression models. *Econometric society monographs*, 36:312–357, 2003.

[11] R. Carbonneau, K. Laframboise, and R. Vahidov. Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research*, 184(3):1140–1154, 2008.

[12] C. W. Cobb and P. H. Douglas. A theory of production. *The American Economic Review*, 18(1):139–165, 1928.

[13] P. Cohen, R. Hahn, J. Hall, S. Levitt, and R. Metcalfe. Using big data to estimate consumer surplus: The case of uber. Working Paper 22627, National Bureau of Economic Research, September 2016.

[14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2022.

[15] G. Debreu. Representation of a preference ordering by a numerical function. *Decision processes*, 3:159–165, 1954.

[16] W. E. Diewert. Afriat and revealed preference theory. *The Review of Economic Studies*, 40(3):419–425, 1973.

[17] P. B. P. B. Dixon, S. Bowles, and D. A. D. A. Kendrick. *Notes and problems in microeconomic theory / Peter B. Dixon, Samuel Bowles, David Kendrick; in collaboration with Lance Taylor and Marc Roberts.* Advanced textbooks in economics ; v.15. North-Holland Pub. Co., Amsterdam ;, 1979.

[18] R. P. Ferreira, A. Martiniano, A. Ferreira, and R. J. Sassi. Study on daily demand forecasting orders using artificial neural network. *IEEE Latin America Transactions*, 14(3):1519–1525, 2016.

[19] F. Forges and E. Minelli. Afriat's theorem for general budget sets. *Journal of Economic Theory*, 144(1):135–145, 2009.

[20] J. A. Hausman and D. A. Wise. A conditional probit model for qualitative choice: Discrete decisions recognizing interdependence and heterogeneous preferences. *Econometrica: Journal of the econometric society*, pages 403–426, 1978.

[21] J. Heckman and R. Pinto. Causal analysis after haavelmo. *Econometric Theory*, 31(1):115–151, 2015.

[22] J. J. Heckman, H. Ichimura, J. A. Smith, and P. E. Todd. Characterizing selection bias using experimental data, 1998.

[23] J. J. Heckman and R. Robb Jr. Alternative methods for evaluating the impact of interventions: An overview. *Journal of econometrics*, 30(1-2):239–267, 1985.

[24] J. J. Heckman and E. Vytlacil. Structural equations, treatment effects, and econometric policy evaluation 1. *Econometrica*, 73(3):669–738, 2005.

[25] J. J. Heckman and E. J. Vytlacil. Local instrumental variables and latent variable models for identifying and bounding treatment effects. *Proceedings of the national Academy of Sciences*, 96(8):4730–4734, 1999.

[26] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[27] H. S. Houthakker. Revealed preference and the utility function. *Economica*, 17(66):159–174, 1950.

[28] M. Houtman and J. Maks. Determining all maximal data subsets consistent with revealed preference. *Kwantitatieve Methoden*, 19:89–104, 1985.

[29] W. Karush. Minima of functions of several variables with inequalities as side constraints. *M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago*, 1939.

[30] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.

[31] L. Kocoska, A. Eberhard, D. Ralph, and S. Schreider. Numerical simulations of techniques related to utility function and price elasticity estimators. In *Proceedings of the 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation*, pages 1154–1160. Modelling and Simulation Society of Australia and New Zealand, 2009.

[32] D. Kraft. A software package for sequential quadratic programming. *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*, 1988.

[33] V. Krishnamurthy, D. Angley, R. Evans, and W. Moran. Identifying cognitive radars – inverse reinforcement learning using revealed preferences. 2019.

[34] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley and Los Angeles, 1951. University of California Press.

[35] A. Lahouar and J. Ben Hadj Slama. Random forests model for one day ahead load forecasting. In *IREC2015 The Sixth International Renewable Energy Congress*, pages 1–6, 2015.

[36] 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.

[37] D. McFadden. Modelling the choice of residential location. Cowles Foundation Discussion Papers 477, Cowles Foundation for Research in Economics, Yale University, 1977.

[38] D. McFadden. Econometric models of probabilistic choice. *Structural analysis of discrete data with econometric applications*, 198272, 1981.

[39] M. Mogstad and A. Torgovitsky. Identification and extrapolation of causal effects with instrumental variables. *Annual Review of Economics*, 10:577–613, 2018.

[40] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 663–670, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[41] P. A. Samuelson. Consumption theory in terms of revealed preference. *Economica*, 15:243, 1948.

[42] P. A. Samuelson. Complementarity: An essay on the 40th anniversary of the hicks-allen revolution in demand theory. *Journal of Economic literature*, 12(4):1255–1289, 1974.

[43] B. Smeulders, Y. Crama, and F. C. Spieksma. Revealed preference theory: An algorithmic outlook. *European Journal of Operational Research*, 272(3):803–815, 2019.

[44] L. G. Telser. Iterative estimation of a set of linear regression equations. *Journal of the American Statistical Association*, 59(307):845–862, 1964.

[45] E. Tipoe, A. Adams, and I. Crawford. Revealed preference analysis and bounded rationality. *Oxford Economic Papers*, 74(2):313–332, 08 2021.

[46] M. Ulrich, H. Jahnke, R. Langrock, R. Pesch, and R. Senge. Distributional regression for demand forecasting in e-grocery. *European Journal of Operational Research*, 294(3):831–842, 2021.

[47] H. R. Varian. The nonparametric approach to demand analysis. *Econometrica: Journal of the Econometric Society*, pages 945–973, 1982.

[48] H. R. Varian. Goodness-of-fit in optimizing models. *Journal of Econometrics*, 46(1):125–140, 1990.

[49] H. R. Varian. Intermediate microeconomics: a modern approach. 2006.

[50] J. Von Neumann and O. Morgenstern. Theory of games and economic behavior, 2nd rev. 1947.

[51] S. Warshall. A theorem on boolean matrices. *J. ACM*, 9:11–12, 1962.

[52] M. Wulfmeier, P. Ondruska, and I. Posner. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888*, 2015.

[53] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.