Learning Non-parametric Choice Models with Discrete Fourier Analysis

[1], Haoyu Song[1], Hai H. Nguyen[1], and Thanh Nguyen[2]

[1]Department of Computer Science, Purdue University
[2]Mitchell E. Daniels, Jr. School of Business, Purdue University

January 21, 2024

Abstract

Non-parametric choice models offer broad applicability and robustness. However, the exponentially large parameter space leads practitioners to use heuristics for estimation. We introduce an alternative approach to modeling and estimating non-parametric choice models using discrete Fourier analysis. We demonstrate that any choice function can be approximated with a small number of Fourier parameters. Our sample-efficient, active-learning algorithm, without requiring an explicit model description, needs at most $\mathsf{poly}(\log n, \frac{1}{\varepsilon})$ data queries to estimate any choice function up to ε accuracy. Computational studies show significant error reduction with Fourier methods compared to common heuristics for non-parametric choice estimation in both simulated and real data.

1 Introduction

Choice models play a crucial role in informing operational and economic decisions. Recently, there has been a significant focus on non-parametric choice models, driven by their flexibility and robustness. These models stand out for their ability to capture intricate patterns and relationships without depending on a predetermined functional form. Additionally, they avoid making explicit assumptions about the distribution of the data. Two notable models are the random rankings model (Farias et al. (2013)) and the decision forest model (Chen and Mišić (2022)). Their greatest appeal is generality. It has been known that all rational choice functions (RUM) can be expressed as a distribution over rankings Luce (1959). Furthermore, Chen and Mišić (2022) goes a step further, demonstrating that any choice function, irrespective of its rationality, can be represented as a decision forest.

However, the broad applicability of these non-parametric models comes with a trade-off. The exponential size of the parameter space presents two challenges: computational and the need for large amounts of data. Hence practitioners often rely on heuristics like random sampling and heuristic choice generation during the estimation of non-parametric choice models, which lack theoretical guarantee and can be far from optimal. For instance, in the case of the decision forest model, a common approach is to consider sub-classes, such as decision forests with bounded depth/leaf-size. However, this sub-class is also extensive when the number of products is large. Consequently, estimating them becomes a challenging task: a substantial number of samples will be required, and solving complicated optimization problems becomes necessary.

Our paper introduces an alternative approach to modeling and estimating non-parametric choice models through discrete Fourier analysis. To the best of our knowledge, this has not been explored previously. The advantage of our approach is twofold. Firstly, it is agnostic to specific modeling assumptions. Secondly, it provides an algorithm to estimate choice models with theoretical guarantees on sample size.

Discrete Fourier analysis describes functions on a discrete domain through a linear combination of a set of "basic" functions known as the Fourier basis. These basis functions are highly structured, facilitating easy understanding and estimation. Each basis function depends on a subset of the underlying variables (each variable represents a product in our setting). The size of the subset determines the "complexity" of the basis function, also referred to as its degree. Two important measures of a choice function are its sparsity (the number of basic functions needed to describe it) and its degree (the maximum degree of the basis in its Fourier expression).

First, we show that non-parametric choice models can be approximated by functions with low sparsity. An essential technical aspect to prove this result involves the use of the L_1 Fourier norm. In particular, we show that, regardless of the number of trees involved, a decision forest's L_1 Fourier norm is upper bounded by the maximum number of leaves in a tree. Building on this, we exploit the characterization of choice functions as binary choice forests to establish that any choice function is ε-close to a decision forest with a leaf size of $\log(\frac{1}{\varepsilon})$, and therefore has a concise Fourier representation.

Expanding on this insight, our second contribution is to design an efficient active-learning algorithm estimating parameters in the Fourier domain tailored for choice functions. There are two significant features of our algorithm. First, it does not depend on the explicit description of the choice model (say the number of trees), but rather on the natural Fourier properties of the underlying choice function, which we show to be small. Second, the sample complexity of the learning algorithm scales with $\log(n)$ rather than n, where n is the number of products. Applying the Goldreich-Levin algorithm (Goldreich and Levin, 1989; Kushilevitz and Mansour, 1993) to the high-dimensional choice setting is feasible but demands a sample complexity polynomial in n, impractical in many real-world applications. To achieve lower sample complexity, we show that the considered choice functions have a Fourier degree d significantly smaller than n. Leveraging this low-degree property, we integrate the Goldreich-Levin algorithm with the hashing technique from Amrollahi et al. (2019), resulting in a sample complexity polynomial in $\log n$.

In the specific case of ranking models, we develop a distinct algorithm. By leveraging a novel inductive structure on the Fourier spectrum of ranking models, the algorithm also scales proportionally to $\log(n)$. Although less general than the first one, this algorithm is built upon the easy procedure of estimation through uniform sampling, which involves querying uniform samples and taking an average of the returned results, thus making its implementation much more straightforward.

In addition to theoretical results, we perform computational studies on synthetic and real datasets. We observe the Fourier-based method outperforming Multinomial Logit Models (MNL) and two common heuristics in non-parametric choice model estimation, reducing Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) by over 20% on average.

1.1 Related works

The literature on modeling and understanding customer preferences is extensive and continuously evolving. Alongside the classical choice model of multinomial logit McFadden et al. (1973), Markov chain models Blanchet et al. (2016), and other rational choice models (refer to Ben-Akiva and Bierlaire (1999) for a survey), recent years have witnessed a growing body of literature focusing on complex non-parametric choice models capable of capturing irrational behavior (e.g., Chen and Mišić (2022), Chen et al. (2019)). This trend is fueled by the increased adoption of data-driven approaches. Our paper contributes to this expanding literature.

The works most closely related to ours include Farias et al. (2013), presenting the random rankings model, and Chen and Mišić (2022), introducing the decision forest model. These works address the challenge of managing a large number of model configurations by employing heuristics such as column generation and random sampling for model estimation. In contrast, our contribution introduces Fourier analysis, a novel addition that offers similar modeling flexibility while providing theoretical guarantees on learning algorithms.

The second line of literature related to our paper pertains to Fourier-based learning in computer science. Although this literature is extensive, its primary focus lies in learning boolean functions mapping to a binary range or real numbers (O'Donnell (2014)). For instance, Stobbe and Krause (2012) and Haviv and Regev (2017) investigated learning Boolean functions using random samples, while Mossel et al. (2003), Blum (1994), and Kolountzakis et al. (2005) delved into learning junta Boolean functions. Additionally, Linial et al. (1993) examined binary functions modeled by constant-depth circuits, and Kushilevitz and Mansour (1993) studied using Fourier analysis to learn decision trees. In contrast to these works, our focus is on choice functions that are inherently high-dimensional. While it is conceivable to treat high-dimensional functions as collections of multiple one-dimensional functions, the sample complexity required for learning increases multiplicatively with the dimension. This can become impractical for real-world applications. Our objective is to achieve a logarithmic dependency on the dimension.

The novelty of our algorithm lies in combining the idea of the Goldreich-Levin algorithm (Goldreich and Levin, 1989; Kushilevitz and Mansour, 1993) with hashing of sparse and low-degree functions. Our hashing technique is inspired by Amrollahi et al. (2019), which utilizes both approximate sparsity and low-degree but doesn't address multi-dimensional settings nor operate under the *sample oracle model* as in our paper. In particular, that paper requires the precise valuation of the function to be learned, $f(x)$, at each query point x. On the contrary, in our sample oracle model, with each query, we exclusively acquire one item i sampled from $f(x)$. This represents observing the selection of one customer. The key component for our algorithm to function in this general model is the combination of the L_2 Fourier weight estimation, as demonstrated in the Goldreich-Levin algorithm, with hashing techniques inspired by Amrollahi et al. (2019).

The rest of the paper is structured as follows. Section 2 covers the fundamentals of discrete Fourier analysis and its application to choice functions and our learning model. Section 3 demonstrates that any choice function can be approximated with a small number of parameters in the Fourier domain. The algorithms for learning the parameters of these functions in the Fourier domain are presented in Section 4. In Section 5, we present numerical results indicating that Fourier-based methods outperform MNL and common heuristics used to learn non-parametric models. The appendix contains the missing proofs.

2 Preliminaries

2.1 Choice functions

In this section, we define some important models of choice functions which will be discussed in more detail in later sections.

Let N be the set of products, and let $n = |N|$, a choice function f is a mapping from a subset of the products (an assortment) to a distribution over the items being purchased (including the option of not purchasing any item). Then each choice set can be expressed by a vector $x \in \{0,1\}^n$ where $x_i = 1$ means that the i-th item is present in the choice set. The no-purchase option is labeled $n+1$ and is assumed to be always present. And for each x, $f_i(x)$ gives the probability that the i-th item is selected. Thus a choice function can be formally defined as follows.

Definition 1. *$f : \{0,1\}^n \to R^n$ is a choice function if the following conditions are satisfied: (1) $f_i(x) = 0$ whenever $x_i = 0$ (2) $f_i(x) \geqslant 0$ (3) $\sum\limits_i f_i(x) \leq 1$.*

The Random Utility Maximization model (RUM) is the most common model of choice functions. The model is often abbreviated as the "rational model," since it matches the common sense that a rational agent will choose the option maximizing its utility.

Definition 2. *In the RUM model, each item i is associated with a fixed utility u_i and a randomized error term ε_i. Then $f_i(x)$ is the probability that $u_i + \varepsilon_i$ is greater than $u_j + \varepsilon_j$ for all j s.t. $x_j \neq 0$.*

This formulation of RUM may fail to yield a closed-form formula. Therefore, it is sometimes more desirable to work with an equivalent characterization, the *ranking model*. Each customer is represented as a permutation/ranking, who selects the item positioned at the highest place among all available options. The ranking model is formulated as a distribution over all rankings. The equivalence between the two models is first proved by Block and Marshak (1959). Though the two models are equivalent, we will exclusively use the term 'ranking model' to avoid confusion.

Definition 3. *For each permutation/ranking σ over $[n+1]$, there is an associated ranking function $r_\sigma : \{0,1\}^n \to R^n$ s.t. $[r_\sigma(x)]_i = 1$ if $\sigma(i) < \sigma(j)$ for all j s.t. $x_j = 1$. And a choice function $f : \{0,1\}^n \to R^n$ is a ranking model if $f = \sum\limits_{\sigma \in K_{n+1}} c_\sigma r_\sigma$, where K_{n+1} denotes the set of permutations/rankings over $[n+1]$ and $\sum c_\sigma = 1$.*

Recently Chen and Mišić (2022) generalized the ranking model to the decision forest model (DF) to capture irrational choice behavior, in which each customer type is represented as a decision tree.

An agent starts from the root node of the corresponding decision tree. At each node i, the agent moves to the left child node if the i-th product is present and to the right child otherwise. This process continues until reaching a leaf node ℓ, and the agent then selects the ℓ-th item.

Figure 1 visualizes an example of a decision tree. If the choice set is $S = \{1, 2, 3\}$, then the customer will choose product 2; if $S = \{1, 4\}$, she will choose no purchase option 0.

To formulate the choice function corresponding to a decision tree as a function f mapping from $\{0, 1\}^n$ to $\mathbb{R}^n$, we give an alternative representation of the decision tree as follows (refer to Figure 2 for an example). Each non-leaf node is labeled by a variable x_i. Each leaf node is labeled by a vector $e_i \in \mathbb{R}^n$, where e_0 denotes a vector of zeros, and e_i denotes the standard basis vector in $\mathbb{R}^n$. Every edge going from a parent node to a left child node is labeled by 1. Other edges are labeled by 0. So given a choice set expressed as $x \in \{0, 1\}^n$, the customer's decision can be determined from the decision tree. For instance, if $x = 1110$ (corresponding to the case $S = \{1, 2, 3\}$), $f(x) = e_2$ means that the customer will choose product 2.

A decision forest is a choice model that can be expressed as a distribution of decision trees.

Definition 4. *A choice function $f : \{0, 1\}^n \to R^n$ is a decision forest if $f = \sum\limits_{T \in T_n} c_T T$, where T_n denotes the set of all trees with depths smaller than n and $\sum\limits_{T \in T_n} c_T = 1$.*

We use two parameters to measure the size of a decision tree. One is *leaf-size*, which is the number of leaves. Another is *depth*, the maximum number of nodes a customer will encounter in a path from the root to a leaf. The leaf-size/depth of a decision forest is defined as the maximum leaf-size/depth of trees with non-zero weights.

An alternative model equivalent to the decision forest, recently proposed by Chen et al. (2019), is the binary choice forest model (BCF). While the model is also expressed as a distribution over trees, each tree has a much simpler form: there is only one node at each level. We will discuss the model more thoroughly in section Section 3, where we use the formulation to derive Theorem 4.

We make Figure 3 to clarify the relations among the models mentioned in this section.

2.2 Discrete Fourier Analysis

This section presents the basics of discrete Fourier analysis which we shall use as the technical tools for our learning algorithms. We start with the straightforward case of real-valued functions.

Fourier basics of real-valued functions

Let $f, g \colon \{0, 1\}^n \to \mathbb{R}$ be two real-valued functions. The inner product of two functions f and g is defined as

$$\langle f, g \rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x) \cdot g(x) = \mathop{\mathbb{E}}_{x \in \{0,1\}^n} [f(x) \cdot g(x)] \,.$$

For each $S \subseteq [n]$, the characteristic function $\chi_S(x) = (-1)^{\sum_{i \in S} x_i}$ is a *linear* function that computes the parity of the bits $(x_i)_{i \in S}$. The set of all χ_S forms an orthonormal basis for the space of real-valued functions on $\{0, 1\}^n$ and are called *Fourier bases*, or bases for short. When it is clear from context, we will use S to stand for χ_S. Notice that we abuse notation to use S to represent both a set and a Fourier base.

Any function f can be uniquely expressed as $f = \sum_{S \subseteq [n]} \widehat{f}(S) \chi_S$ which is called the *Fourier expansion* of f. $\widehat{f}(S)$ is called the *Fourier coefficient* of f at S and by orthonormality can be evaluated as $\widehat{f}(S) = \langle f, \chi_S \rangle = \mathbb{E}_{x \in \{0,1\}^n} f(x) \chi_S(x)$. We say that $\widehat{f}(S)$ is the *coefficient* of the base

S and S is the *location* of $\widehat{f}(S)$. The L_2 *Fourier weight* of f on a base/set $S \subseteq [n]$ is defined to be $\widehat{f}(S)^2$. Parseval's identity says that $\mathbb{E}_{x\in\{0,1\}^n} f(x)^2 = \sum_{S\subseteq[n]} \widehat{f}(S)^2$. We denote $|S|$ as the size of the set S. We refer the readers to O'Donnell (2014) for more details on Fourier analysis.

Fourier basics of vector-valued functions

Fourier analysis is naturally extended to function mapping from $\{0,1\}^n$ to $\mathbb{R}^n$. The *inner product* of any two functions $f, g\colon \{0,1\}^n \to \mathbb{R}^d$ is defined as:

$$\langle f, g\rangle := \mathop{\mathbb{E}}_{x\in\{0,1\}^n}[f(x)\cdot g(x)],$$

where $f(x)\cdot g(x)$ denotes the dot product of the two vectors $f(x), g(x) \in \mathbb{R}^n$. For any vector $x \in \mathbb{R}^n$, the *p-norm* of x is defined as $\|x\|_p := (x_1^p + x_2^p + \ldots + x_n^p)^{1/p}$. For any function $f\colon \{0,1\} \to \mathbb{R}^n$, the L_p norm of f is defined as

$$\|f\|_p := \left(\mathop{\mathbb{E}}_{x\in\{0,1\}^n} \|f(x)\|_p^p\right)^{1/p} = \left(\mathop{\mathbb{E}}_{x\in\{0,1\}^n} \sum_{i=1}^{d} |f(x)_i|^p\right)^{1/p}.$$

Also, given any two functions f, g, their L_p distance is defined as

$$L_p(f, g) = \|f - g\|_p^p.$$

And when f is the target function, we will refer to this as the L_p error of g.

As same as in the one-dimensional case, the *characteristic function* $\chi_S\colon \{0,1\}^n \to \mathbb{R}$ is defined as $\chi_S(x) = (-1)^{\sum_{i\in S} x_i}$. It follows immediately from the uniqueness of Fourier expansion in one-dimensional output space that the Fourier expansion in multi-dimensional output space is also unique. That is, any function $f\colon \{0,1\}^n \to \mathbb{R}^n$ can be uniquely represented as

$$f(x) = \sum_{S\subseteq[n]} \widehat{f}(S)\chi_S(x).$$

Now the *Fourier coefficient* $\widehat{f}(S)$ is a vector in $\mathbb{R}^n$ for every $S \subseteq [n]$, and $\chi_S(x) = (-1)^{\sum_{i\in S} x_i}$ and can be evaluated as $\widehat{f}(S) = \mathbb{E}_{x\in\{0,1\}^n} f(x)\chi_S(x)$. Again, we say that $\widehat{f}(S)$ is the *coefficient* of the base S and S is the *location* of $\widehat{f}(S)$.

For each integer p, define the L_p *Fourier weight* of a base S to be $||\widehat{f}(S)||_p^p$. And the L_2 Fourier weight is particularly important due to the following fact.

Fact 1 (Paserval's identity)**.** $\|f\|_2^2 = \sum_{S\subseteq[n]} \left\|\widehat{f}(S)\right\|_2^2$.

Intuitively, the identity says that the square of the L_2 norm of a function is equal to the sum of the L_2 Fourier weights of all bases.

Also, for any function $f\colon \{0,1\}^n \to \mathbb{R}^n$, we define the L_1 *Fourier norm* of f as

$$\hat{\|} f \hat{\|} := \sum_{S\subseteq[n]} \left\|\widehat{f}(S)\right\|_1.$$

The following proposition states some properties of the L_1 Fourier norm which will be useful for proving Fourier properties of decision trees. It follows directly from the definition of the L_1 Fourier norm and the triangle inequality.

Proposition 1. *Let $f, g\colon \{0,1\}^n \to \mathbb{R}^n$. Then, the following hold.*

1. $\hat{\|} f+g \hat{\|} \leqslant \hat{\|} f \hat{\|} + \hat{\|} g \hat{\|}$.

2. $\hat{\|} cf \hat{\|} = c \hat{\|} f \hat{\|}$ *for any real value* $c \geqslant 0$.

Next, we present the definitions of Fourier sparsity and the degree of a function, which are key measures in our approach.

Definition 5. *Given $f\colon \{0,1\}^n \to \mathbb{R}^n$, we say that $g\colon \{0,1\}^n \to \mathbb{R}^n$ is $\underline{\varepsilon\text{-close}}$ to f if $L_2(f,g) = \|f-g\|_2^2 \leqslant \varepsilon$*

Definition 6. *We define the $\underline{\text{degree}}$ of a function $f\colon \{0,1\}^n \to \mathbb{R}^n$, denoted by* $\deg(f)$*, to be* $\max\{|S|\colon \widehat{f}(S) \neq 0\}$.

Definition 7. *Let $f\colon \{0,1\}^n \to \mathbb{R}^n$. f is $\underline{s\text{-sparse}}$ if the number of non-zero Fourier coefficients is at most s.*

3 Fourier Representation of Non-parametric Choice Models

In this section, we initially demonstrate that a decision forest with a small leaf-size can be approximated by a limited number of Fourier bases, irrespective of the number of trees involved in the forest. Subsequently, leveraging this result and the characterization of choice functions as binary choice forests, we establish that any choice function possesses a sparse Fourier representation.

3.1 Decision Forest with Bounded Leaf-size

The alternative method of representing the decision tree within the decision forest model enables us to effectively apply discrete Fourier analysis. The intuition for this formulation comes from the representation of a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ as a decision tree (see section 3.2 in O'Donnell (2014)). The main difference in our case is that because we are concerned with choice models, the leaves are not restricted to $0, 1$ but instead represent one of the products that the customer chooses.

First, we need to represent a decision tree as a sum of vector indicator functions.

Let P be a path on a decision tree with internal node labels $x_1, x_2, \ldots, x_d$, and corresponding edge labels $i_1, i_2, \ldots, i_d \in \{0,1\}$, and the leaf label e_i. Then we will write $P = \{x_1 = i_1, x_2 = i_2, \ldots, x_d = i_d, e_i\}$. The indicator function of a path P, denoted as 1_P, is a function mapping from $\{0,1\}^n$ to R^n such that

$$1_P(x) = 1(x_1 = i_1) \cdot 1(x_2 = i_2) \cdots 1(x_d = i_d) \cdot e_i.$$

Example 1. *To better see how a path can be represented in this way, let us go back to Figure 2. Here the leaf is x_1 and we will use the vector e_1 to denote it. This corresponds to the fact that item 1 is the only possible output choice for this path. For the agent to follow this path, he/she will first check whether item 1 is offered. If it is not, then this path will not be followed and the output of this path will be zero. This corresponds to the indicator function $1(x_1 = 1)$. Then the agent arrives at x_2 and checks whether item 2 is not offered because he/she is now going rightwards. This corresponds to the indicator function $1(x_2 = 0)$. The logic is the same for x_4. Since the agent will arrive at the leaf only if all those conditions are met, the three indicators need to be multiplied together. Thus we have the representation*

$$1(x_1 = 1, x_2 = 0, x_4 = 1) \cdot e_1.$$

Now observe that a decision tree can be expressed in terms of path functions corresponding to various paths from the root to the leaves as:

$$f(x) = \sum_{\text{Path } P} \mathbb{1}_P(x) f(P),$$

where $f(P)$ is the label on the leaf when the function f takes the path P in its decision tree. Every indicator function of a path in a decision tree enjoys some nice Fourier properties as follows.

Lemma 1. *Let P be a path of a decision tree $T\colon \{0,1\}^n \to \mathbb{R}^n$. Suppose there are d variables in the path P. Then, it holds that (1) $\deg(\mathbb{1}_P) = d$ and (2) $\hat{\|}\mathbb{1}_P\hat{\|}_1 = 1$.*

Proof. Suppose $x_1, x_2, \ldots, x_d$ are the variables in the path P and $i_1, i_2, \ldots, i_d$ are the labels of edges in the path. Observe that each indicator function $\mathbb{1}(x_1 = i_1)$ can be written in terms of the characteristic function $\chi_{x_1}\colon \{0,1\}^n \to \{-1,1\}$ as follows.

$$\mathbb{1}(x_1 = i_1) = \frac{(-1)^{i_1}\chi_{x_1}(y) + 1}{2}$$

Therefore, we have

$$\begin{aligned}\mathbb{1}_P(x) &= \mathbb{1}(x_1 = i_1)\cdot \mathbb{1}(x_2 = i_2)\cdots \mathbb{1}(x_d = i_d)\cdot e_i\\ &= \frac{(-1)^{i_1}\chi_{x_1}(y) + 1}{2}\cdot \frac{(-1)^{i_2}\chi_{x_2}(y) + 1}{2}\cdots \frac{(-1)^{i_d}\chi_{x_d}(y) + 1}{2}\cdot e_i.\end{aligned}$$

Expanding the first d terms and using the homomorphism property of the characteristic functions $\chi_{S\cup T} = \chi_S \cdot \chi_T$ implies that there are 2^d non-zero Fourier coefficients, each is of magnitude $1/2^d$. Thus, $\mathbb{1}_P$ has degree d and $\hat{\|}\mathbb{1}_P\hat{\|}_1 = 1$. □

Next, we show that a decision tree has a bounded L_1 Fourier norm.

Lemma 2. *If $f\colon \{0,1\}^n \to \mathbb{R}^n$ is a decision tree with leaf-size s , then $\hat{\|}f\hat{\|}_1 \leqslant s$.*

Proof. Recall that $f(x) = \sum_{\text{Path } P} \mathbb{1}_P(x) f(P)$. By definition of the choice function, it holds that $\sum_i f(P)_i \leqslant 1$. By Proposition 1, we have

$$\hat{\|}f\hat{\|}_1 = \hat{\|}\sum_{\text{Path } P} \mathbb{1}_P \cdot f(P)\hat{\|}_1 \leqslant \sum_{\text{Path } P} \hat{\|}\mathbb{1}_P \cdot f(P)\hat{\|} \leqslant \sum_{\text{Path } P} \hat{\|}\mathbb{1}_P\hat{\|}\cdot \sum_i f(P)_i \leqslant \sum_{\text{Path } P} \hat{\|}\mathbb{1}_P\hat{\|} \leqslant s,$$

which completes the proof. □

Now we demonstrate that trees with small leaf-size can be approximated by low-degree bases. The intuition is that for each long path $\mathbb{1}_P$ with length ℓ, the probability that $\mathbb{1}_P(x) \neq \mathbf{0}$ is as small as $\frac{1}{2^\ell}$. So we can safely cut down a portion of the path without significantly affecting the result.

Lemma 3. *Given a decision tree T with leaf-size s, there exists a decision tree T' with depth at most $\log(\frac{s^2}{\varepsilon})$ such that $\|T - T'\|_2^2 \leqslant \varepsilon$.*

Proof. Let $P = \{x_1 = i_1, x_2 = i_2, ..., x_k = i_k, e_i\}$ be an arbitrary path of T. If $k \leqslant \log(\frac{s}{\varepsilon})$, we keep P as it is. Otherwise, let $k' = \log(\frac{s}{\varepsilon})$. Then form a path P' by removing all internal nodes after the k'-th one and output the same value e_i. Then, $T(x) \neq T(x')$ only when x takes a path of length greater than $\log(s^2/\varepsilon)$ in T. This happen with probability $2^{\log(s^2/\varepsilon)} = \varepsilon/s^2$. So we have

$$\|\mathbb{1}_P - \mathbb{1}_{P'}\|_2^2 = \Pr_{x\in\{0,1\}^n}[T(x) \neq T'(x)] \leqslant \varepsilon/s^2$$

Now we can apply triangle inequality to obtain

$$\left\|T - T'\right\|_2 \leqslant \sum_P \|1_P - 1_{P'}\|_2 \leqslant s \cdot \sqrt{\frac{\varepsilon}{s^2}} = \sqrt{\varepsilon}.$$

This implies that $\|T - T'\|_2^2 \leqslant \varepsilon$, as desired □

As a consequence of Lemma 3, a decision forest can be approximated by a low-degree decision forest as follows.

Theorem 1. *For any decision forest $f\colon \{0,1\}^n \to \mathbb{R}^n$ with leaf-size s, and $\varepsilon > 0$, there exists a decision forest $g\colon \{0,1\}^n \to \mathbb{R}^n$ such that (1) $\|f - g\|_2^2 \leqslant \varepsilon$ and (2) $\deg(g) = \log(s^2/\varepsilon)$.*

Proof. For each tree T, cut down T to T' with depth at most $\log(\frac{s^2}{\varepsilon})$ as in Lemma 3. Define $f' = \sum T'$. Then, by triangle inequality, we have

$$\left\|f - f'\right\|_2 \leqslant \sum_T c_T \left\|T - T'\right\|_2 \leqslant \max_T \left\|T - T'\right\|_2 \leqslant \sqrt{\varepsilon}.$$

Since each T' only contains Fourier bases with a degree less than $\log(\frac{s^2}{\varepsilon})$, our proof is complete. □

Now we are ready to establish the approximate Fourier sparsity of decision forests. The key insight is that any decision forest with a small leaf-size has a small Fourier L_1 norm, which then implies that the function can be approximated by a sparse function.

Lemma 4. *Let $f : \{0,1\}^n \to \mathbb{R}^n$ satisfies $\hat{\|} f \hat{\|}_1 \leqslant L$, and $\varepsilon > 0$. Then, there exists a function $g : \{0,1\}^n \to \mathbb{R}^n$ such that (1) $\|f - g\|_2^2 \leqslant \varepsilon$, and (2) g is $\frac{L^2}{\varepsilon}$-sparse.*

Proof. The main idea is that a small L_1 norm allows us to throw away many bases with a small L_1 Fourier weight. Let us define

$$\mathbb{S} = \left\{S \subseteq \{0,1\}^n \colon \left\|\widehat{f}(S)\right\|_1 \geqslant \frac{\varepsilon}{L}\right\}, \text{ and } g = \sum_{S \in \mathbb{S}} \hat{f}(S)\chi_S.$$

It is clear that $|\mathbb{S}| \leqslant \frac{L}{\varepsilon/L} = \frac{L^2}{\varepsilon}$. This implies that g contains at most $\frac{L^2}{\varepsilon}$ bases with non-zero Fourier coefficients. Using Parseval's identity, we have

$$\begin{aligned}
\|f - g\|_2^2 &= \sum_{S \notin \mathbb{S}} \left\|\widehat{f}(S)\right\|_2^2 \leqslant \max_{S \notin \mathbb{S}} \left\|\widehat{f}(S)\right\|_2 \sum_{S \notin \mathbb{S}} \left\|\widehat{f}(S)\right\|_2 \\
&\leqslant \max_{S \notin \mathbb{S}} \left\|\widehat{f}(S)\right\|_1 \sum_{S \notin \mathbb{S}} \left\|\widehat{f}(S)\right\|_1 \leqslant \max_{S \notin \mathbb{S}} \left\|\widehat{f}(S)\right\|_1 \left(\sum_{S \notin \mathbb{S}} \left\|\widehat{f}(S)\right\|_1 + \sum_{S \in \mathbb{S}} \left\|\widehat{f}(S)\right\|_1\right) \\
&\leqslant \frac{\varepsilon}{\hat{\|} f \hat{\|}_1} \cdot \hat{\|} f \hat{\|}_1 \leqslant \frac{\varepsilon}{L} \cdot L = \varepsilon.
\end{aligned}$$

This completes the proof. □

Applying the triangle inequality to the Fourier L_1 norm, we obtain the following result. The Fourier sparsity of *any* decision forest is always upper-bounded by a function that depends on s and ε, but not on the number of trees in the forest. This argument relies crucially on the fact that the sum of coefficients of the trees in the forest is bounded by 1.

Theorem 2. *Let $f\colon \{0,1\}^n \to \mathbb{R}^n$ be a decision forest with leaf-size s and $\varepsilon > 0$. Then, there exists a decision forest $f' : \{0,1\}^n \to \mathbb{R}^n$ such that (1) $\|f - f'\|_2^2 \leqslant 2\varepsilon$, (2) $\deg(f') \leqslant \log(\frac{s^2}{\varepsilon})$, and (3) f' is $\frac{s^2}{\varepsilon}$-sparse.*

Proof. Let $f = \sum_T c_T T$. As in Theorem 1, we can cut down each T to be of depth at most $\log(\frac{s^2}{\varepsilon})$. Since cutting down paths will not increase leaf-size, every resulting tree T' has a leaf-size smaller than s, and therefore its L_1 Fourier norm is smaller than s. Define $f_{cut} = \sum\limits_T c_T T'$. By Theorem 1, f_{cut} is within ε of f. Then using triangle inequality for L_1 Fourier norm, we get

$$\hat{\|} f_{cut} \hat{\|}_1 \leqslant \sum_T c_T \hat{\|} T' \hat{\|}_1 \leqslant \max_T \hat{\|} T' \hat{\|}_1 \leqslant s.$$

Then by Lemma 4, f_{cut} can be approximated within ε by a function f' consisting of at most $\frac{s^2}{\varepsilon}$ non-zero Fourier coefficients. According to Lemma 4, Fourier bases of f' are all bases of f_{cut}, which are of degree less than $\log(\frac{s^2}{\varepsilon})$. This completes the proof. □

3.2 Choice Models as Binary Choice Forests

In this section, we utilize the characterization of choice models as binary choice forests Chen et al. (2019) to show that any choice function can be approximated by a decision forest with a small leaf-size and therefore enjoys a sparse, low-degree Fourier representation.

Definition 8. *A binary choice tree f is a decision tree of n nodes (x_i, σ_i, o_i) with $c_i \in [n], \sigma_i \in \{0,1\}, o_i \in [n]$. Given $x \in \{0,1\}^n$, $f(x)$ is computed as follows: starting from $i = 0$, f outputs o_i if $x_i = \sigma_i$ and proceeds to the $(i+1)$-th node otherwise. A binary choice forest is a distribution over binary choice trees.*

Similar to ranking, a binary choice tree has only one node at each level. However, there are two distinctions. Firstly, in ranking, it is necessary that $c_i = o_i$. Secondly, in ranking, $\sigma_i = 1$ for all i—meaning we proceed only when $x_i = 0$. A binary choice tree eliminates these two constraints, making it a more general form of ranking. An example can be found in Figure 4.

Theorem 3 (Chen et al. (2019))**.** *Every choice function can be expressed as a binary choice forest with at most $n * 2^{n-1} + 1$ binary choice trees.*

Since a binary choice tree is a tree with leaf-size n, naively applying Theorem 2 yields a sparsity bound $\frac{n^2}{\varepsilon}$ and a degree bound $\log(\frac{n^2}{\varepsilon})$. However, resorting to the special structure of ranking trees, we can obtain a degree/sparsity bound completely dependent only on ε.

As seen in Figure 4, an important observation is that there is exactly one leaf at each level except the lowest level. Therefore, if we can cut down the tree to a smaller depth, the leaf size will also be reduced.

Lemma 5. *Let $R\colon \{0,1\}^n \to \mathbb{R}^n$ be an arbitrary binary choice tree. Then, there exists a decision tree $R'\colon \{0,1\}^n \to \mathbb{R}^n$ such that (1) $\|R - R'\|_2^2 \leqslant \varepsilon$, (2) the leaf size of R' is smaller than $\log(1/\varepsilon)$.*

Proof. Suppose R is given by $\{(x_1, \sigma_1, o_1), (x_2, \sigma_2, o_2) ..., (x_n, \sigma_n, o_n)\}$. Then we obtain R' by cutting down all nodes after $x_{\log(\frac{1}{\varepsilon})}$. This implies that $\deg(R') \leqslant \log(1/\varepsilon)$. Since there is only one leaf at each level, the leaf size of R' is also $\log(\frac{1}{\varepsilon})$.

Then by the definition of a binary choice tree, x_{i+1} will be accessed only if $x_j \neq \sigma_j$ for all $1, 2, ..., i$. It follows

$$\left\|R - R'\right\|_2^2 \leqslant 1 \cdot \Pr[x_1 \neq \sigma_1, x_2 \neq \sigma_2, ..., x_{\log(\frac{1}{\varepsilon})} \neq \sigma_{\log(\frac{1}{\varepsilon})}] = \left(\frac{1}{2}\right)^{\log(\frac{1}{\varepsilon})} = \varepsilon,$$

which completes the proof. □

Example 2. *To better illustrate the idea of this lemma, consider the following example. We use an example of ranking for convenience. Suppose an agent has a ranking* $\{1 \to 2 \to 3.... \to 100\}$. *Suppose we want to estimate the ranking function to the precision of* $\frac{1}{16}$. *Now we cut down the ranking to* $\{1 \to 2 \to 3 \to 4 \to 5\}$. *For the two rankings to produce different results, it must be the case that items 1,2,3,4,5 are all not available in the choice set. And this probability is* $(\frac{1}{2})^5 = \frac{1}{32} \leqslant \frac{1}{16}$ *and by Parseval, this is also the distance between the two rankings. And the new ranking's leaf size is only five. This matches the common sense that given a long ranking, only the first few items are particularly relevant.*

Combining *Lemma* 5 and *Theorem* 2, we can conclude that every binary choice forest, and therefore every choice function, has a concise Fourier representation.

Theorem 4. *For any choice function* f, *there exist a function* $g\colon \{0,1\}^n \to \mathbb{R}^n$ *such that (1)* $\|f - g\|_2^2 \leqslant \varepsilon$, *(2)* g *is* $\frac{\log^2(1/\varepsilon)}{\varepsilon}$*-sparse, and (3)* $\deg(g) \leqslant \log(1/\varepsilon)$.

4 Fourier-based Learning

We begin with a discussion of our learning model. Subsequently, we present three learning algorithms. The first one in Section 4.2 extends the well-known Goldreich-Levin algorithm, to functions with range $\mathbb{R}^n$. We include it here primarily because the subroutine for estimating Fourier coefficients is required in our second algorithm in Section 4.3. However, it's important to note that, to avoid an unnecessary blow-up of a factor of n, the analysis becomes more intricate in the multi-dimensional setting with the use of vectorized concentration inequalities. Furthermore, we demonstrate the algorithm's effectiveness in the presence of bounded noise, thereby supporting the sample-oracle model.

Our primary contribution is the learning algorithm presented in Section 4.3 designed for sparse and low-degree functions. Specifically, the algorithm involves a unique combination of Goldreich-Levin and the hashing technique in Amrollahi et al. (2019). As demonstrated in the preceding section, where we establish that all choice functions are approximately Fourier-sparse with low degree, the algorithm can be effectively applied for efficient learning.

Finally, the algorithm in Section 4.4 is designed for a special case of ranking models. Its simplicity facilitates straightforward uniform demand queries, making it attractive for practical implementation.

4.1 Data Access Model

Given a choice function $f(.)$, we are allowed to query a choice set $x \in \{0,1\}^n$, and the oracle returns a randomized output vector $y(x)$. Our purpose is to use the returned data to find a function $f'(.)$ that has a small L_2-norm error between $f(.)$ and $f'(.)$.[1]

We say that an algorithm $\mathcal{A}$ learns f to an error ε in confidence δ with m queries if, using results obtained from m queries, $\mathcal{A}$ returns a f' s.t. $L_2(f', f) \leqslant \varepsilon$ with probability at least $1 - \delta$. In this paper, we set δ to be a fixed constant, says $\frac{1}{1000}$. Boosting this probability involves repeating the algorithm a constant number of times, averaging the outcomes, and achieving a constant value arbitrarily close to 1.

Let us denote the noise function $y(x) - f(x)$ as $\rho(x)$. If for all x, $\rho(x)$ has mean zero and $\|\rho(x)\|_2 \leqslant \gamma$ with probability 1, we say that the oracle has *output noise* $\mathcal{N}(\gamma)$. If $\gamma = 0$, we say that the oracle is an *exact oracle.* Under the choice estimation setting, it can be difficult to acquire an exact oracle or even an oracle having an output noise $\mathcal{N}(\gamma)$ with small γ, because it frequently occurs that practitioners can only observe a small number of individual selections given a choice set and the average may fail to be a sufficiently good estimate of the true choice probabilities.

We adopt what Chierichetti et al. (2018) defined as a *sample oracle.* Each time we query x, an item $i \in [n]$ is sampled according to x and the oracle returns to us e_i. This has the natural interpretation of observing one purchase from an assortment in the choice estimation setting. Notice that since $\|e_i\|_2 \leqslant 1$ and $\|f(x)\|_2 \leqslant 1$ always, then by triangle inequality a sample oracle is also an oracle having output noise $\mathcal{N}(2)$.

4.2 Learning Sparse Functions

We present an extension of the Goldreich-Levin algorithm that learns any function $f\colon \{0,1\}^n \to \mathbb{R}^n$ that closely approximates a sparse function.

Theorem 5. *Assume $f\colon \{0,1\}^n \to \mathbb{R}^n$ is ε-close to a s-sparse function, then f is learnable with error ε with sample complexity $\mathcal{O}\Big(\frac{ns^3}{\varepsilon^3}\log(ns/\varepsilon)\Big)$ and time complexity $\mathsf{poly}(n, s, 1/\varepsilon)$.*

Before presenting the algorithm, let's first observe that the L_2 Fourier weights of an approximately sparse function must concentrate on a small set of Fourier bases with L_2 weights above a threshold (Claim 1). The primary challenge is to identify these bases.

Claim 1. *Assume $f : \{0,1\}^n \to \mathbb{R}^n$ is ε-close to an s-sparse function. Define $\mathcal{L} = \{S \subseteq [n]\colon \left\|\widehat{f}(S)\right\|_2^2 \geqslant \frac{\varepsilon}{s}\}$. Then $|\mathcal{S}| \leqslant 2s$ and $\left\|f - \sum_{S\in\mathcal{L}} \widehat{f}(S)\chi_S\right\|_2^2 \leqslant 2\varepsilon$.*

From now on, we will call $\mathcal{L}$ the set of *significant* Fourier bases.

Definition 9. *Given a predefined threshold $\tau > 0$, we say a Fourier base S is significant if $\hat{\|}\widehat{f}(S)\hat{\|}_2^2 \geqslant \tau$.*

At a high-level idea, the learning algorithm consists of two steps. The first step is to find all significant bases of the function (Lemma 6) and ignore all others, that is, find a list $\mathcal{L} = \{S \subseteq [n]\colon \left\|\widehat{f}(S)\right\|_2^2 \geqslant \tau\}$ for some appropriate threshold $\tau \geqslant 0$. Then, we use random samples to estimate every Fourier coefficient $\widehat{f}(S)$ of $S \in \mathcal{L}$ (Lemma 7), which, in turn, gives a good approximation of the original function. In the second step, we use Azuma-like concentration bounds (Hayes (2005)) for random vectors, while Chernoff-Hoeffding bounds are used in the one-dimensional case.

Lemma 6. *Given query access with output noise $\mathcal{N}(\gamma)$ to a function $f\colon \{0,1\}^n \to \mathbb{R}^n$ as well as a threshold $0 \leqslant \tau \leqslant 1$, there is a randomized algorithm that with high probability outputs a list $\mathcal{L}$ of subsets of $[n]$ satisfying*

1. *if $\left\|\widehat{f}(U)\right\|_2^2 \geqslant \tau$, then $U \in \mathcal{L}$, and*

2. *if $U \in \mathcal{L}$, then $\left\|\widehat{f}(U)\right\|_2^2 \geqslant \tau/4$.*

Furthermore, the running time is polynomial in $(n, 1/\tau)$*, and sample complexity* $\mathcal{O}\Big(\frac{(1+\gamma)n\log(n/\tau)}{\tau^3}\Big)$.

Lemma 7. *Given access to uniformly random samples of a function* $f\colon \{0,1\}^n \to \mathbb{R}^n$ *with output noise* $\mathcal{N}(\gamma)$*, there is a randomized algorithm which takes as input* $S \subseteq [n], 0 < \varepsilon, \delta \leqslant 1$ *and outputs an estimate* $\tilde{f}(S)$ *for* $\widehat{f}(S)$ *such that*

$$\left\|\tilde{f}(S) - \widehat{f}(S)\right\|_2^2 \leqslant \varepsilon$$

except with probability at most δ*. Furthermore, the number of samples needed is* $\frac{(2+\gamma)\log(4e^2/\delta)}{\varepsilon}$.

We provide the proofs of Theorem 5, Lemma 6, and Lemma 7 in the appendix.

ALGORITHM 1: Estimate

```
Input: Random access to $f$, confident parameter $\delta$, accuracy parameter $\varepsilon$, and a subset
        $S \subseteq [n]$
Output: $\tilde{f}(S)$ such that $\left\|\tilde{f}(S) - \widehat{f}(S)\right\|_2^2 \leqslant \varepsilon$
$\tilde{f}(S) = 0$, $m = \frac{\log(1/\delta)}{\varepsilon}$
for $i = 1$ to $m$ do
    Sample $x$ uniformly at random from $\{0,1\}^n$,
    $\tilde{f}(S) = \tilde{f}(S) + \frac{1}{m} \cdot f(x) \cdot \chi_S(x)$
Return $\tilde{f}(S)$
```

Comparison with a naive approach. A naive approach would learn function $f\colon \{0,1\}^n \to \mathbb{R}^n$ by learning each $f_i\colon \{0,1\}^n \to \mathbb{R}$ separately using Goldreich-Levin, where $f = (f_1, f_2, \ldots, f_n)$. This implies that the number of samples needed to approximately learn f with the same error ε and confident parameter δ is scaled up by a factor of n. We provide more details in the appendix.

4.3 Learning Sparse and Low-degree Functions

In this section, we focus on function mapping from $\{0,1\}^n$ to $\mathbb{R}^n$ that closely approximates a sparse and low-degree function. We will demonstrate that learning this specific class of functions can be achieved more efficiently. Specifically, our main result is as follows:

Theorem 6. *Assume* $f\colon \{0,1\}^n \to \mathbb{R}^n$ *is* ε*-close to a* s*-sparse, degree-*d *function, then* f *is learnable using* $\mathcal{O}\big((ds^2\log(n)\log(sd\log n))/\varepsilon^2\big)$ *samples with time complexity* $\mathsf{poly}(n, s, 1/\varepsilon)$.

Compared to Theorem 5, the sample complexity is diminished by a multiplicative factor of $n/(d\log n)$, a notable reduction, especially when $d\log n \ll n$. Our algorithm, akin to the learning algorithm in Section 4.2, comprises two steps. The first step involves the development of a new algorithm for identifying significant Fourier bases (Definition 9), while the second step remains consistent with the approach in Section 4.2.

4.3.1 Identifying Significant Fourier Bases

Recently, Amrollahi et al. (2019) presented an efficient algorithm that learns a sparse and low-degree function mapping from $\{0,1\}^n$ to $\mathbb{R}$. In particular, they showed that there exists a set of measurement vectors $\{v^{(i)}\}$ such that any $S \in \{0,1\}^n$ can be recovered from the linear measurements $\langle v^{(i)}, S\rangle$.

Lemma 8 (Amrollahi et al. (2019))**.** *For any integers* n, d*, there exists a set of measurements vectors* $\{v^{(i)}\}_{i=0}^m$ *for* $m = d\log n$ *such that, every* $S \in \{0,1\}^n$ *with* $|S| \leqslant d$ *can be recovered given the linear measurements* $\langle v^{(i)}, S\rangle$ *for all* $1 \leqslant i \leqslant m$.

Inspired by this idea and the GL/KM algorithm, we develop an efficient algorithm for identifying the set of all significant Fourier bases (Definition 9).

Let us introduce some notations before presenting the algorithm.

Definition 10. *Given a set B of Fourier bases, the L_2 Fourier weight of B is weight*$(B) = \sum\limits_{S \in B} \|\widehat{f}(S)\|_2^2$.

Let $V = \{v^{(1)}, v^{(2)}, \ldots, v^{(m)}\}$ be the set of vectors as defined in Lemma 8. For any matrix $\sigma \in \{0,1\}^{n \times k}$ and $b \in \{0,1\}^k$, we denote $H_\sigma^b = \{x \in \{0,1\}^n \colon \sigma^T \cdot x = b\}$. For each $v^{(i)} \in V$, we define $\sigma_i = \left[\dfrac{\sigma}{v_i}\right]$. So $H_{\sigma_i}^{b,j} = \{x \in \{0,1\}^n \colon \sigma_i^T \cdot x = (b,j)\}$.

```
ALGORITHM 2: Locate
  Input: Query access to $f$, degree $d$, sparsity parameter $s$, weight threshold $\tau$
  Output: A list $\mathcal{L} = \{S \subseteq [n] \colon \left\|\widehat{f}(S)\right\|_2^2 \geqslant \tau\}$
  $L = \emptyset$; $M_b = \emptyset$ for each $b \in \{0,1\}^k$
  Sample $k = \log s + 6$ vectors in $\{0,1\}^n$ uniformly randomly and form $\sigma$
  Get measurement vectors $V = \{v^{(1)}, v^{(2)}, ..., v^{(d \log n)}\}$
  for each $v^{(i)} \in V$ do
      Estimate weight($H_\sigma^b$) simultaneously with $\frac{\log(sd\log n)}{16\tau^2}$ samples for all $b \in \{0,1\}^k$
      Estimate weight($H_{\sigma_i}^{b,0}$) simultaneously with $\frac{\log(sd\log n)}{16\tau^2}$ samples for all $b \in \{0,1\}^k$
  for each $b \in \{0,1\}^k$ do
      if weight($H_\sigma^b$) $\geqslant \frac{\tau}{2}$ then
          for each $v^{(i)} \in V$ do
              if weight($H_{\sigma_i}^{b,0}$) $\geqslant \frac{\tau}{2}$ then
                  Set $\langle v^{(i)}, \cdot \rangle = 0$ and add the measurement to $M_b$
              else
                  Set $\langle v^{(i)}, \cdot \rangle = 1$ and add the measurement to $M_b$
      Recover some $S_b$ from measurements in $M_b$ and add $S_b$ to $\mathcal{L}$
  Return $\mathcal{L}$
```

The algorithm first selects $m = \mathcal{O}(\log(s))$ random vectors from $\{0,1\}^n$ and then concatenates them to form a matrix $\sigma \in \{0,1\}^{m*n}$. Then all Fourier bases are divided into 2^m buckets $B_j = \{S \in \{0,1\}^n | \sigma \cdot S = j\}$ for each $j \in \{0,1\}^m$. It will later be shown that if the entries of σ are generated uniformly i.i.d, with high probability bucket B_j contains only one significant Fourier base S such that $\left\|\widehat{f}(S)\right\|_2^2 \geqslant \tau$ and $\mathtt{weight}(B_j \setminus S) \leqslant \frac{\tau}{16}$. Next, concatenate σ with v_i to form a new matrix $\sigma_i = \left[\dfrac{\sigma}{v_i}\right]$. So we have formed a new bucket system with twice the number of buckets. That is, each bucket $B_j = \{S \in \{0,1\}^n \colon \sigma \cdot S = j\}$ is split into two buckets $B_{j,0} = \{S \in \{0,1\}^n | \sigma \cdot S = j, v^{(i)} \cdot S = 0\}$, $B_{j,1} = \{f \in \{0,1\}^n | \sigma \cdot S = j, v^{(i)} \cdot S = 1\}$. We estimate the weight of each bucket $B_{j,0}$. If $\mathtt{weight}(B_{j,0}) \geqslant \tau$, we can conclude that $\langle v^{(i)}, S \rangle = 0$ and $\langle v^{(i)}, S \rangle = 0$ otherwise. In this way, we can use estimations of L_2 Fourier weights to obtain all measurements and consequently recover S.

In the one-dimensional case of Amrollahi et al. (2019), the linear measurements $\langle v^{(i)}, S \rangle$ are estimated via a combination of the random hashing technique and the Walsh-Hadamard transform. To be more specific, given a bucket B, they obtain the corresponding linear measurement through

$sign(\sum_{S\in B}\widehat{f}(S))$, which can be computed using the Walsh-Hadamard transform. On the other hand, we compute these linear measurements through random hashing and estimation of the L_2 Fourier weights of the buckets. The main advantage of using L_2 estimations is that we can handle both multi-dimensional output space and our sample oracle query model. This is a crucial step in our analysis to extend Amrollahi et al. (2019) to our setting. It is worth noting that the GL/KM algorithm also utilizes L_2 estimations, but it does not employ either the random hashing technique or the low-degree structure

For the rest of this section, we shall prove that Locate (Algorithm 2) is correct and efficient.

We first introduce some notations about Fourier hashing.

Fourier Hashing. Let $f\colon \{0,1\}^n \to \mathbb{R}^n$ be a function. Let H be a subspace of $\{0,1\}^n$. Let $H^\perp := \{x \in \{0,1\}^n\colon x\cdot h = 0 \text{ for every } h \in H\}$ be the *dual* of H. Given $a \in H$, the *coset* $a + H$ is defined by $a + H := \{a + h\colon h \in H\}$. Then the L_2 Fourier weight of a coset $a + H$ is $\mathtt{weight}(a + H) := \sum_{S\in a+H}\left\|\widehat{f}(S)\right\|_2^2$

The following fact about the L_2 Fourier weight of a coset will be useful for our learning algorithm.

Claim 2. *For any function $f\colon \{0,1\}^n \to \mathbb{R}^n$, it holds that*

$$\sum_{S\in a+H}\left\|\widehat{f}(S)\right\|_2^2 = \mathop{\mathbb{E}}_{x\in\{0,1\}^n, z\in H^\perp} \chi_a(z)\langle f(x), f(x+z)\rangle.$$

It implies that the L_2 Fourier weight of a coset can be estimated using query access to f.

Claim 3. *For $0 < \varepsilon, \delta \leqslant 1$, the L_2 Fourier weight of the coset $a + H$ can be estimated within an error of ε with a probability of at least δ using $\frac{(1+\gamma)^2}{\varepsilon^2}$ queries access to f with noise $\mathcal{N}(\gamma)$.*

Hashing is an important technique in machine learning and theoretical computer science. In our context, a hashing function $h : \{0,1\}^n \to \{0,1\}^b$ is associated with a matrix σ where all entries are generated uniformly i.i.d. We would like to highlight some notable properties of hashing that are crucial for our algorithm. The first one is *pairwise independence.*

Fact 2. *Let σ be a random hashing matrix. Then for any $S, T \in \{0,1\}^n$,*

$$Pr(\exists\alpha \in \{0,1\}^n \ s.t.\ S \in \alpha + null(\sigma), T \in \alpha + null(\sigma)) = \frac{1}{2^m}.$$

Here 2^m gives the number of cosets. This essentially says that given any two vectors in $\{0,1\}^n$, they will be hashed to different cosets with high probability if the number of cosets is big enough. This allows us to obtain linear measurements for each of them without being influenced by the other.

Another crucial observation is that when σ is full rank, each coset corresponds to a bucket, and vice versa. We will demonstrate that if σ is generated with uniformly independent and identically distributed (i.i.d.) entries, full-rankness is satisfied with high probability. Our algorithm critically depends on this duality. The bucket structure enables us to execute splitting and consequently acquire linear measurements, while the coset structure facilitates efficient L_2 weight estimation.

The following claims are needed for the proof of Theorem 6.

Claim 4. *Let $f\colon \{0,1\}^n \to \mathbb{R}^n$ be ε-close to an s-sparse function. Let $S \subseteq [n]$ be such that $\left\|\widehat{f}(S)\right\|_2^2 \geqslant \frac{\varepsilon}{s}$. Suppose a sub-space H of co-dimension $16s$ is drawn uniformly at random and $S \in a + H$. Then the following events happen with a probability of at least $\frac{3}{5}$:*

(1) there does not exist another T with $\left\|\widehat{f}(T)\right\|_2^2 \geqslant \frac{\varepsilon}{s}$ such that $T \in a + H$.

(2) $\sum_{\substack{U \neq S, \\ U \in a+H}} \left\|\widehat{f}(U)\right\|_2^2 \leqslant \frac{\varepsilon}{16s}$.

Intuitively, this lemma conveys that significant Fourier bases do not interfere with each other, and insignificant ones within each coset do not overshadow the significant ones.

Additionally, we must demonstrate that σ_i is full-rank with high probability, thus ensuring the validity of the bucket-coset duality.

Claim 5. *Given a fixed set of vectors $v_1, v_2, ..., v_{dlogn} \in F_2^n$. Suppose we sample $2log(s)$ vectors uniformly randomly from F_2^n and form matrix $\sigma \in F_2^{n*s}$. Then it happens with negligible probability that for any v_i, σ_i is full-rank.*

Claim 6. *Suppose that f is ε-close to being s sparse. Let $\tau \geqslant \frac{\varepsilon}{s}$ be given. For any $S \subseteq [n]$ such that $\left\|\widehat{f}(S)\right\|_2^2 \geqslant \tau$. The* Locate *(Algorithm 2) can recover S with a probability of at least $\frac{1}{2}$. The sample complexity of the procedure is $\mathcal{O}\left(\frac{dlog(n)}{\tau^2}\right)$ and the time complexity is $\mathcal{O}\left(\frac{ndlog(n)}{\tau^2}\right)$.*

4.3.2 Learning Algorithm

We present our algorithm for learning sparse and low-degree functions as follows.

ALGORITHM 3: LearningSparseLowDegree

```
Input: Random access to $f$, confident parameter $\delta$, accuracy parameter $\varepsilon$
Output: $g\colon \{0,1\}^n \to \mathbb{R}^n$ such that $\|g - f\|_2 \leqslant \varepsilon$ with constant probability
$L = \emptyset$
for $i \leqslant log(s)$ do
    $L_i = \mathsf{Locate}(f, d, s, \varepsilon/s)$
    $L = L \cup L_i$
for each $S \in L$ do
    $\widehat{g}(S) = \mathsf{Estimate}(f, \delta, \varepsilon, S)$
Let $g = \sum_{S \in L} \widehat{g}(S)\chi_S(x)$.
Return $g$
```

Observe that as long as we repeat Locate (Algorithm 2) sufficiently number of times, we can recover all significant Fourier bases.

Proof of Theorem 6 . Let S be a given significant Fourier base and use $\mathbf{1}_{S,t}$ to denote the random vector indicating whether S is already successfully recovered at round t. Since i.i.d random vectors are produced for hashing in each round, $\mathbf{1}_{S,t}$ is independent across different rounds. By *Claim* 6, $\mathbf{1}_{S,t} \geqslant \frac{1}{2}$ for any t. Then by independence, the probability that S fails to be recovered and included in L for T rounds is $\frac{1}{2^T}$. And by union bounds, the probability that there exists one significant Fourier that fails to be recovered is $\frac{s}{2^T}$. Then setting $T = \log(\frac{s}{\delta})$ gives the desired confidence parameter. Then as for time/sample complexity, we plug $\tau = \varepsilon$ into Claim 6 and scale the results by $\log(\frac{s}{\delta})$ □

Now, leveraging the bounds provided in Theorem 4, it readily follows that the efficient estimation of any non-parametric choice model can be achieved using Fourier methods.

Theorem 7. *Let f be an arbitrary choice function. For every $0 < \varepsilon < 1$, f is learnable with error at most ε with time complexity $\mathcal{O}(n, 1/\varepsilon)$ and sample complexity $\tilde{\mathcal{O}}\left(\frac{\log n}{\varepsilon^4}\right)$.*

4.4 Learning Ranking Models by Uniform Sampling

This section is dedicated to illustrating that the Fourier spectrum of a ranking model exhibits a favorable inductive structure. This structure enables us to learn the function using the straightforward strategy of uniform sampling which, similar to Estimate (Algorithm 1), involves sampling uniformly random choice sets and averaging over the returned results.

Our first observation is on the Fourier coefficients of the sum of two ranking functions.

Lemma 9. *Given two ranking functions $f, f' : \{0,1\}^n \to \mathbb{R}^n$, $g = f + f'$ and $S \subseteq [n]$, it follows* $\|\widehat{g}(S)\|_1 = \left\|\widehat{f}(S)\right\|_1 + \left\|\widehat{f'}(S)\right\|_1$

Proof. Fix an item $j \in S$, then it suffices to show

$$|\widehat{g}(S)_j| = |\widehat{f}(S)_j| + |\widehat{f'}(S)_j|.$$

Recall that f can be written as a sum of indicator functions corresponding to paths. Among them, only $\mathbf{1}_{P_j}$, the path having j as the last stop, has a non-zero value at the j-th coordinate. Therefore, we have

$$\widehat{f}(S)_j = \widehat{\mathbf{1}_{P_i}}(S)_j.$$

Notice that on this path $P_j = \{x_1, x_2, ..., x_j\}$ with corresponding labels $i_1, i_2, ..., i_j$, we have $i_j = 1$ and $i_k = 0$ for any $k \neq j$. We can assume $S \subset P$, since otherwise, it must be the case that $\widehat{\mathbf{1}_{P_i}}(S)_j = 0$ and the claim trivially follows. Now we expand $\widehat{\mathbf{1}_{P_i}}(S)_j$ as in Section 3 and obtain

$$\widehat{\mathbf{1}_{P_i}}(S)_j = \begin{cases} (-1)^{|S|-1} \cdot \frac{1}{2^{|S|+1}}, & \text{if } i \in S \\ (-1)^{|S|} \cdot \frac{1}{2^{|S|+1}}, & \text{otherwise} \end{cases}$$

We can see that the sign depends only on whether $i \in S$ and the oddity of S. This is the same for both f and f' and therefore we have

$$sign(\widehat{f}(S)_j) = sign(\widehat{f'}(S)_j)$$

Then the claim follows.

□

An immediate consequence of the fixed sign patterns as seen in Lemma 9 is that the L_1 weight of a Fourier base can now be estimated easily as an expectation.

Claim 7. *Let f be a ranking model. Given $S \subseteq [n]$,*

$$\left\|\widehat{f}(S)\right\|_1 = \mathop{E}_{x \in \{0,1\}^n} \sum_{i=1}^{n} f(x)_i \cdot \chi_S(x) \cdot (-1)^{|S| + \mathbb{1}(i \in S)}$$

Claim 8. *For any ranking model f and $S \subseteq T \subseteq [n]$, $\left\|\widehat{f}(S)\right\|_1 \geqslant \left\|\widehat{f}(T)\right\|_1$.*

Proof. We will show that for each ranking r in the model, $\left\|\widehat{f}(S)\right\|_1 \leqslant \left\|\widehat{f}(T)\right\|_1$. Then the result follows from *Lemma* 9. Without loss of generality, assume the ranking is $\{1, 2, 3, ..., n\}$. Then given $j \in [n]$, we want to show $|\widehat{r}(S)_j| \geqslant |\widehat{r}(T)_j|$. There are three cases.

Case 1: j is smaller than the largest element of S. In this case, we have $\widehat{r}(S)_j = 0 = \widehat{r}(T)_j$.

Case 2: j is larger than the largest element of S but smaller than the greater element of T. In this case, we have $|\widehat{r}(S)_j| = \frac{1}{2^j}$ while $\widehat{r}(T)_j = 0$

Case 3: j is larger than the largest element of T. In this case, we have $\widehat{r}(S)_j = \frac{1}{2_j} = \widehat{r}(T)_j$.

Thus $|\widehat{r}(S)_j| \geqslant ||\widehat{r}(T)_j|$ in all cases and this completes the proof. □

Claim 8 naturally suggests a simple inductive procedure to recover Fourier bases with L_1 weight greater than a threshold τ and of degree at most d. We start from $\Psi_0 = \{\emptyset\}$, where $\Psi_i = \{S \subseteq [n] \colon \left\|\widehat{f}(S)\right\|_1 \geqslant \tau, |S| \leqslant i\}$. Then by Claim 8, if a Fourier base S' has an L_1 weight greater than τ, it must also be the case for all $S \subseteq S'$. Therefore, given Ψ_i, we can recover all bases in Ψ_{i+1} by looping through Ψ_i and evaluating the L_1 norm of all bases in the set $\{S \cup \{i\} | S \in \Psi_i, i \in [n]\}$ with uniformly random samples. It is easy to verify the correctness of this procedure and therefore we do not include a proof here.

ALGORITHM 4: RankingLocate

```
Input: Access of $m$ uniform samples to the ranking model $f$, degree bound $d$, weight
       threshold $\tau$
Output: A list $\Psi = \{S \subseteq [n] \colon \left\|\widehat{f}(S)\right\|_1 \geqslant \tau, |S| \leqslant d\}$
$\Psi_0 = \{\emptyset\}, \Psi_i = \emptyset$ for any $i \in [d]$, $\Psi = \Psi_0$
for $i = 0$ to $d - 1$ do
    for $S \in \Psi_i, j = 0$ to $n$ do
        $S' = S \cup \{j\}$
        $L_1(S') = 0$
        for $k = 0$ to $m$ do
            sample a uniform random $x \in \{0,1\}^n$
            $L_1(S') = L_1(S') + \frac{1}{m}\sum_{l=1}^{n} f(x)_i \cdot \chi_{S'}(x) \cdot (-1)^{|S'| + \mathbb{1}(l \in S')}$
        if $L_1(S') \geqslant \tau$ then
            add $S'$ to $\Psi_{i+1}$
    $\Psi = \Psi \cup \Psi_{i+1}$
Return $\Psi$
```

It then follows that ranking models can be learned from $O(\log(n), \frac{1}{\varepsilon})$ *uniform* samples with polynomial time complexity.

Theorem 8. *Let f be an arbitrary ranking model and $\varepsilon \in (0,1)$. Then f is learnable with error ε using* $\mathsf{poly}(\log(n), \frac{1}{\varepsilon})$ *uniform samples and* $\mathsf{poly}(n, \frac{1}{\varepsilon})$ *time complexity.*

Proof. By Theorem Theorem 4, it suffices to identify all Fourier bases with L_1 weight $\frac{\varepsilon}{\log(\frac{1}{\varepsilon})}$ and degree at most $\log(\frac{1}{\varepsilon})$. Therefore, it suffices to run RankingLocate (Algorithm 4) with $d = \log(\frac{1}{\varepsilon})$ and $\tau = \frac{\varepsilon}{\log(\frac{1}{\varepsilon})}$. There are at most $\frac{\log^2(\frac{1}{\varepsilon})}{\varepsilon}$ such Fourier bases. So we need to estimate the L_1 weights of at most $n \cdot \frac{\log^2(\frac{1}{\varepsilon})}{\varepsilon}$ Fourier bases. By Hoeffding inequality, it then requires $\mathsf{poly}(\log(n), \frac{1}{\varepsilon})$ to estimate the L_1 weights of all those bases within ε. And since updating the L_1 weight of one base takes $O(n)$ operations, the total time complexity is $\mathsf{poly}(n, \frac{1}{\varepsilon})$ and this completes the proof. □

5 Implementation and Numerical Experiments

5.1 Implementation for Ranking Models

The algorithms in Section 4.3 and Section 4.4 identify a set of Fourier bases $\mathcal{S}$ and coefficients incurring provably small L_2 error, which corresponds to the uniform distribution and squared loss.

In this section, we aim to demonstrate that the method can also be adapted to obtain superior performance under other distributions/metrics for the ranking model. Suppose we have already obtained a set of Fourier bases $\mathcal{S}$ and now we are given a new loss function L and a dataset $\mathcal{X}_{\mathcal{P}}$ generated from another distribution $\mathcal{P}$, a straightforward extension is to solve the following optimization problem:

$$\begin{aligned} \min_{\hat{f}_S} \quad & \sum_{(x,y)\in\mathcal{X}_{\mathcal{P}}} L\big(y, \sum_{\substack{S\in\mathcal{S},\\ i\in[n]}} \hat{f}_{S,i}\chi_{S,i}(x)\big) \\ \text{s.t.} \quad & \sum_{S\in\mathcal{S}, i\in[n]} |\hat{f}_{S,i}| \leqslant d \end{aligned}$$

where $d = \underset{S\in\mathcal{S}}{max}|S|$ and $\chi_{S,i}$ denotes the coordinate of vector χ_S corresponding to item i.

The L_1 norm constraint is cumbersome to work with. To avoid this problem, we recall from Section 4.4 that the Fourier coefficients of a ranking model have a fixed pattern.

Thus we now define $\bar{\chi}_{S,i}(x) = \begin{cases} (-1)^{|S|+1}\chi_{S,i}(x), & \text{if } i \in S \\ (-1)^{|S|}, & \text{otherwise} \end{cases}$. Then we can scale all coefficients to have sum 1 and rewrite the optimization problem as

$$\begin{aligned} \min_{\hat{f}_S,\, \hat{f}_0} \quad & \sum_{(x,y)\in\mathcal{X}_{\mathcal{P}}} L\Bigg(y, \quad \hat{f}_0 \cdot \overrightarrow{0} + \sum_{\substack{S\in\mathcal{S},\\ i\in[n]}} \hat{f}_{S,i} \cdot \frac{\bar{\chi}_{S,i}(x)}{d}\Bigg) \\ \text{s.t.} \quad & \sum_{\substack{S\in\mathcal{S},\\ i\in[n]}} \hat{f}_0 + \hat{f}_{S,i} = 1, \\ & \hat{f}_0, \hat{f}_S \geqslant 0 \quad \forall S \in \mathcal{S} \end{aligned}$$

where $d = \underset{S\in\mathcal{S}}{max}|S|$ and $\chi_{S,i}$ denotes the coordinate of χ_S corresponding to item i.

In case $\mathcal{S}$ is not sufficient, we can also start from $\mathcal{S}$ and adopt a column generation strategy similar to Jagabathula and Venkataraman (2022), whose objective is to find a distribution over rankings that best fits data. They observe that since the parameter space is a convex hull over rankings, at each point the steepest direction of descent at each point must correspond to a single ranking. Similarly, since we require the sum of coefficients in our formulation to be 1, the steepest direction corresponds to a single Fourier base. In practice, it is sufficient to just find a descending direction at each step rather than the steepest one and halt at a local minimum. We refer the readers to Jagabathula and Venkataraman (2022) for a more detailed discussion of column-generation strategies for non-parametric choice models.

5.2 Computational Studies on Simulated Data

We use the ranking model as the ground truth and follow a set-up similar to Francisca et al. (2022). There are n products and K major rankings, which correspond to major customer groups, with weights added up to 0.9. There are also 100 “noisy” rankings with a total weight of 0.1. We select $K = \{10, 15, 20, 25\}$ and $n = \{20, 25, 30, 35, 40, 45, 50\}$. Major rankings and “noisy rankings” are all generated uniformly and the coefficients are drawn from the Dirichlet distribution. The ”no-purchase” threshold is set to be $d = 4$. This is a reasonable choice since, as pointed out by Hauser (2014), there are many scenarios in which customers will only consider a small number of items before making the final decision.

To demonstrate the effectiveness of the Fourier-based approach across different distributions, for each n, K we create 5 product distributions over choice sets, where the mean probability of each item is generated uniformly in $[0, 1]$. For each of those distributions we generate 500,000 transactions (2,000 choice sets and 250 transactions for each) as training samples.

For each choice function $f_{n,K}$, we first use algorithm Algorithm 4 to obtain a set of Fourier bases $\mathcal{S}$ with L_1 weight greater than 0.003. We limit the number of queries to $100, 000$. Then we apply the procedure as outlined in section 5.1 on training samples generated from the choice set distributions.

Apart from the standard MNL model, we also compare our approach against two common heuristics used when estimating ranking-based models. The first one is random sampling (RS) as proposed by Farias et al. (2013), which first selects a uniformly random collection of rankings R and then finds the distribution on R that best fits the data. We set $|T| = 1000$. The second one is heuristic column generation (HCG) Chen and Mišić (2022), which starts from a small set of rankings (often single-item rankings) and then iteratively incorporates new ones into the set.

We use two popular metrics for comparisons: root mean squared error (RMSE) and mean absolute percentage error (MAPE). Given a choice set x, the ground truth choice function f, an estimator f', the $RMSE/MAPE$ errors of f' with respect to f at x are defined to be

$$RMSE(f'(x)) = \sqrt{\frac{\sum_{i=1}^{n}(f_i(x) - f'_i(x))^2}{n}}$$

$$MAPE(f'(x)) = \frac{1}{n} \cdot \sum_{i=1}^{n} \frac{|f_i(x) - f'_i(x)|}{f_i(x)}.$$

We then sample 400 assortments from the choice set distribution for testing and compute the average RMSE/MAPE error.

In Figure 5, we plot the RMSE/MAPE errors of the Fourier-based method, MNL, and the better of the two ranking heuristics as n, k varies. we can see that as n varies, the Fourier-based method consistently outperforms MNL and the two ranking heuristics. There are two noteworthy trends. First, we notice that as n increases, the performance of two ranking heuristics deteriorates in the sense that the error becomes close to the vanilla MNL model. This is not surprising since when the parameter size becomes big, it is much more likely that the randomly selected rankings for RS and the starting ranking set for HCG are from the true model. Second, the error incurred by the Fourier-based method is very stable as n stable, which shows that the performance of the Fourier-based method depends mainly on the inherent complexity of the choice function rather than on n.

5.3 Computational Studies on the Sushi Dataset

To illustrate that our method is effective with data exhibiting a high level of heterogeneity, characterized by a larger number of rankings, and where the set of rankings is not necessarily generated uniformly, we conduct computational studies on real data.

The dataset, as documented by Kamishima et al. (2005), consists of rankings provided by 5,000 Japanese respondents across 10 distinct types of sushi. This dataset has been used in a large number of papers, including Francisca et al. (2022), Desir et al. (2021), and Vitelli et al. (2014). It is noteworthy that, in addition to the rankings, the dataset records valuable demographic information of the respondents such as age, gender, and the prefectures (analogous to states in the U.S.) of their residence. This info enables practitioners to explore choice behaviors among different subgroups within the dataset.

Now we describe how to generate the choice functions of different subgroups from the dataset. Given a respondent i, we use f^i to denote the choice function corresponding to his/her ranking over sushi types. Then for a subset S of respondents, we give each respondent a unit weight and produce the aggregate choice function

$$f_S(x) = \frac{1}{|S|} \cdot \sum_{i \in S} f^i(x).$$

Our objective in this section is to learn the choice functions corresponding to the top 15 populous prefectures. Then we follow the same setup as in Section 5.2 except for one difference. Since the choice functions are generated from much more diverse populations, we double the size of the allowed queries for the Fourier algorithm to 200,000 and the number of transactions for each testing distribution to 1,000,000 (2,000 choice sets and 500 transactions for each).

Before heading to the result section, we want to note that the Sushi dataset overcomes the two limitations of the synthetic dataset as mentioned at the beginning of this section. On the one hand, the choice function corresponding to each prefecture exhibits a high degree of heterogeneity: the average number of rankings over the top 15 prefectures exceeds 200 and there are even 764 rankings in Tokyo, the most populous prefecture. On the other hand, the rankings in each prefecture are far from being uniformly generated. This is evident in the observation that certain sushi types, such as tuna and fat tuna, are significantly more likely to appear on the top of lists than other types, for instance, cucumber and egg.

In Table 1, we record the percentage drop of the Fourier-based method in RMSE/MAPE errors compared to the best of MNL and the two ranking heuristics. Across the prefectures, we can see that the Fourier-based method exhibits a consistent advantage over other methods. While we expected that the improvement would be more significant in the case of sparse rankings, we are surprised to find that the steepest drop in error occurs in Tokyo, where there are 764 rankings. These results demonstrate that the Fourier-based approach has the potential to be effective not only in simulated environments but also in real settings with heterogeneous customer preferences.

5.4 Computational Studies on the Electoral Reform Society Dataset

Given that there are only ten different types in the Sushi dataset, it is natural to wonder whether the Fourier-based method can also deliver good performance in real settings with more alternatives. To address this concern, we also conduct computational studies on the Electoral Reform Society (ERS) datasets.

The ERS datasets, which are publicly available on PrefLib Mattei and Walsh (2013), contain the results of 86 separate *ranked-choice* elections held by non-profit organizations, trade unions, and professional organizations. Each dataset records the partial rankings of voters over candidates in the corresponding election. Since our objective is to investigate the efficacy of the Fourier-based method in high-dimensional, real choice settings, we run experiments on the 5 datasets with more than 20 alternatives: ERS4, ERS5, ERS7, ERS16, and ERS78.

For each dataset, we produce the corresponding choice function in the same way as in Section 5.3. The rest of the setting is kept identical.

We again record the improvements in the percentage of the Fourier-based method over the three other approaches in Table 2. Overall the Fourier-based method delivers superior performance, with the reduction in MAPE error more significant. These results further establish discrete Fourier analysis as a practical tool for choice estimation.

6 Conclusion

Our paper presents a new approach to modeling and estimating non-parametric choice models through discrete Fourier analysis. The key advantage of our approach lies in its flexibility, accommodating a wide range of choice models. Additionally, we provide theoretical guarantees on both algorithm performance and sample complexity for learning. Computational studies demonstrate a substantial reduction in errors compared to existing methods.

As part of our future work, we aim to explore more specific classes of choice models to enhance sample and time complexity. Further computational studies are planned to validate and extend our findings.

Notes

1. The L_2 error based on uniform distribution is standard in the Fourier literature Amrollahi et al. (2019); O'Donnell (2014) etc. and has also been previously employed in management science literature (Chen and Mišić (2022)) to derive theoretical bounds. While our theoretical guarantee is obtained with L_2 error, our method can be combined with other heuristics to deliver good performance under other metrics and distributions, as demonstrated in Section 5.

Appendix

A Omitted Proofs in Section Section 4.2

A.1 Proof of Theorem 5

First, by Lemma 6, the algorithm runs the extended Goldreich-Levin algorithm on f with threshold $\tau = \frac{\varepsilon}{2s}$ to obtain a list L of significant Fourier bases. Next, approximate each $\widehat{f}(S)$ for $S \in L$ by $\tilde{f}(S)$ to within error α (to be chosen later) by Lemma 7. Define

$$g(x) = \sum_{S \in L} \tilde{f}(S)\chi_S(x).$$

We claim that $\|f - g\|_2^2 \leqslant \varepsilon$. We have

$$\begin{aligned}
\|f - g\|_2^2 &= \sum_{S \subseteq [n]} \left\|\widehat{f}(S) - \widehat{g}(S)\right\|_2^2 && \text{(Parseval's identity)}\\
&= \sum_{S \in L} \left\|\widehat{f}(S) - \widehat{g}(S)\right\|_2^2 + \sum_{S \notin L} \left\|\widehat{f}(S)\right\|_2^2 && (\widehat{g}(S) = 0 \text{ for } S \notin L)\\
&\leqslant \alpha^2 |L| + \sum_{S \notin L} \left\|\widehat{f}(S)\right\|_2^2
\end{aligned}$$

We upper bound the sum over $S \notin L$ by

$$\sum_{S \notin L} \left\|\widehat{f}(S)\right\|_2^2 \leqslant \max_{S \notin L} \left\|\widehat{f}(S)\right\|_2 \cdot \sum_{S \notin L} \left\|\widehat{f}(S)\right\|_2 \leqslant \frac{\varepsilon}{2s} \cdot \sum_{S \notin L} \left\|\widehat{f}(S)\right\|_1 \leqslant \frac{\varepsilon}{2s} \cdot s = \varepsilon/2.$$

Choosing $\alpha = \sqrt{1/(2\varepsilon|L|)}$ gives $\|f - g\|_2^2 \leqslant \varepsilon$.

Next, we compute the number of queries needed. By Claim 1, $|L| \leqslant 2s$. Finding the set L requires $\mathcal{O}\Big(\frac{ns^3}{\varepsilon^3} \log(ns/\varepsilon)\Big)$. For each $S \in L$, we approximate $\widehat{f}(S)$ to within α with except probability γ (chosen later). This requires $\mathcal{O}\Big(\frac{\log(1/\gamma)}{\alpha^2}\Big)$ samples. If we choose $\gamma = 1/10|L|$, then by union bound the probability that we estimate all Fourier coefficients in L to within accuracy α with except probability $1 - |L| \cdot \gamma = 9/10$, and the number of random samples needed is $\mathcal{O}\Big(\frac{s^2}{\varepsilon}\log(s^2/\varepsilon)\Big)$. Therefore, the sample complexity is $\mathcal{O}\Big(\frac{ns^3}{\varepsilon^3}\log(ns/\varepsilon)\Big) + \mathcal{O}\Big(\frac{s^2}{\varepsilon}\log(s^2/\varepsilon)\Big) = \mathcal{O}\Big(\frac{ns^3}{\varepsilon^3}\log(ns/\varepsilon)\Big)$, and the running time is polynomial in $(n, s, 1/\varepsilon)$.

A.2 Proof of Lemma 7

We shall use the following concentration bound for the proof.

Definition 11. *Let $X = (X_1, X_2, \ldots, X_m)$ be a sequence of random vectors such that $X_i \colon \Omega \to \mathbb{R}^d$, $X_0 = 0$, and for every $j < i$, $\mathbb{E}[\|X_i\|] < \infty$ and $\mathbb{E}[X_i | X_j] = X_j$. Then, we call X a weak martingale sequence.*

Theorem 9 (Hayes (2005))**.** *Let $X = (X_1, X_2, \ldots, X_m)$ be a weak martingale sequence taking value in $\mathbb{R}^d$ such that $X_0 = 0$ and for every i, $\|X_i - X_{i-1}\|_2 \leqslant 1$. Then for every $\varepsilon > 0$,*

$$\Pr[\frac{1}{m}\|X_m\|_2^2 \geqslant \varepsilon] \leqslant 2e^2 e^{-2m\cdot\varepsilon}.$$

Fact 3. *For any non-empty $S \subseteq [n]$, it holds that $\underset{x\in\{0,1\}^n}{E}\chi_S(x) = 0$.*

Proof of Lemma 7 . First, we show how to estimate $\tilde{f}(S)$ using query access to f without noise.

Recall that $\widehat{f}(S) = \mathbb{E}[f(x)\chi_S(x)]$. The main idea is to use m (chosen later) independent samples of f, say $f(x_1), f(x_2, \ldots, f(x_m)$, to empirically approximate $\widehat{f}(S)$.

Let $\tilde{f}(S) = \frac{1}{m}\sum_{i=1}^m f(x_i)\chi_S(x_i)$. Let $Y_i = f(x_i)\chi_S(x_i) - \widehat{f}(S)$. First, observe that $\mathbb{E}[Y_i] = 0$ for every i. Let $X_0 = 0$, $X_i = Y_1 + Y_2 + \ldots + Y_i$ for every $i > 0$. It is easy to see that, for every $j < i$, $\mathbb{E}[X_i | X_j] = X_j$. Therefore, $X = (X_0, X_1, \ldots, X_m)$ is a martingale sequence. Furthermore, we have

$$\begin{aligned}\|X_i - X_{i-1}\|_2 = \|Y_i\|_2 = \left\|f(x_i)\chi_S(x_i) - \widehat{f}(S)\right\|_2 &\leqslant \|f(x_i)\chi_S(x_i)\|_2 + \left\|\widehat{f}(S)\right\|_2 \\ &\leqslant \|f(x_i)\chi_S(x_i)\|_2 + \mathbb{E}\|f(x)\chi_S(x)\|_2 \\ &\leqslant \|f(x_i)\chi_S(x_i)\|_1 + \mathbb{E}\|f(x)\chi_S(x)\|_1 \\ &\leqslant 1 + 1 = 2,\end{aligned}$$

since $\|f(x)\chi_S(x)\|_1 = \|f(x)\|_1 \leqslant 1$ for every x. Applying Theorem 9 for the sequence $X/2 = (X_1/2, X - 2/2, \ldots, X_m/2)$, we have, for every $\varepsilon > 0$,

$$\Pr\left[\left\|\frac{1}{m}\sum_{i=1}^m f(x_i)\chi_S(x_i) - \widehat{f}(S)\right\|_2^2 \geqslant \varepsilon\right] \leqslant 4e^2 e^{-2m\cdot\varepsilon}.$$

Now, let $\tilde{f}(S) = \frac{1}{m}\sum_{i=1}^m f(x_i)\chi_S(x_i)$, and choose $m = \frac{2\log(4e^2/\delta)}{\varepsilon} = \mathcal{O}\Big(\frac{\log(1/\delta)}{\varepsilon}\Big)$, we have

$$\Pr\left[\left\|\tilde{f}(S) - \widehat{f}(S)\right\|_2^2 \geqslant \varepsilon\right] \leqslant \delta$$

Adding noise to the outputs. In this setting, on input $x \in \{0,1\}^n$, the output is $f(x) + \rho(x)$, where $\rho(x)$ such that $\mathbb{E}[\rho(x)] = 0$ and $\|\rho(x)\|_2^2$ is bounded by γ. For the sake of convenience, we will shorten $\rho(x_i)$ as ρ_i. Define $Y_i = (f(x_i) + \rho_i)\chi_S(x_i) - \widehat{f}(S)$, and the same for X_i. Then, it is still the case that X is a martingale sequence since $\mathbb{E}[\rho_i] = 0$.

$$\mathbb{E}[Y_i] = \mathbb{E}[(f(x_i) + \rho_i)\chi_S(x_i) - \widehat{f}(S)] = \mathbb{E}[f(x_i)\chi_S(x_i) - \widehat{f}(S)] + \mathbb{E}[\rho_i\chi_S(x_i)] = 0 + \mathbb{E}[\rho_i]\,\mathbb{E}[\chi_S(x_i)]$$

If $S \neq \emptyset$ then $\mathbb{E}[\chi_S(x_i)] = 0$. If $S = 0$, then $\mathbb{E}[Y_i] = \mathbb{E}[\rho_i] = 0$. So it always holds that $\mathbb{E}[Y_i] = 0$. The only difference from no noise case is that

$$\|X_i - X_{i-1}\|_2 = \|Y_i\|_2 \leqslant \|(f(x_i) + \rho_i)\chi_S(x_i)\|_2 + \left\|\widehat{f}(S)\right\|_2$$

is bounded by a larger number but smaller than $2 + \gamma$. Effectively, the number of samples needed is still $\frac{(2+\gamma)\log(4e^2/\delta)}{\varepsilon}$. □

Comparison with a naive approach. Suppose we learn $\widehat{f}(S)$ by approximating each coordinate separately. Let $f = (f_1, f_2, \ldots, f_n)$ where each $f_i \colon \{0,1\}^n \to \mathbb{R}$ for every $1 \leqslant i \leqslant n$. For every $S \subseteq [n]$, if we use m independent samples to estimate $\widehat{f}_i(S)$ by $\tilde{f}_i(S)$, then

$$\Pr\left[\left|\tilde{f}_i(S) - \widehat{f}_i(S)\right|^2 \geqslant \varepsilon\right] \leqslant 2e^{-2m\varepsilon/4}$$

A naive approach would require that $\left|\tilde{f}_i(S) - \widehat{f}_i(S)\right|^2 \geqslant \varepsilon/n$ for every i to ensure that $\left\|\tilde{f}(S) - \widehat{f}(S)\right\|_2^2 \geqslant \varepsilon$. Therefore, we have

$$\begin{aligned}
\Pr\left[\left\|\tilde{f}(S) - \widehat{f}(S)\right\|_2^2 \geqslant \varepsilon\right] &\leqslant \Pr\left[\exists i\colon \left|\tilde{f}_i(S) - \widehat{f}_i(S)\right|^2 \geqslant \varepsilon/n\right] \\
&\leqslant \sum_{i=1}^{n} \Pr\left[\left|\tilde{f}_i(S) - \widehat{f}_i(S)\right|^2 \geqslant \varepsilon/n\right] && \text{(union bound)} \\
&\leqslant \sum_{i=1}^{n} 2e^{-2m\varepsilon/4n} \\
&= 2n \cdot e^{-2m\varepsilon/4n}
\end{aligned}$$

To achieve δ confidence, one needs to choose $m = \mathcal{O}\left(\frac{n\log(1/\delta)}{\varepsilon}\right)$.

A.3 Proof of Lemma 6

This section extends the Golreich-Levin/KM algorithm to vector-valued functions. To prove the theorem we need the following results.

Theorem 10 (Hoeffding's Bound). *Let $X_1, X_2, \ldots, X_n$ be independent variables such that $X_i \in [a, b]$. Let $\bar{X} = \mathbb{E}[X_i]$. Then for any $\varepsilon \geqslant 0$, the following bound holds*

$$\Pr[|\bar{X} - \mathbb{E}[\bar{X}]| \geqslant \varepsilon] \leqslant 2\exp(-2n\varepsilon^2/(b-a)^2).$$

Proposition 2. *Let $f\colon \{0,1\}^n \to \mathbb{R}^d$ be a function in $\mathcal{H}$. Let $J \subseteq [n]$. Given query access to f with some random noise σ such that $\mathbb{E}[\sigma] = 0$ and $\|\sigma\|_2 \leqslant \beta$ and only depends on the current queried point x. For any $S \subseteq J$, the quantity $\sum_{T\subseteq\bar{J}}\left\|\widehat{f}(S \cup T)\right\|_2^2 = \mathbb{E}_z\left\|\widehat{f_{J|z}}(S)\right\|_2^2$ can be estimated with error at most ε (except with probability at most δ) using $\frac{2(1+\beta)^2\log(2/\delta)}{\varepsilon^2}$ queries.*

Proof. Treating each coordinate separately for each restriction function and then adding things up gives us the following formula.

$$\begin{aligned}
\sum_{T\subseteq \bar{J}} \left\|\widehat{f}(S\cup T)\right\|_2^2 &= \mathop{\mathbb{E}}_{z}\left\|\widehat{f_{J|z}}(S)\right\|_2^2 \\
&= \mathop{\mathbb{E}}_{z}\sum_{i=1}^{n} \widehat{f_{J|z}}(S)_i^2 \\
&= \sum_{i=1}^{n} \mathop{\mathbb{E}}_{z} \widehat{f_{J|z}}(S)_i^2 \\
&= \sum_{i=1}^{n} \mathop{\mathbb{E}}_{z} \mathop{\mathbb{E}}_{y}[f(y,z)_i \chi_S(y)]^2 \\
&= \sum_{i=1}^{n} \mathop{\mathbb{E}}_{z} \mathop{\mathbb{E}}_{y,y'}[f(y,z)_i \ \chi_S(y) \ f(y',z)_i \ \chi_S(y')] \\
&= \mathop{\mathbb{E}}_{z} \mathop{\mathbb{E}}_{y,y'}\left[\sum_{i=1}^{n} f(y,z)_i \ \chi_S(y) \ f(y',z)_i \ \chi_S(y')\right] \\
&= \mathop{\mathbb{E}}_{z\sim\{0,1\}^{\bar{J}}}\left[\mathop{\mathbb{E}}_{y,y'\sim\{0,1\}^{J}} f(y,z)\cdot f(y',z)\chi_S(y)\chi_S(y')]\right]
\end{aligned}$$

Next, we will show that $f(y,z)\cdot f(y',z)\chi_S(y)\chi_S(y')$ is a bounded real value.

$$\begin{aligned}
\left|f(y,z)\cdot f(y',z)\chi_S(y)\chi_S(y')\right| &= \left|f(y,z)\cdot f(y',z)\right| && (\chi_S \text{ is } \{0,1\}\text{-valued}) \\
&\leqslant \|f(y,z)\|_2 \left\|f(y',z)\right\|_2 && \text{(Cauchy-Schwartz inq.)} \\
&\leqslant 1
\end{aligned}$$

Thus, $f(y,z)\cdot f(y',z)\chi_S(y)\chi_S(y')$ is a bounded random variable that can be sampled by using query access to f. By Hoeffding's bound, we can estimate $\mathbb{E}_z\left\|\widehat{f_{J|z}}(S)\right\|_2^2$ with error at most ε except with probability at most δ using $2\frac{\log(2/\delta)}{\varepsilon^2}$ queries access to f.

Adding noise. Even in the presence of the noise, it still holds that

$$\mathop{\mathbb{E}}_{z,y,y'}(f(y,z)+\sigma)\cdot(f(y',z)+\sigma')\chi_S(y)\chi_S(y') = \sum_{T\subseteq\bar{J}}\left\|\widehat{f}(S\cup T)\right\|_2^2.$$

This is an easy consequence of the law of iterated expectation. Once we fix y, y', z, the only random variables are σ, σ' with $E[\sigma] = E[\sigma'] = 0$. Given that y, y', z are now constants, we only need to worry about $E[\sigma\cdot\sigma']$. Since σ, σ' are mutually independent, it follows that $E[\sigma\cdot\sigma'] = 0$ and the desired equality is obtained.

The random variable $(f(y,z)+\sigma)\cdot(f(y',z)+\sigma')\chi_S(y)\chi_S(y')$ is bounded if $\|\sigma\|_2, \|\sigma'\|_2$ are bounded. Once again, it only changes the constant in the big O notation.

$$\begin{aligned}
\left|(f(y,z)+\sigma)\cdot(f(y',z)+\sigma')\chi_S(y)\chi_S(y')\right| &= \left|(f(y,z)+\sigma)\cdot(f(y',z)+\sigma')\right| \\
&\leqslant \|f(y,z)+\sigma\|_2\left\|f(y',z)+\sigma'\right\|_2 \\
&\leqslant (\|f(y,z)\|+\|\sigma\|_2)(\left\|f(y',z)\right\|+\left\|\sigma'\right\|_2) \\
&\leqslant (1+\beta)^2
\end{aligned}$$

By Hoeffding's bound, we can estimate $\mathbb{E}_z\left\|\widehat{f_{J|z}}(S)\right\|_2^2$ with error at most ε except with probability at most δ using $\frac{2(1+\beta)^2\log(2/\delta)}{\varepsilon^2}$ queries access to f. □

Proof of Lemma 6 . We begin with an overview of how the algorithm works.

Initialization: all 2^n possible subsets of $[n]$ are put in a single bucket. The algorithm then repeats the following loop:

1. Select any bucket $\mathcal{B}$ (containing 2^m sets for some $m \geqslant 1$).

2. Split $\mathcal{B}$ into two buckets $\mathcal{B}_1$ and $\mathcal{B}_2$ of 2^{m-1} sets each.

3. Estimate $\sum_{U\in\mathcal{B}_i}\left\|\widehat{f}(U)\right\|_2^2$ for each $i = 1, 2$.

4. Discard $\mathcal{B}_1$ or $\mathcal{B}_2$ if its weight estimate is at most $\tau/2$.

The algorithm stops when all buckets contain only 1 set, and then outputs the list of these sets.

Bucket system. Next, we describe the bucketing system. For $1 \leqslant k \leqslant n$ and $S \subseteq [n]$, we define

$$\mathcal{B}_{k,S} = \{S \cup T \colon T \subseteq \{k+1, k+2, \ldots, n\}\}.$$

The initial bucket is $\mathcal{B}_{0,\emptyset}$. The buckets at the end of the algorithm have the form $\mathcal{B}_{k,S} = \{S\}$. Note that $|\mathcal{B}_{k,S}| = 2^{n-k}$. The algorithms always split the bucket $\mathcal{B}_{k,S}$ into two buckets $\mathcal{B}_{k+1}, S$ and $\mathcal{B}_{k+1,S\cup\{k+1\}}$. The weight of bucket $\mathcal{B}_{k,S}$ is exactly $\sum_{T\subseteq\{k+1,\ldots,n\}}\left\|\widehat{f}(S\cup T)\right\|_2^2$.

Correctness. Any set U with $\left\|\widehat{f}(U)\right\|_2^2 \geqslant \tau$ is never be discarded since it always contributes at least $\tau \geqslant \tau/2$ to the bucket it's in. On the other hand, any set U with $\left\|\widehat{f}(U)\right\|_2^2 \leqslant \tau/2$ is always be discarded since $\tau/4 \leqslant \tau/2$. Therefore, as long as the estimation is accurate within $\pm\tau/4$, the algorithm is correct.

Running time. Any non-discarded bucket has a weight of at least $\tau/2$ even assuming that the weight estimate is only accurate within $\pm\tau/4$. The total weight of all non-discarded buckets is at most $\sum_{S\in\subseteq[n]}\left\|\widehat{f}(S)\right\|_2^2 \leqslant 1$. Thus, at any time, there are at most $2/\tau$ non-discarded buckets. Note that each bucket can be split at most n times. Therefore, the algorithm repeats at most $2n/\tau$ loops. In each loop, the algorithm will need to estimate two buckets to accuracy $\pm\tau/4$ and confidence $1-\delta$, each requires $\mathcal{O}\big(\log(1/\delta)/\tau^2\big)$ queries. The algorithm overall needs to make at most $4n/\tau$ weighings. This implies that we need $\mathcal{O}\big(n\log(1/\delta)/\tau^3\big)$ queries. By union bound, the probability that all weighings are accurate to within $\tau/4$ is at least $1 - 4n/\tau\cdot\delta$. Choose $\delta = \tau/40n$, we get that the probability is at least $9/10$. □

B Omitted Proof in Section 4.3

B.1 Proof of Claim 1

Suppose f is ε-close to an s-sparse function g. Then define $\mathcal{L}_g = \{S \in [n] \big| \widehat{g}(S) \neq 0\}$. Obviously $|\mathcal{S} \cap \mathcal{L}_g| \leqslant s$. Now consider $\mathcal{S} \cap \mathcal{L}_g^{\complement}$. Then by Parseval,

$$\sum_{S\in\mathcal{L}\cap\mathcal{L}_g^{\complement}}\left\|\widehat{f}(S)\right\|_2^2 = \sum_{S\in\mathcal{L}\cap\mathcal{L}_g^{\complement}}\left\|\widehat{f}(S)-\widehat{g}(S)\right\|_2^2 \leqslant \sum_{S\in[n]}\left\|\widehat{f}(S)-\widehat{g}(S)\right\|_2^2 \leqslant \varepsilon.$$

Then it follows $\left|\mathcal{L}\cap\mathcal{L}_g^{\complement}\right| \leqslant \frac{\varepsilon}{\varepsilon/s} = s$ and $|\mathcal{L}| \leqslant 2s$.

Now we need to give the accuracy of the approximation. We can easily calculate

$$\sum_{S\in\mathcal{L}_g\cap\mathcal{L}^{\complement}} \left\|\widehat{f}(S)\right\|_2^2 \leqslant \frac{\varepsilon}{s}\cdot s = \varepsilon$$

$$\sum_{S\in\mathcal{L}_g^{\complement}\cap\mathcal{L}^{\complement}} \left\|\widehat{f}(S)\right\|_2^2 \leqslant \sum_{S\in\mathcal{L}_g^{\complement}} \left\|\widehat{f}(S)\right\|_2^2 = \sum_{S\in\mathcal{L}\cap\mathcal{L}_g^{\complement}} \left\|\widehat{f}(S) - g(S)\right\|_2^2 \leqslant \varepsilon.$$

Now merging the two inequalities above and applying Parseval yields

$$\left\|f - \sum_{S\in\mathcal{S}}\widehat{f}(S)\chi_S\right\|_2^2 = \sum_{S\in\mathcal{L}^{\complement}} \left\|\widehat{f}(S)\right\|_2^2 \leqslant 2\varepsilon$$

and the proof is complete.

B.2 Proof of Claim 3

Proof. Now noise is considered. We need to estimate the weight with

$$\mathop{E}_{x,z,\rho,\rho'}\left[\chi_a(z)\langle f(x)+\rho, f(x+z)+\rho'\rangle\right].$$

where $x\sim\{0,1\}^n, z\in H^{\perp}$. Using the fact that ρ,ρ' are mutually independent and independent from $x, x+z$, we can obtain

$$\mathop{E}_{x,z,\rho,\rho'}\left[\chi_a(z)\langle f(x)+\rho, f(x+z)+\rho'\rangle\right] = \mathop{E}_{x,z,\rho,\rho'}\left[\chi_a(z)\langle f(x), f(x+z)\rangle\right] = \texttt{weight}(a+H).$$

Thus the estimation is unbiased.

Now we show that the random variable is bounded. By triangle inequality,

$$\|f(x)+\rho\|_2 \leqslant \|f(x)\|_2 + \|\rho\|_2 \leqslant 1+\gamma$$

Then we can apply the Cauchy-Schwartz inequality to bound the inner product

$$\begin{aligned}\left|\chi_a(z)\langle f(x)+\rho, f(x+z)+\rho'\rangle\right| &= \left|\langle f(x)+\rho, f(x+z)+\rho'\rangle\right| \\ &\leqslant \|f(x)+\rho\|_2\|f(x)+\rho\|_2 \\ &\leqslant (1+\gamma)^2\end{aligned}$$

Using this bound with Hoeffding will yield the desired result. □

B.3 Proof of Claim 4

First show (1). Since hashing is pairwise independent, the probability of S hashed into the same bucket with another significant Fourier base is $\frac{1}{2^{\log s+6}} = \frac{1}{64s}$. Since there are a total of s significant Fourier bases, by union bound, collision happens with probability at most $\frac{1}{64}$.

Now let us show (2) We need to analyze the insignificant Fourier bases hashed into $a+H$. Then by Parseval and claim Claim 1, we have

$$\sum_{U\colon \|\widehat{f}(U)\|_2^2\leqslant\frac{\varepsilon}{s}} \left\|\widehat{f}(U)\right\|_2^2 = \left\|f - \sum_{S\in\mathcal{S}}\widehat{f}(S)\chi_S\right\|_2^2 \leqslant \varepsilon.$$

Let $\mathcal{V} = \{U \colon U \in a+H, \left\|\widehat{f}(U)\right\|_2^2 \leqslant \varepsilon/s\}$. We can calculate

$$\begin{aligned}
\mathbb{E}\left[\sum_{U\in\mathcal{V}} \left\|\widehat{f}(U)\right\|_2^2\right] &= E\left[\sum_{U\colon \|\widehat{f}(U)\|_2^2 \leqslant \frac{\varepsilon}{s}} \left\|\widehat{f}(U)\right\|_2^2 \cdot 1(U \in a+H)\right] \\
&= \sum_{U\colon \|\widehat{f}(U)\|_2^2 \leqslant \frac{\varepsilon}{s}} \left\|\widehat{f}(U)\right\|_2^2 \cdot E\left[1(U \in a+H)\right] \\
&= \sum_{U\colon \|\widehat{f}(U)\|_2^2 \leqslant \frac{\varepsilon}{s}} \left\|\widehat{f}(U)\right\|_2^2 \cdot \Pr(U \in a+H) \\
&\leqslant \frac{1}{64s} \cdot \sum_{U\colon \|\widehat{f}(U)\|_2^2 \leqslant \frac{\varepsilon}{s}} \left\|\widehat{f}(U)\right\|_2^2 \\
&\leqslant \frac{1}{64s} \cdot \varepsilon
\end{aligned}$$

Then it follows from Markov's inequality that

$$\Pr\left[\sum_{U\in\mathcal{V}} \left\|\widehat{f}(U)\right\|_2^2\right] \leqslant \frac{1}{4}.$$

Now by union bound $(1), (2)$ fails with probability at most $\frac{2}{5}$.

B.4 Proof of Claim 5

Let us only consider a single v_i and then the lemma follows from union bound. If we select a h_1 uniformly randomly from F_2^n, then since $|span(v)| = 2$ and the probability of $h_1 \notin span(v)$ is $\frac{2^n - 2^1}{2^n}$. Now we select h_2 uniformly from F_2^n. Then since $|span(v, h_1)| = 2^2$, the probability of $h_2 \notin span(v, h_1)$ is $\frac{2^n-2^2}{2^n}$. Continuing in this fashion, we can calculate the probability that σ_i is full-rank is

$$\frac{2^n - 2^1}{2^n} \cdot \frac{2^n - 2^2}{2^n} \cdot \cdots \cdot \frac{2^n - 2^{log(s)+10}}{2^n},$$

which is extremely close to 1 since $2^n \gg 2^{log(s)+10}$ when n is big enough.

B.5 Proof of Claim 6

Let us first show correctness. For the recovery to be successful, three conditions must hold. Two conditions given in Claim 4 hold with a probability of at least $\frac{3}{5}$. Another condition is that σ_i is full rank for every i so that we can estimate $\texttt{weight}(H_{\sigma_i}^{b,0}), \texttt{weight}(H_{\sigma_i}^{b,1})$. Claim 5 has shown that this holds with extremely high probability.

Suppose the three conditions hold. Then S is the single Fourier base with L_2 weight $\geqslant \tau$ hashed into some bucket H_σ^b. If $< v, S >= 0$, then $S \in H_{\sigma_i}^{b,0}$ and therefore $\texttt{weight}(H_{\sigma_i}^{b,0}) \geqslant \tau$. And since with high probability, our estimation of the weight is accurate within $\frac{\tau}{4}$, the estimation is greater than $\frac{\tau}{2}$. If $< v, S >= 1$, then $H_{\sigma_i}^{b,0}$ contains only insignificant Fourier bases and the sum of weight cannot be more than $\frac{\tau}{16}$. Then the estimation must be smaller than $\frac{\tau}{2}$. In this way, the algorithm can always correctly obtain all measurements $< v, S >$.

The sample complexity and time complexity are straightforward. We need $d \log n$ linearly measurements for successful recovery, and for each measurement $\frac{\log(sd \log n)}{16\tau^2}$ queries are used to obtain the desired accuracy. Therefore, the sample complexity is $\frac{d \log(n) \log(sd \log n)}{16\tau^2}$. And the time complexity is scaled by a n since we need to perform tasks such as calculating the inner product between two vectors, which takes time n.

References

Amrollahi, A., A. Zandieh, M. Kapralov, and A. Krause (2019). Efficiently learning fourier sparse set functions. In *NeurIPS*, pp. 15094–15103.

Ben-Akiva, M. and M. Bierlaire (1999). Discrete choice methods and their applications to short term travel decisions. In *Handbook of transportation science*, pp. 5–33. Springer.

Blanchet, J., G. Gallego, and V. Goyal (2016). A markov chain approximation to choice modeling. *Operations Research 64*(4), 886–905.

Block, H. and J. Marshak (1959). Random orderings and stochastic theories of response. Technical report, Cowles Foundation for Research in Economics, Yale University.

Blum, A. (1994). Relevant examples and relevant features: Thoughts from computational learning theory. In *AAAI Fall Symposium on 'Relevance*, Volume 5, pp. 1.

Chen, N., G. Gallego, and Z. Tang (2019, August). The use of binary choice forests to model and estimate discrete choices. *ArXiv e-prints*.

Chen, Y.-C. and V. V. Mišić (2022). Decision forest: A nonparametric approach to modeling irrational choice. *Management Science*.

Chierichetti, F., R. Kumar, and A. Tomkins (2018). Discrete choice, permutations, and reconstruction. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 576–586.

Desir, A., V. Goyal, S. Jagabathula, and D. Segev (2021). Mallows-smoothed distribution over rankings approach for modeling choice. *Operation Research 69*(4), 1206–1227.

Farias, V. F., S. Jagabathula, and D. Shah (2013). A nonparametric approach to modeling choice with limited data. *Management science 59*(2), 305–322.

Francisca, S., N. Golrezaei, E. Emamjomeh-Zadeh, and D. Kempe (2022, August). Active learning for non-parametric choice models. *ArXiv e-prints*.

Goldreich, O. and L. A. Levin (1989). A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pp. 25–32.

Hauser, J. (2014). Consideration-set heuristics. *Journal of Business Research 67*(8), 1688–1699.

Haviv, I. and O. Regev (2017). The restricted isometry property of subsampled fourier matrices. In *Geometric aspects of functional analysis*, pp. 163–179. Springer.

Hayes, T. P. (2005). A large-deviation inequality for vector-valued martingales. *Combinatorics, Probability and Computing*.

Jagabathula, S. and A. Venkataraman (2022). Nonparametric estimation of choice models. In X. Chen, S. Jasin, and C. Shi (Eds.), *The Elements of Joint Learning and Optimization in Operations Mangement*, pp. 177–209. Springer.

Kamishima, T., H. Kazawa, and S. Akaho (2005). supervised ordering-an empirical survey. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pp. 4–.

Kolountzakis, M. N., E. Markakis, and A. Mehta (2005). Learning symmetric k-juntas in time nˆ o (k). *arXiv preprint math/0504246*.

Kushilevitz, E. and Y. Mansour (1993). Learning decision trees using the fourier spectrum. *SIAM Journal on Computing 22*(6), 1331–1348.

Linial, N., Y. Mansour, and N. Nisan (1993). Constant depth circuits, fourier transform, and learnability. *Journal of the ACM (JACM) 40*(3), 607–620.

Luce, R. (1959). *Individual choice behavior: A theoretical analysis.* Wiley.

Mattei, N. and T. Walsh (2013). Preflib: A library for preferences http://www.preflib.org. In *Algorithmic Decision Theory*, pp. 259–270.

McFadden, D. et al. (1973). Conditional logit analysis of qualitative choice behavior.

Mossel, E., R. O'Donnell, and R. P. Servedio (2003). Learning juntas. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pp. 206–212.

O'Donnell, R. (2014). *Analysis of boolean functions.* Cambridge University Press.

Stobbe, P. and A. Krause (2012). Learning fourier sparse set functions. In *Artificial Intelligence and Statistics*, pp. 1125–1133. PMLR.

Vitelli, V., O. Sorensen, M. Crispino, A. Frigessi, and E. Arjas (2014). Probabilistic preference learning with the mallows rank model. *Foundations and Trends® in Machine Learning 18*(1), 5796 – 5844.

Tables

Prefecture	RMSE Improvements	MAPE Improvements
Tokyo	39.1	34.06
Osaka	27.18	24.53
Kanagawa	26.04	21.81
Aichi	7.99	7.01
Hyogo	22.45	18.24
Saitama	40.02	34.52
Hokkaido	30.69	28.06
Chiba	18.7	19.46
Fukuoka	38.17	30.91
Shizuoka	20.7	14.69
Kyoto	22.07	30.49
Hiroshima	20.09	19.61
Niigata	38.93	32.34
Ibaraki	38.05	32.17
Okayama	27.7	28.79

Table 1: Percentage Improvements of the Fourier-based method over MNL and ranking heuristics as measured in RMSE and MAPE for the sushi dataset

Dataset	RMSE Improvements	MAPE Improvements
ERS4	40.79	52.29
ERS5	4.86	20.33
ERS7	39.28	47.73
ERS16	15.71	29.01
ERS78	0.29	9.9

Table 2: Percentage improvements of the Fourier-based method over MNL, HCG and RS measured in RMSE and MAPE for ERS datasets

Figures

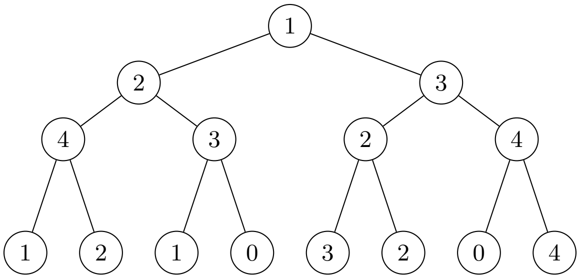

Figure 1: An example of a decision tree as formulated in Chen and Mišić (2022)

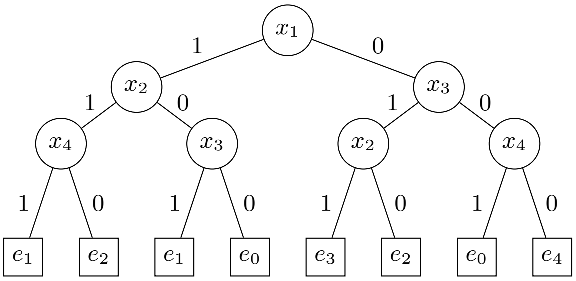

Figure 2: Our formulation of the decision forest

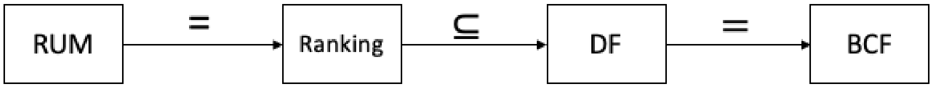

Figure 3: Relationships between the choice models mentioned in Section 2.1

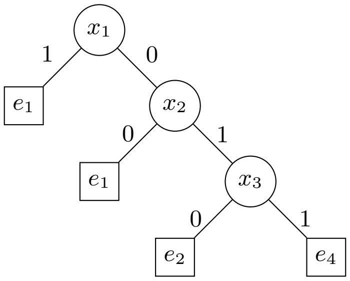

Figure 4: An example of a binary choice tree

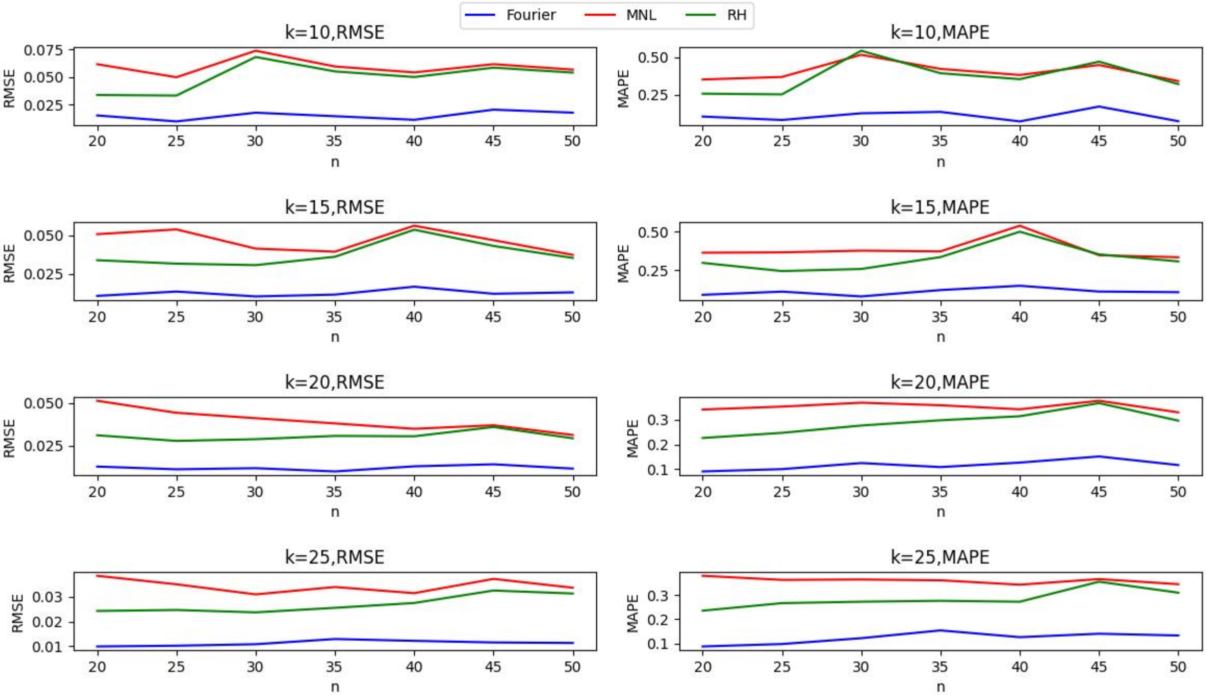

Figure 5: Comparison of the Fourier-based method with MNL and the best of HCG/RS for $k = \{10, 15, 20, 25\}$ and $n = \{20, 25, 30, 35, 40, 45, 50\}$