Learning Mixed Multinomial Logit Models

by

Yiqun Hu

B.S., University of Michigan - Ann Arbor (2013)
S.M., Massachusetts Institute of Technology (2017)

Submitted to the Center for Computational Science and Engineering
and Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Computational Science and Engineering
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
September 2022

Author . . . .
Center for Computational Science and Engineering
and Department of Civil and Environmental Engineering
August 12, 2022

Certified by. . . . .
David Simchi-Levi
Professor, Institute of Data, Systems, and Society,
Department of Civil and Environmental Engineering,
Operations Research Center
Thesis Supervisor

Accepted by . . . .
Colette Heald
The Germeshausen Professor,
Department of Civil and Environmental Engineering
Chair, Graduate Program Committee

Accepted by . . . .
Youssef Marzouk
Professor, Aeronautics and Astronautics
Co-director, MIT Center for Computational Science and Engineering

Learning Mixed Multinomial Logit Models

by

Yiqun Hu

Submitted to the Center for Computational Science and Engineering
and Department of Civil and Environmental Engineering
on August 12, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computational Science and Engineering

Abstract

Multinomial logit (MNL) model is widely used to predict the probabilities of different outcomes. However, standard MNL model suffers from several issues, including but not limited to heterogeneous population, the restricted independence of irrelevant alternative (IIA) assumption, insufficient model capacity, etc. To alleviate these issues, mixed multinomial logit (MMNL) models were introduced. MMNL models are highly flexible. McFadden and Train [2000] showed that it can approximate any random utility based discrete choice models to arbitrary degree of accuracy under appropriate assumptions. In addition, it removes other limitations of standard MNL models, including lifting the IIA assumption, allowing correlation in unobserved utility factors overtime, and most importantly, reducing the chance of model misspecification when modeling real world applications where the data composition is often found to be heterogeneous.

Despite its importance and versatility, the study on the learning theory of MMNL is limited and learning MMNL models remains an open research topic. In this thesis, we will tackle this learning problem from two different perspectives. First, inspired by the recent work in Gaussian Mixture Models (GMM), we aim to explore the polynomial learnability of MMNL models from a theoretical point of view. Next, we present an algorithm that is designed to be more applicable and utilizes the rich source of data available in the modern digitalization era, yet still yielding ideal statistical properties of the estimators.

Chapter 2 studies the polynomial learnability of MMNL models with a general K number of mixtures. This work aims to extend the current results that only apply to 2-MNL models. We analyze the existence of ϵ-close estimates using tools from abstract algebra and will show that there exists an algorithm that can learn a general K-MNL models with probability at least $1-\delta$, if identifiable, using polynomial number of data samples and polynomial number of operations (in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$), under some reasonable assumptions.

In Chapter 3, motivated by the Frank-Wolfe (FW) algorithm, we propose a framework that learns both mixture weights and component-specific logit parameters with provable convergence guarantees for arbitrary number of mixtures. Our algorithm utilizes historical choice data to generate a set of candidate choice probability vectors, each being ϵ-close to the ground truth with high probability. The convex hull of this set forms a shrunken feasible region with desired properties to the linear subproblems in FW, which subsequently enables independent parameter estimation within each mixture and in turn, leads to convergence of the mixture weights. This framework also resolves the issue of unboundedness in estimated parameters present in the original FW approach. Complexity analysis shows that only a polynomial number of samples is required for each candidate in the target population.

Extensive numerical experiments are conducted in Chapter 4, including both simulation and case studies on the well-known Nielsen Consumer Panel Data, to demonstrate the effectiveness of recovering the true model parameters and/or learning realistic component-level parameters, as compared to the original FW framework.

Thesis Supervisor: David Simchi-Levi
Title: Professor, Institute of Data, Systems, and Society,
Department of Civil and Environmental Engineering,
Operations Research Center

This thesis is dedicated to my grandmother,
Zhengying Tian (1934-2017)

Acknowledgments

This thesis would not have been possible without the help of many people.

First, I would like to thank my thesis advisor, Professor David Simchi-Levi, for his continuous support. I am grateful for the freedom David granted when I tried to explore new research ideas. Hie broad and extensive experience and insights in the synergy of operations research and machine learning have many times guided me to the correct direction. The skills and experience I accumulated through the projects working with David have helped me become a better problem solver and researcher.

I would also like to thank my thesis committee members, Professor John Williams and Professor Saurabh Amin. I appreciate their time for serving on my committee and providing many insightful comments and suggestions, which play an important role in improving the quality of this thesis.

I feel very fortunate to have collaborated with wonderful people during my PhD journey. I enjoyed working with Zhenzhen Yan, who encouraged me, helped me consolidate many research questions to actionable items, and solved many difficulties together with me. I appreciate all the inspiring and valuable discussions with Yunzong Xu, who did not only offer great help in completing this thesis, but also widened my research vision in various aspects with his immense knowledge. I am also truly thankful to my friends, Yan Zhao and Xiao Fang, for having enlightening conversations with me regarding both life and career choices, without which I would not have been who I am today.

I would like to take this opportunity to express my gratitude to all of my friends who accompanied me through my ups and downs at MIT: Rui Sun, Hanzhang Qin, Peter Zhang, Jing Lu, Yanlin Cheng, Annie Dong, Li Wang, Jinglong Zhao, Jinzhi Bu, Louis Chen. I would also like to thank Ms. Janet Kerrigan for her constant support during my PhD life.

Finally, I owe my deepest gratitude to my parents, for their unconditional love and support. Thank you for accepting and embracing my imperfections and always having faith in me for whatever decisions I make during my entire study journey.

Part of this work uses the Nielsen Consumer Panel Data provided by the Kilts Center for Marketing at the University of Chicago School of Business. Any opinions, findings, and conclusions in this thesis are those of the authors and do not necessarily reflect the views of NielsenIQ or the Kilts Center.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Multinomial logit models (MNL) are widely used in a variety of settings to predict the probabilities of discrete outcomes. It generalizes the standard logistic regression to accommodate for the situations where there are more than 2 outcomes and computes the probability of choosing outcome j from a set of n possible values using the following formula:

$$\mathbb{P}_j = \frac{\exp(v_j)}{\sum_{i=1}^{n} \exp(v_i)},$$

where $v_j = \sigma(\mathbf{z}_j)$ can be thought as a score function that takes the attributes of option j, $\mathbf{z}_j \in \mathbb{R}^d$, as input. It is termed as *multinomial logistic regression* in statistics, where linear predictor function $\sigma(\mathbf{z}_j) = \boldsymbol{\beta} \cdot \mathbf{z}_j$ is a popular choice for σ with $\boldsymbol{\beta} \in \mathbb{R}^d$ representing the coefficients to be estimated from data. Such simplicity makes the analysis of statistical properties of the estimators more accessible. In many machine learning frameworks, it is well known as the `softmax` function for multi-class classification problems where σ can be highly complex and non-linear. For instance, in neural networks, σ is a chain of sequential linear transformations and activation filters. In choice modeling, it is a type of discrete choice model that is broadly used to analyze and understand people's choice behaviors, where v_j is viewed as the *utility* of choosing option j.

However, in many cases using a single MNL model to model the entire data does not yield good performances. This can happen due to the following reasons:

- heterogeneous population

  Think of a case where a population of decision makers have different valuations on the options' attributes. In the linear score function case, this means there are multiple $\boldsymbol{\beta}$ values for $\sigma = \boldsymbol{\beta} \cdot \mathbf{z}$. If we model the data with one universal function σ, we face the problem of model misspecification.

- violation of the "independence of irrelevant alternative" (IIA) assumption

  When an MNL is used to model choices, it relies on the assumption of IIA, which states that the relative odds of choosing one option over another do not depend on the presence of other alternatives or their attributes. This can be seen from taking the following computation:

  $$\frac{\mathbb{P}_j}{\mathbb{P}_i} = \frac{\exp(v_j)/\sum_k \exp(v_k)}{\exp(v_i)/\sum_k \exp(v_k)} = \frac{\exp(v_j)}{\exp(v_i)}.$$

  Nevertheless, this is not always desirable. In the famous blue-bus-red-bus problem [Chipman, 1960], a traveler originally has a 1:1 odds ratio choosing between traveling by a car and a blue bus. Suppose a new option of red bus is introduced, with all aspects the same as the blue bus except for the color. Intuitively, we should observe a 1:1 odds ratio between the blue bus and the red bus. In order for this to be true, the ratio between the three options becomes 1:1:1, which means there are now twice the probability to choose the bus compared to the car. However, introducing a bus of a new color should not really alter the traveler's preference for car versus bus in reality. Such violation of IIA arises because a red bus was a perfect substitute for a blue bus. Similar examples are also discussed in de Dios Ortuzar [1983], Brownstone and Train [1998].

- insufficient model capacity

  In high dimensional multi-class classification problems, the number of possible outcomes can be much larger than the attribute size, i.e. $n \gg d$. For instance, in language models, attributes are usually word embeddings with length between 2^7 to 2^{10}, while the number of classes is the size of the vocabulary which can easily reach tens of thousands. Yang et al. [2018] formulate this situation using matrix factorization and demonstrates that using a single `softmax` does not have enough capacity to theoretically recover the model parameters. This is referred to as *softmax bottleneck*.

To alleviate these issues, mixed multinomial logit (MMNL) models were introduced. They were used extensively to model the automobile markets in the U.S in the 1980s, as well as other industries such as telephone services and coffee purchases, in order to "explicitly incorporates variations in consumer tastes across the car-buying population" [Boyd and Mellman, 1980, Cardell and Dunbar, 1980, Train et al., 1987, Guadagni and Little, 2008]. McFadden and Train [2000] shows that MMNL models can lift the restrictive IIA assumption present in single MNL models. In a similar fashion for the high dimensional classification problem, new method named *Mixture of Softmaxes* (MoS) has been shown to have higher expressiveness and can better incorporate the contextual information [Yang et al., 2018, 2019].

Despite its importance and versatility, the study on the learning theory of MMNL is limited and learning MMNL models remains an open research topic. In this thesis, we will tackle this learning problem from two different perspectives. First, inspired by the recent work in Gaussian Mixture Models (GMM), we aim to explore the polynomial learnability of MMNL models from a theoretical point of view. Next, we present an algorithm that is designed to be more applicable and utilizes the rich source of data available in the modern digitalization era, yet still yielding ideal statistical properties of the estimators. Note that we will mainly adopt the discrete choice modeling setting to concretize concepts, definitions, and data assumptions in this work; however, the algorithms and properties we develop are not application dependent.

In this chapter, we will first introduce the basics of MNL and MMNL models, including random utility models (RUM), derivation of logit formula from RUM under appropriate assumptions, as well as the mathematical formulation of MMNL and its advantage over standard MNL models. In Section 1.2, we summarize the previous work on learning MMNL, together with the challenges and existing issues, many of which have inspired our work, from both theoretical and empirical perspectives.

Finally, we will outline the thesis structure in Section 1.3 aligning with these two streams of work.

1.1 MMNL Formulation

1.1.1 Multinomial Logit and Random Utility Models

MNL models are discrete choice models derived under an assumption of utility-maximizing behavior by the decision maker [Train, 2009]. This type of models are called random utility models (RUM) [Manski, 1977]. Consider a decision maker who needs to make a choice among n options. The decision maker will obtain certain level of utility from each option and the decision maker will choose the option that offers the largest utility. However, in reality we do not observe their utilities directly when trying to model the choice behavior of the decision makers. Instead, we observe some attributes of the options, denoted as $\mathbf{z}_j \in \mathbb{R}^d$. As a remedy, we assume utility is a function that models the valuation of decision makers with respect to these option attributes. Mathematically, this means the utility of choosing option j is $v_j = \sigma(\mathbf{z}_j)$.

On the other hand, note that there are certain aspects of utility which may depend on attributes that are not being observed. We let ϵ_j capture the factors that affect utility but are not included in v_j. In other words, the true utility $u_j = v_j + \epsilon_j$ is consisted of two parts: an observed portion and an unobserved portion. RUM claims that option j will be chosen by the decision maker if $u_j > u_i,\ \forall\ i \neq j$.

The characteristics of ϵ_j can vary depending on the model specifications. In probit models, ϵ_j are assumed to be normally distributed. In logit models, ϵ_j are assumed to be independently, identically distributed (i.i.d.) random variables drawn from a standard Gumbel distribution, i.e. Gumbel$(0,1)$, which is also called Generalized Extreme Value distribution Type-I. Gumbel distribution is usually used to represent the distribution of maximum value of a sequence of i.i.d random variables. The associated density is

$$f(\epsilon_j) = e^{-\epsilon_j} e^{-e^{-\epsilon_j}}$$

and the corresponding cumulative distribution is

$$F(\epsilon_j) = e^{-e^{-\epsilon_j}}$$

Thus we can write the probability that option j is chosen as

$$\begin{aligned}\mathbb{P}_j &= \mathbb{P}(v_j + \epsilon_j > v_i + \epsilon_i, \forall\ i \neq j)\\ &= \mathbb{P}(\epsilon_i < \epsilon_j + v_j - v_i, \forall\ i \neq j)\end{aligned}$$

Since all ϵ_i are independent, we can write

$$\mathbb{P}_j | \epsilon_j = \prod_{i \neq j} e^{-e^{-(\epsilon_j + v_j - v_i)}}$$

by plugging the cumulative distribution function. Note that ϵ_j is not give, so we need to integrate over all values of ϵ_j to obtain

$$\mathbb{P}_j = \int \left(\prod_{i \neq j} e^{-e^{-(\epsilon_j + v_j - v_i)}} \right) e^{-\epsilon_j} e^{-e^{-\epsilon_j}} d\epsilon_j$$

By variable substitution and some algebraic manipulation, we then obtain

$$\mathbb{P}_j = \frac{\exp(v_j)}{\sum_i \exp(v_i)} = \frac{\exp \sigma(\mathbf{z}_j)}{\sum_i \exp \sigma(\mathbf{z}_i)}$$

which gives us the familiar logit formula of MNL models.

1.1.2 Mixed Multinomial Logit

In the most generic form, MMNL are the integrals of standard logit probabilities over a density of parameters [Train, 2009]. Specifically, the choice probabilities of an MMNL model exhibit the form of

$$\mathbb{P}_j = \int \frac{\exp \sigma(\mathbf{z}_j; \boldsymbol{\beta})}{\sum_i \exp \sigma(\mathbf{z}_j; \boldsymbol{\beta})} f(\boldsymbol{\beta}) d\boldsymbol{\beta} \tag{1.1}$$

where $f(\boldsymbol{\beta})$ is the density function of parameters $\boldsymbol{\beta}$ in the utility function σ.

In this thesis, we consider one class of MMNL models, where $\boldsymbol{\beta}$ can take K discrete values with certain probabilities, so Eqn. (1.1) becomes

$$\mathbb{P}_j = \sum_{k=1}^{K} \alpha_k \frac{\exp \sigma(\mathbf{z}_j; \boldsymbol{\beta}_k)}{\sum_i \exp \sigma(\mathbf{z}_j; \boldsymbol{\beta}_k)} \tag{1.2}$$

where $f(\boldsymbol{\beta}_k) = \alpha_k$, $\forall\ k$ and $\sum_{k=1}^{K} \alpha_k = 1$.

In this case, MMNL are consisted of K individual MNL *components* or *mixtures* and is sometimes referred to as *latent class models* (LCM) [Greene and Hensher, 2003]. We can think of the situation where we want to model the choice behavior of a decision maker population which contains K different types. Each type possesses difference choice preferences when facing a set of options and can be modeled by one MNL model. Such model is frequently used in psychology and marketing [Chintagunta et al., 1991].

We call $\mathbb{P}_j$ the *aggregated choice probability*, which is computed as a weighted sum of all corresponding choice probability values for option j in each individual MNL component, with the weight equal to the mixture weights, i.e. α_k's.

MMNL models are highly flexible. As shown by McFadden and Train [2000], it can approximate any random utility based discrete choice models to arbitrary degree of accuracy under appropriate assumptions. In addition, it removes other limitations of standard MNL models, including lifting the IIA assumption, allowing correlation in unobserved utility factors overtime, and most importantly, reducing the chance of model misspecification when modeling real world applications where the data composition is often heterogeneous.

1.2 Literature Review

In this section, we will discuss the existing literature on learning the MMNL model and their limitations. One common approach is to presume a parametric family of distributions on the parameters and apply parametric estimation methods such as the maximum likelihood estimation (MLE) method (c.f. Train [2009]) or the least square regression model to compute the parameters. A well-known work in this regime is by Berry et al. [1995]. They assume the component-specific parameters are normal distributed and proposed a two-step estimation method to learn the parameters using an aggregated market share data. A potential issue for this stream of methods is model misspecification, which leads to inaccurate predictions. In particular, if the assumed parametric family is different from the true one, there could be systematical errors in the estimation.

Non-parametric estimation is also widely adopted when learning MMNL models, with the most commonly known and used method being the Expectation-Maximization (EM) algorithm (c.f. Dempster et al. [1977]). Train [2008] studied three different types of EM algorithms using historical choice data from each individual decision maker. The two main restrictions for the EM algorithm are 1) the algorithm can get stuck in local optimum, meaning there is no guarantee on the convergence of the estimators; and 2) the number of mixture types needs to be pre-specified, which is usually done via additional heuristics such as Akaike information criterion (AIC) or Bayesian information criterion (BIC). In the case where such heuristic fails to identify the ground-truth number of mixtures, model misspecification will also happen.

Jagabathula et al. [2020] recently developed an estimation method for mixture models also from a non-parametric perspective based on the Frank-Wolfe (FW) algorithm, which is an iterative method originally designed to solve constrained quadratic optimization. Jaggi [2013] established a sublinear convergence rate for the FW algorithm. Since achieving remarkable performance in various applications such as the collaborative filtering [Jaggi and Sulovskỳ, 2010] and submodular function optimization [Bach, 2013], various variants have also been proposed. Examples of some important variants include Harchaoui et al. [2015] who incorporated a regularization term in the loss function to improve predictive performance, and Lacoste-Julien and Jaggi [2015] who introduced away-steps FW, pairwise FW and fully-corrective FW to achieve global linear convergence rate under mild conditions. Jagabathula et al. [2020] incorporated these desired properties of the FW variants and in turn established a sublinear convergence rate of the proposed estimation method for MMNL models. However, this convergence only applies to the aggregated choice probabilities and their algorithm cannot recover the true individual MNL parameters.

There is also another stream of work related to MMNL models from a theoretical learning perspective. The key idea is to solve a system of equations in terms of all the parameters and prove that there is one and only one solution (identifiability). Due to the entanglement of the individual logit parameters and mixture weights, as well as the high non-linearity in the system of equations, only 2-MNL models (MMNL with two mixture types) have been studied so far. Chierichetti et al. [2018] considers the setting where the mixture weights are equal (i.e., each mixture represents 50% of the population) and Tang [2020] recently studied 2-MNL with unknown weights. Both have shown that 2-MNL models are polynomial learnable.

We compare the past attempts for learning MMNL models in Table 1.1.

Table 1.1: MMNL learning algorithm comparisons

Algorithm	Data Assumption	Model Assumption	Theoretical Properties
Parametric Estimation	○	×	○
EM Algorithm	×	⊗	×
Frank-Wolfe	○	○	⊗
2-MNL Models	○	×	○

- The *data assumption* column indicates whether the algorithm requires only population level data (○) versus personal level choice data (×). The population level data usually refers to the overall distribution of the option set from the population over multiple time periods, while personal level choice data records the repetitive choices from each decision maker in the population. Even though the latter poses additional requirement on the data, we do not consider it a strong assumption due to the abundant data availability in such format, which is called *panel data* and are commonly used statistics and econometrics for longitudinal studies.

- The *model assumption* column indicates whether the algorithm can be applied to any MMNL models (○) or are restricted to a smaller subset (×). It can also be thought as an indicator whether model misspecification is likely to occur, i.e., when there are stronger model assumptions, the chance of them being violated is also higher if the algorithm is not being applied appropriately. Note for the EM case, while the algorithm itself is very generic, it requires the knowledge of the number of mixtures in advance. Model misspecification can still happen if this hyperparameter is not set correctly, hence we give it a ⊗ mark.

- The *theoretical properties* column indicates whether the estimators possess any desired statistical properties, such as convergence. Note that the Frank-Wolfe algorithm receives a ⊗ mark because its convergence property only applies to the aggregated choice probability values, instead of the model parameters which are the target of interest in the learning problem.

1.3 Overview

In this thesis, we aim to develop new learning algorithms for MMNL that can address the issues in existing work as shown in Table 1.1, by relaxing the assumptions on model structures and proving desirable theoretical properties. In particular, we are interested in the provable convergence of the estimators. Formally, we want the algorithms to generate estimates $\hat{\alpha}_k$ and $\hat{\boldsymbol{\beta}}_k$, such that $\forall \epsilon > 0, 0 < \delta < 1$, $\mathbb{P}(|\hat{\alpha}_k - \alpha_k| < \epsilon) \geq 1 - \delta$ and $\mathbb{P}(\|\hat{\boldsymbol{\beta}}_k - \boldsymbol{\beta}_k\| < \epsilon) \geq 1 - \delta$, $\forall k$.

Chapter 2 studies the polynomial learnability of identifiable MMNL models with a general K number of mixtures. This work relaxes the model assumption of 2-MNL models as shown in Table 1.1, while not trying to impose additional *data assumption* or sacrificing the important *theoretical properties.* We analyze the existence of ϵ-close estimates using tools from abstract algebra and will show that there exists an algorithm that can learn a general K-MNL models with probability at least $1 - \delta$, if identifiable, using polynomial number of data sample and polynomial number of operations (in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$) under some reasonable assumptions.

Chapter 3 proposes a new algorithm called Stochastic Subregion Frank-Wolfe (SSRFW), as inspired by the recent work by Jagabathula et al. [2020] that uses Frank-Wolfe algorithm to iteratively learn new MNL components by solving a linear sub-problem then redistributing the mixture weights via another optimization step. The advantage of the FW framework over the EM algorithm is that no prior knowledge on the number of mixtures, K, is needed, hence eliminating the possibility of model misspecification. SSRFW is designed with the objective to recover the ground truth parameter values, which the original FW approach failed to achieve. This is done by making better utilization of personal level choice data. In addition, we are also able to provide a sample complexity analysis to show that a polynomial number of data points are required under the SSRFW framework.

Table 1.2 summarizes the two algorithms we develop in this thesis. The algorithm used to demonstrate the polynomial learnability of K-MNL models is marked with $\otimes$ as we impose some mild conditions on the model hyperparameters, but they are much more generic than the 2-MNL model setting. On the other hand, while the algorithm is polynomial in sample and time, it is designed to explore the theoretical structure of MMNL models instead for practical usage. Therefore we also developed SSRFW as a good complement as it is more applicable in many real world situations.

Table 1.2: MMNL learning algorithms developed in this thesis

	Algorithm	Data Assumption	Model Assumption	Theoretical Properties
Ch2	K-MNL (grid-search)	○	$\otimes$	○
Ch3	SSRFW	×	○	○

Chapter 4 conducts numerical experiments to evaluate the performance of the SSRFW algorithm in various settings. In the first part, we carry out comprehensive simulation studies to demonstrate the effectiveness of SSRFW in recovering the ground truth parameter values using several evaluation metrics. In the second part, we run a real case study that applies the SSRFW algorithm to the well-known Nielsen Consumer Panel data, which is one of the most popular datasets used for longitudinal studies in marketing science and have been collecting consumer grocery purchase data for around 20 years.

Finally, Chapter 5 provides a summary to the work done in this thesis. It also provides examples of use cases where MMNL model can be applied to solve real world problems.

Chapter 2

Polynomial Learnability of MMNL

In this chapter, we will explore the polynomial learnability of MMNL models. While various heuristics have been developed for learning MMNL models, theoretical approaches that utilize the unique structures of MMNL models are scarce. Chierichetti et al. [2018] and Tang [2020] recently studied this problem but have limited their scope to 2-MNL models. In comparison, there exits another stream of work that also study the (mixture of) choice models under random utility model (RUM) assumptions. For instance, see Ragain and Ugander [2016], Blanchet et al. [2016], Seshadri et al. [2020]. Such models are called *Plackett-Luce* (PL) models and instead of choice data, they use *rank data*. Nevertheless, PL models are closely related to MNL models. In fact, the logit formula was first derived by Luce [1959] and its important property "independence from irrelevant alternatives" (IIA) are also known as *Luce's Choice Axiom*. We will discuss more about PL and mixed PL models in Section 2.1.

Aside from the lacking of study of MMNL models from the theoretical perspective in literature, this work is also inspired by the recent progress in understanding Gaussian mixture models (GMM) for its polynomial learnability. Even though GMM have been around for more than 100 years, not until recently did we observe the settling of polynomial learnability for general GMM without assuming any special structures. Kalai et al. [2010] first derived results for mixtures of two Gaussians, followed by Moitra and Valiant [2010] who then generalized the results to K-GMMs. Belkin and Sinha [2015] further extends the theory to a broader class of mixture models whose moments are polynomial functions of the model parameters.

To show that K-MNL models are polynomial learnable under certain assumptions, we will break down the process into several steps. First, we will re-introduce the problem in Section 2.1 to follow the convention of existing works in this area and discuss the assumptions we impose. Second, we will present some concepts and theories in algebraic geometry and mathematical logic in Section 2.2.1, followed by the core building block of our approach, the method of moments (MOM), in Section 2.2.2. Section 2.3 will utilize these tools to derive the main theorem, based on a proposed grid search algorithm.

2.1 Problem Formulation

We introduce a slightly different notation in this chapter to be consistent with the past work done in the theoretical learning setting. Specifically, consider $\{p_\theta\}_{\theta\in\Theta}$ the family of K-MNL models for a set of m alternatives, denoted by $[m] = \{1, \dots, m\}$. We also refer to this setting as K-MNL models under an *m-item universe*. We can also think of p_θ as an m-size vector, where the j-th element in p_θ is the probability of choosing alternative j.

p_θ is parameterized by $\theta \in \Theta \subset \mathbb{R}^{K(m+1)}$. Specifically, we can write

$$\begin{aligned}\Theta = \{\theta \in \mathbb{R}^{K(m+1)} | (\theta_{m(k-1)+1}, \dots, \theta_{mk}) \in \Delta_{m-1}, k = 1, \dots, K,\\ (\theta_{Km+1}, \dots, \theta_{K(m+1)}) \in \Delta_{K-1}\}\end{aligned} \tag{2.1}$$

where $\Delta_{n-1} = \{(\theta_1, \dots, \theta_n) \in \mathbb{R}^n_+ \big| \sum_{j=1}^n \theta_j = 1\}$ is the $(n-1)$-simplex. To simplify notation, we use $\theta^{(k)} = (\theta_{k1}, \dots, \theta_{km})$ to denote $\left(\theta_{m(k-1)+1}, \dots, \theta_{mk}\right)$ for $k = 1, \dots, K$ and $(w_1, \dots, w_K)$ for $\left(\theta_{Km+1}, \dots, \theta_{K(m+1)}\right)$. The k-th MNL mixture is characterized by $(\theta_{k1}, \dots, \theta_{km})$, with θ_{kj} representing the probability of choosing alternative j given the entire alternative universe. The remaining $(w_1, \dots, w_K)$ parameters specifies the mixture weight. By definition, Θ is a compact set.

Note that this parameter set has a one-to-one mapping with the notations introduced in Chapter 1. Specifically, we have $\theta_{kj} = \exp \sigma(\mathbf{z}_j; \boldsymbol{\beta}_k)$ and $w_k = \alpha_k$. Once we learn all θ_{kj}'s, we can compute $\boldsymbol{\beta}_k$ values given that σ is an invertible function, such as the linear utility function.

An s-slate is a subset $\mathcal{S} \subseteq [m]$ of size s and the probability of choosing alternative j in this slate is defined as $q_k(j|\mathcal{S}) = \dfrac{\theta_{kj}}{\sum_{i \in S} \theta_{ki}}$ for mixture k. The aggregated population-level choice probability for alternative j given slate $\mathcal{S}$ is $p_\theta^{\mathcal{S}}(j) = \sum_{k=1}^{K} w_k q_k(j|\mathcal{S})$ for $j \in \mathcal{S}$. We will use $q_k^{\mathcal{S}} \in \mathbb{R}^s$ and $p_\theta^{\mathcal{S}} \in \mathbb{R}^s$ to denote the individual logit vector (for mixture k) and the aggregated mixed logit vector respectively. In addition, when we consider the MMNL model for given a slate $\mathcal{S} \subseteq [m]$, we let $\Theta^{\mathcal{S}}$, as a parameter subspace of Θ, denote the components of θ for alternatives in $\mathcal{S}$.

Let $D_\theta^{\mathcal{S}}$ be an oracle that returns the true value of $p_\theta^{\mathcal{S}}(j), \forall\ j \in \mathcal{S}$. In general, we do not have access to this oracle and will use choice (order) data to approximate the values of $p_\theta^{\mathcal{S}}$. A *choice (order) data* is the observed probability distribution for alternatives in a slate $\mathcal{S}$, $\boldsymbol{y}^{\mathcal{S}} \in \mathbb{R}^s$. With n samples, we can compute $\hat{p}_\theta^{\mathcal{S}} = \frac{1}{n} \sum_{t=1}^{n} \boldsymbol{y}_t^{\mathcal{S}}$.

Polynomial Learnability. Let K be a fixed constant. Let $\{p_\theta\}_{\theta \in \Theta}$ be the family of identifiable K-MNL models parameterized by $\theta \in \Theta \subseteq \mathbb{R}^{K(m+1)}$ with Θ defined in 2.1. Assume $\|\cdot\|$ is the l_2 norm. Given precision ϵ and confidence δ, if there exists an algorithm that can provide an estimate to θ, $\hat{\theta}$, such that $\|\hat{\theta} - \theta\| < \epsilon$ with probability at least $1 - \delta$ using n data samples, where n is a polynomial function of $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$, i.e. $n = \text{poly}(\frac{1}{\delta}, \frac{1}{\epsilon})$, and the number of operations in the algorithm is also polynomial, we say this problem is *polynomial learnable.*

While the sample complexity is polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$, we expected it to be exponential in K in general. As for the Gaussian Mixture Model (GMM) case, where K being in exponential is unavoidable [Moitra and Valiant, 2010], we conjecture the same for K-MNL models. Since the learnability of K-MNL models is not well-understood even when K is small constant, we focus on studying the polynomial learnability of K-MNL models assuming that K is an arbitrarily given constant.

2.1.1 Identifiability

Identifiability of a model determines whether it is possible to learn the true values of the model's parameters after obtaining an infinite number of observations from it. We first define the identifiability of MMNL models.

Definition 2.1 (Identifiability). *Let $\{p_\theta\}_{\theta \in \Theta}$ be a family of K-MNL models over an m-item universe, $\theta \in \Theta \subseteq \mathbb{R}^{K(m+1)}$. We say p_θ, for all $\theta \in \Theta$, is identifiable if $p_{\theta_1}^{\mathcal{S}} = p_{\theta_2}^{\mathcal{S}}$ for every $\mathcal{S} \subseteq [m]$ implies $\theta_1 = \theta_2$, $\forall\ \theta_1, \theta_2 \subseteq \Theta$.*

Denote the power set of a set S as $\mathcal{P}(S)$, which contains all of the subsets of S. We have $|\mathcal{P}(S)| = 2^n$ if $|S| = n$.

Adapting from Tang [2020, Assumption 1.2], we make the following assumption:

Assumption 2.1 (ϕ-identifiability). *The family of K-MNL models $\{p_\theta\}_{\theta \in \Theta}$ over a $\phi(K)$-item universe is identifiable.*

In other words, we consider $\phi(K)$ as the smallest size of an alternative set in order for an MMNL model with K mixtures to be identifiable. For notation simplicity and when there is not ambiguity, we use m_0 to denote $\phi(K)$, assuming K as given.

The intuition why such m_0 exists is the follows:

The parameters need to satisfy the following system of equations:

$$\sum_{k=1}^{K} w_k \frac{\theta_{kj}}{\sum_{i \in \mathcal{S}} \theta_{ki}} = D_\theta^{\mathcal{S}}(j), \forall\ j \in \mathcal{S}, \forall\ \mathcal{S} \in \mathcal{P}([m_0]) \tag{2.2}$$

$$\sum_{k=1}^{K} w_k = 1 \tag{2.3}$$

$$\sum_{i=1}^{m_0} \theta_{ki} = 1, \forall\ k = 1, \dots, K \tag{2.4}$$

where D_θ^S is the oracle defined as above. In Equation (2.2), there are $s-1$ independent equations for each $|\mathcal{S}| = s$. Therefore, this system has a total of

$$\sum_{s=1}^{m_0} (s-1) \binom{m_0}{K} + 1 + K = 1 + K + 2^{m_0 - 1}(m_0 - 2)$$

equations. If $\{p_\theta\}_{\theta\in\Theta}$ is identifiable over the m_0-item universe, this system has a unique solution θ. When we increase the universe size such that $m \geq m_0$, we introduce more equations to the system and making the solution set for the system even restrictive. This means it is extremely unlikely that there exists both θ and θ' such that $\theta' \neq \theta$ while both θ and θ' satisfy the larger number of $1 + K + 2^{m-1}(m-2)$ equations.

In this chapter, we will study K-MNL models with m-item universe where $m \geq m_0$ using choice data.

Identification condition

A natural question to ask is what is $\phi(K)$ for a K-MNL model. In fact, the definition of $\phi(K)$ precisely characterizes the *identification condition* of K-MNL models. At the time of this work, only the identification conditions for 2-MNL models have been established, namely $m \geq 3$ for uniform 2-MNL ($w_1 = w_2 = \frac{1}{2}$) by Chierichetti et al. [2018] and $m \geq 4$ for arbitrary weight 2-MNL (more discussion below).

There is another stream of work that studies a different type of choice models that is closely related to MNL, namely the Plackett-Luce (PL) models. We will briefly discuss the (non)-identification conditions for K-PL models since there are better established results associated with this class of models.

For PL models, let $\mathcal{X} = \{x_1, \ldots, x_m\}$ be a set of m alternative and denote $\mathcal{L}(\mathcal{X})$ as the set of linear orders (full rankings) over $\mathcal{X}$. A *ranking* $R \in \mathcal{L}(\mathcal{X})$ is $x_{i_1} \succ x_{i_2} \succ \ldots \succ x_{i_m}$ where x_{i_1} and x_{i_m} are the most and least preferred alternative respectively. Ranking data are obtained by repeatedly selecting items after removing the previously selected items, according to the MNL choice model. Developed by Plackett [1968, 1975], the *ranking distribution* for $R = [x_{i_1} \succ x_{i_2} \succ \ldots \succ x_{i_m}]$ is characterized by

$$\begin{aligned}\mathbb{P}(R|\theta) &= \prod_{p=1}^{m-1} \mathbb{P}(x_{i_p}|\mathcal{X} \setminus \cup_{q=1}^{p-1} x_{i_q}; \theta) \\ &= \prod_{p=1}^{m} \frac{\theta_{i_p}}{\sum_{q=p}^{m} \theta_{i_q}} \end{aligned} \tag{2.5}$$

Manski [1977] shows that any RUM models, e.g. MNL, can be composed into a utility-based ranking model, e.g. PL, via such repeated selection.

Zhao et al. [2016] showed that the identifiable condition for K-PL models using rank data is as follows:

- For any $K \geq 2$, the mixture of K PL models for no more than $2K-1$ alternatives is non-identifiable and this bound is tight for $K = 2$.

- Mixture of K PL models over m alternatives is *generically* identifiable if $K \leq \lfloor \frac{m-2}{2} \rfloor !$.

The concept of *generic identifiability* is introduced by Allman et al. [2009]. It describes the property of a model, which is not strictly identifiable in the parameter space Θ, but the non-identifiable parameter choices form a set of Lebesgue measure zero. The way to understand the identifiability conditions for K-PL models above is: when $\frac{m+1}{2} \leq K \leq \lfloor \frac{m-2}{2} \rfloor !$, though K-PL model is not identifiable in the strict sense (i.e., $\forall\, p_\theta$, we can recover θ), it is in general safe to ignore this problem in practice due to generic identifiability. They also conjecture that the identification condition for K-PL models is $m \geq 2K$ for $K \geq 3$, but this still remains an open research question.

Later, Zhao and Xia [2019] showed 2-PL model given appropriate choice data are identifiable, which can be used directly to induce the identification condition for 2-MNL models. We summarize this conclusion below.

Corollary 2.1 (Theorem 2 [Zhao and Xia, 2019]). *Let $\Phi^{choice\text{-}l}$ be some structure (oracle) that returns the choice order of l-size slate $\mathcal{S}$. Let $\Phi^* = \cup_{l=2}^{4} \Phi^{choice-l}$. For any $\Phi \supset \Phi^*$, 2-MNL model over $m \geq 4$ alternatives is identifiable.*

The *choice order* in Zhao and Xia [2019] is defined as the probability of choosing any item j, for any given slate $\mathcal{S}$, i.e. $\mathbb{P}(j|\mathcal{S})$, for $j \in \mathcal{S}$. This is exactly what we referred to as the choice data as discussed earlier. On the other hand, even though the original theorem in Zhao and Xia [2019] is proven for 2-PL models, the learned parameters can be used directly to construct the 2-MNL model, as we can see from Equation (2.5). Therefore we can conclude that $m \geq 4$ is the identification condition for 2-MNL models, i.e., $\phi(2) = 4$.

The identification conditions for K-PL models have been shown to be closely related to K-MNL models in Zhao and Xia [2019], and identical in the case of $K = 2$. While rigorous derivations are still needed to establish the (generic) identification conditions for K-MNL models, we believe that the identification conditions hold for a class of K-MNL models are realistic assumptions — as we have seen, $\phi(K)$ is 4 for $K = 2$, and what we assume in Assumption 2.1 is essentially $\phi(K) < \infty$ for any given K (in fact, based on existing literature, we believe that a natural conjecture is "$\phi(K)$ is a polynomial function of K", which we leave as an open problem).

On top of Assumption 2.1, we will focus on exploring the polynomial learnability for K-MNL models with choice data over an m-item universe where $m \geq m_0$ with $m_0 = \phi(K)$ the minimum size of the alternative set for K-MNL being identifiable. While studying the identification conditions for K-MNL models is one of the central topics in studying the learnability of K-MNL models, we believe exploring the lower bounds for m_0 is also a good research direction. Recall that we conjecture that the sample complexity is exponential in K and on top of that, $m_0 = \phi(K)$ is a polynomial function of K. It follows that sample complexity is also exponential in m_0. Based on this conjecture, if we can find smaller m_0 values, it can guide the design of efficient learning algorithms and reduce the sample complexity.

2.2 Mathematical and Statistical Tools

2.2.1 Semialgebraic Set and Tarski-Seidenberg Theorem

In this section, we introduce some tools used in algebraic geometry, including the definition of semialgebraic set and the Tarski-Seidenberg Theorem. We will also discuss some basic mathematical logic concepts that will also be used in the following sections.

Definition 2.2. *A set is said to be a semialgebraic subset of* $\mathbb{R}^d$ *if it can be represented as a finite union of sets defined by a system of polynomial equalities and inequalities.*

By the definition of Θ or $\Theta^{\mathcal{S}}$ if given a slate $\mathcal{S}$, which are composed by a set of algebraic equations, it is a semialgebraic set.

Theorem 2.1 (Tarski-Seidenberg)**.** *A linear projection of a semialgebraic set is semialgebraic.*

The important implication of Tarski-Seidenberg theorem is quantifier elimination, which states that every formula constructed from polynomial equations and inequalities by quantifiers $(\forall, \exists)$ is equivalent to a similar formula without quantifiers. This is known as the *elimination of quantifiers* and will be a key component when we prove our main result. For a simple example, consider the statement that a quadratic polynomial has a real root, i.e. $\exists x \in \mathbb{R}, (a \neq 0 \wedge ax^2 + bx + c = 0)$. This can equivalently be written as $a \neq 0 \wedge b^2 - 4ac \geq 0$, where it no longer involves using the quantifier $\exists x \in \mathbb{R}$.

Proposition 2.1. *Let* $P(x)$ *and* $Q(x)$ *be some polynomial function of* x. $\{x|(P(x) > 0) \Rightarrow (Q(x) > 0)\} = \{x|Q(x) > 0\} \cup \{x|(Q(x) \leq 0)\&(P(x) \leq 0)\}$.

Proof. Denote p and q as the logical statements for $P(x) > 0$ and $Q(x) > 0$ respectively. In the formal form of mathematical logic, the conditional statement $p \rightarrow q$ has the following truth table:

The logical implication $p \Rightarrow q$ holds if $p \rightarrow q$ is true. In other words, $p \Rightarrow q$ is only false when the hypothesis (p) is true and the conclusion (q) is false. Therefore, we

Table 2.1: Truth Table for $p \to q$

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

can establish the equivalence between $p \Rightarrow q$ and $(p \wedge q) \vee (\neg p \wedge q) \vee (\neg q \wedge \neg p)$ which is the same as $q \vee (\neg q \wedge \neg p)$. Plug in back the expression of p and q, we can see that set $\{x|(P(x) > 0) \Rightarrow (Q(x) > 0)\}$ can be written as set $\{x|Q(x) > 0\} \cup \{x|(Q(x) \leq 0)\ \&\ (P(x) \leq 0)\}$. $\square$

2.2.2 Method of Moments

Given a slate $\mathcal{S} \subseteq [m]$ of size s, we can think of the MMNL model p_θ^S as a probability distribution on the discrete set $\{i_1, \ldots, i_s\}$, similar as a standard categorical distribution. This allows us to utilize relevant probability tools to analyze this learning problem, such as the moments and methodologies depending on the concept of moments. Without loss of generality, we index the alternatives with $\{1, \ldots, s\}$.

First developed by the famous statistician Karl Pearson in the early 20-th century, method of moments (MOM) is widely used in statistical estimation through solving a system of equations established by equating sample moments with unobserved ground-truth moments to get the parameters to be estimated [Pearson, 1894]. Compared to the maximum likelihood estimation (MLE) method which can be intractable in certain scenarios without computers, method of moments in general can be computed more quickly and easily. Since intractability happens inevitably for learning mixture models using MLE, MOM is broadly used in developing learning algorithms for mixture models, including mixed exponential distribution [Rider, 1961], mixed hidden Markov models [Anandkumar et al., 2012], mixed PL models [Zhao et al., 2016], and Gaussian Mixture Models (more generically the class of polynomial distributions) [Belkin and

Sinha, 2015].

To use the method of moments in our setting, we start by showing that if we can uniquely determines the first s moments of the MMNL distribution on slate $\mathcal{S}$, then we can uniquely recover the aggregated logit vector.

Proposition 2.2. *Let $p_\theta^{\mathcal{S}}$ be a categorical distribution over s different values in the set $\mathcal{S}$. $p_{\theta_1}^{\mathcal{S}}(x) = p_{\theta_2}^{\mathcal{S}}(x)$, $\forall x \in \mathcal{S}$, if and only if $M_i(\theta_1|\mathcal{S}) = M_i(\theta_2|\mathcal{S})$, $\forall i = 1, \dots, s$, where $M_i(\theta|\mathcal{S}) = \mathbb{E}[x^i|\mathcal{S}]$ is the i-th raw moment of $p_\theta^{\mathcal{S}}$.*

Proof. It is easy to show that if $p_{\theta_1}^{\mathcal{S}} = p_{\theta_2}^{\mathcal{S}}$, we have $M_i(\theta_1|\mathcal{S}) = \sum_{x\in\mathcal{S}} p_{\theta_1}^{\mathcal{S}}(x) \cdot x^i = \sum_{x\in\mathcal{S}} p_{\theta_2}^{\mathcal{S}}(x) \cdot x^i = M_i(\theta_2|\mathcal{S})$, $\forall i$.

For the reverse direction, define $X^i = [1^i, 2^i, \dots, s^i]^\top$ and

$$\begin{aligned} P_i(\theta_1, \theta_2|\mathcal{S}) &\overset{\text{def}}{=} M_i(\theta_1|\mathcal{S}) - M_i(\theta_2|\mathcal{S}) \\ &= \sum_{x=1}^{s} p_{\theta_1}^{\mathcal{S}}(x) \cdot x^i - \sum_{x=1}^{s} p_{\theta_2}^{\mathcal{S}}(x) \cdot x^i \\ &= (p_{\theta_1}^{\mathcal{S}} - p_{\theta_2}^{\mathcal{S}})^\top X^i \end{aligned}$$

If $M_i(\theta_1|\mathcal{S}) = M_i(\theta_2|\mathcal{S})$ for $\forall i = 1, \dots, s$, we can represent $P_i(\theta_1, \theta_2|\mathcal{S}) = 0, \forall i$ as a system of equations as:

$$\underbrace{\begin{bmatrix} X^1 & X^2 & \cdots & X^s \end{bmatrix}^\top}_{A} \underbrace{\begin{bmatrix} p_{\theta_1}^{\mathcal{S}} - p_{\theta_2}^{\mathcal{S}} \end{bmatrix}}_{\xi} = 0$$

where $A \in \mathbb{R}^{s\times s}$ is a Vandermonde matrix and $\xi \in \mathbb{R}^s$. Since the columns of A are independent, A is invertible and the system has the unique solution $\xi = 0$. Therefore, we have $p_{\theta_1}^{\mathcal{S}}(x) = p_{\theta_2}^{\mathcal{S}}(x)$, $\forall x \in \mathcal{S}$. □

Theorem 2.2. *Let $p_{\theta\theta\in\Theta}$ be the family of mixed multinomial logit models over an m-item universe, $\theta \in \Theta \subseteq \mathbb{R}^{K(m+1)}$. Assume p_θ is m_0-identifiable and $m \geq m_0$. Let $\mathcal{S}^0$ be an arbitrary subset of $[m]$ and $|\mathcal{S}^0| = m_0$. There exists $t > 0$ such that $\forall \epsilon > 0$ and $\theta_1, \theta_2 \in \Theta$, if $|M_i(\theta_1|\mathcal{S}) - M_i(\theta_2|\mathcal{S})| \leq \epsilon$, $\forall i \leq |\mathcal{S}|$ for all slates $\mathcal{S} \subseteq \mathcal{S}^0$, we have $\|\theta_1 - \theta_2\| \leq O(\epsilon^t)$.*

Proof. Let $\mathcal{P}(\mathcal{S}^0)$ be the power set of $\mathcal{S}^0$. We start by observing that we can replace the condition $|M_i(\theta_1|\mathcal{S}) - M_i(\theta_2|\mathcal{S})| \leq \epsilon$ by

$$\begin{aligned} Q(\theta_1, \theta_2) &\overset{\text{def}}{=} \sum_{\mathcal{S}\in\mathcal{P}([\mathcal{S}^0])} \sum_{i=1}^{|\mathcal{S}|} |M_i(\theta_1|\mathcal{S}) - M_i(\theta_2|\mathcal{S})|^2 \\ &\leq N\epsilon^2 = \epsilon' \end{aligned} \tag{2.6}$$

where we assume N is the total number of summands in Equation (2.6).

Recall that for $\ell \in \{1, 2\}$

$$M_i(\theta_\ell|\mathcal{S}) = \sum_{x\in\mathcal{S}} x^i \left(\sum_{k=1}^{K} w_k^{(\ell)} \frac{\theta_{kx}^{(\ell)}}{\sum_{j\in\mathcal{S}} \theta_{kj}^{(\ell)}} \right)$$

where we add a superscript "(ℓ)" to the individual parameters to indicate which model they are representing. Next, we define a set of functions that serve as common denominator multipliers in the moment expression. Define

$$\mathscr{F}_{\theta_\ell} = \prod_{\mathcal{S}\in\mathcal{P}(\mathcal{S})} \prod_{k'=1}^{K} \sum_{i\in\mathcal{S}} \theta_{k'i}^{(\ell)}$$

and

$$\mathscr{F}_{\theta_\ell}(\neg k, \mathcal{S}) = \prod_{\mathcal{S}'\in\mathcal{P}(\mathcal{S}^0)} \prod_{\substack{k'=1 \\ k'\neq k}}^{K} \sum_{i\in\mathcal{S}'} \theta_{k'i}^{(\ell)} = \frac{\mathscr{F}_{\theta_\ell}}{\sum_{i\in\mathcal{S}} \theta_{ki}^{(\ell)}}$$

We have $\mathscr{F}_{\theta_\ell}$ and $\mathscr{F}_{\theta_\ell}(\neg k, \mathcal{S}), \forall k$ are polynomials in θ. In addition, $\forall \mathcal{S}, \forall k \in \mathcal{S}$, $0 < \theta_{kj}^{(\ell)} \leq 1$, hence $0 < \mathscr{F}_{\theta_\ell} \leq 1$.

Let

$$\begin{aligned}
\widetilde{Q}(\theta_1,\theta_2) &= \mathscr{F}_{\theta_1}^2 \mathscr{F}_{\theta_2}^2 Q(\theta_1,\theta_2) \\
&= \sum_{\mathcal{S}\in\mathcal{P}([\mathcal{S}^0])} \sum_{i=1}^{|\mathcal{S}|} \left[\sum_{x\in\mathcal{S}} \Big(\mathscr{F}_{\theta_1}\mathscr{F}_{\theta_2} p_{\theta_1}(x) - \mathscr{F}_{\theta_1}\mathscr{F}_{\theta_2} p_{\theta_2}(x)\Big)\cdot x^i\right]^2 \\
&= \sum_{\mathcal{S}\in\mathcal{P}([\mathcal{S}^0])} \sum_{i=1}^{|\mathcal{S}|} \left[\sum_{x\in\mathcal{S}} \mathscr{F}_{\theta_1}\mathscr{F}_{\theta_2} \left(\sum_{k=1}^{K} w_k^{(1)} \frac{\theta_{kx}^{(1)}}{\sum_{j\in\mathcal{S}}\theta_{kj}^{(1)}} - \sum_{k=1}^{K} w_k^{(2)} \frac{\theta_{kx}^{(2)}}{\sum_{j\in\mathcal{S}}\theta_{kj}^{(2)}}\right)\cdot x^i\right]^2 \\
&= \sum_{\mathcal{S}\in\mathcal{P}([\mathcal{S}^0])} \sum_{i=1}^{|\mathcal{S}|} \left[\sum_{x\in\mathcal{S}} \sum_{k=1}^{K} \left(\mathscr{F}_{\theta_1}(\neg k,\mathcal{S})\mathscr{F}_{\theta_2} w_k^{(1)}\theta_{kx}^{(1)} - \mathscr{F}_{\theta_1}\mathscr{F}_{\theta_2}(\neg k,\mathcal{S}) w_k^{(2)}\theta_{kx}^{(2)}\right)\cdot x^i\right]^2
\end{aligned}$$

The intuition here is that by multiplying the multiplier functions to Q, $\widetilde{Q}(\theta_1,\theta_2)$ now becomes a polynomial function in (θ_1,θ_2). In addition, let ϵ'' be some $\epsilon'' \le \epsilon'$, such that when $Q(\theta_1,\theta_2)\le\epsilon'$, we have $\widetilde{Q}(\theta_1,\theta_2)\le\epsilon''$.

The following part of the proof is similar to Belkin and Sinha [2015], which we adapted to our particular setting. Since Θ is compact, $\{(\theta_1,\theta_2)|\theta_1,\theta_2\in\Theta,\widetilde{Q}(\theta_1,\theta_2)\le\epsilon''\}$ is also compact. Hence, there exists some constant C such that for $\epsilon''\le\epsilon'<1$,

$$\sup_{\substack{\theta_1,\theta_2\in\Theta \\ \widetilde{Q}(\theta_1,\theta_2)\le\epsilon''}} \|\theta_1-\theta_2\| < C \tag{2.7}$$

since $\|\theta_1-\theta_2\|$ ($\ell 2$ norm unless otherwise specified) is continuous in (θ_1,θ_2)

Consider the set

$$D_{\epsilon''} = \left\{\delta>0 \,\middle|\, \forall_{(\theta_1,\theta_2)\in\Theta} \quad \Big(\widetilde{Q}(\theta_1,\theta_2)\le\epsilon''\Big) \Rightarrow \big(\|\theta_1-\theta_2\|<\delta\big)\right\}$$

Equation 2.7 indicates that $D_{\epsilon''}$ is non-empty. According to Proposition 2.1, $D_{\epsilon''}$ can be written as

$$\{\delta>0|\forall_{\theta_1,\theta_2\in\Theta} \quad \|\theta_1-\theta_2\|\ge\delta\} \cup \{\delta>0|\forall_{\theta_1,\theta_2\in\Theta} \quad \|\theta_1-\theta_2\|\ge\delta,\widetilde{Q}(\theta_1,\theta_2)>\epsilon''\}$$

By the Tarski-Seidenberg theorem, we can see that $D_{\epsilon''}$ is a semialgebraic set of $\mathbb{R}$. Let $\delta(\epsilon'') = \inf D_{\epsilon''}$. We can show that $\delta(\epsilon'')$ is also a semialgebraic set by first writing

$$\delta(\epsilon'') = \inf S_{\epsilon''} = \{z | \forall x \in S_{\epsilon''}\ (z \leq x)\} \cap \{z | \forall \epsilon > 0\ \exists x \in S_{\epsilon''}\ (z + \epsilon > x)\}$$

then applying again the Tarski-Seidenberg theorem.

According to Proposition 2.2, $Q(\theta_1, \theta_2) = 0$ implies $p_{\theta_1} = p_{\theta_2}$ for any choice of S^0 since p_θ is m_0 identifiable. By definition, we also have $Q(\theta_1, \theta_2) = 0$ if $\widetilde{Q}(\theta_1, \theta_2) = 0$. In turn, we obtain $\lim_{\epsilon'' \to 0} \| p_{\theta_1} - p_{\theta_2} \| = 0$. If p_θ family is identifiable, then $\lim_{\epsilon'' \to 0} \| \theta_1 - \theta_2 \| = 0$ as $\widetilde{Q}(\theta_1, \theta_2)$ is continuous in (θ_1, θ_2). This gives $\lim_{\epsilon'' \to 0} \delta(\epsilon'') = 0$.

Since $\delta(\epsilon'')$ is a single point set and strict inequalities alone cannot define the corresponding semialgebraic set, $\delta(\epsilon'')$ satisfies an algebraic equation whose coefficients are polynomial in ϵ''. We can write this polynomial as $q_{\epsilon''}(x) = q_M(\epsilon'')x^M + \ldots + q_0(\epsilon'')$ such that $q_{\epsilon''}(\delta(\epsilon'')) = 0$ and $\delta(\epsilon'')$ is a positive root $q_{\epsilon'}(x) = 0$.

From Lemma 2.1 (see below), we obtain $\delta(\epsilon'') < C'(\epsilon'')^{\frac{1}{M}}$ for some constant $C' > 0$. The definition of $\delta(\epsilon'')$ indicates that

$$\forall_{\theta_1, \theta_2 \in \Theta} \left(\widetilde{Q}(\theta_1, \theta_2) \leq \epsilon'' \right) \Rightarrow \left(\| \theta_1 - \theta_2 \| < C(\epsilon'')^{\frac{1}{M}} \right)$$

Note that $\forall_{\theta_1, \theta_2 \in \Theta} \left(\widetilde{Q}(\theta_1, \theta_2) \leq \epsilon'' \right)$ means $\forall_{\theta_1, \theta_2 \in \Theta}\ (Q(\theta_1, \theta_2) \leq \epsilon')$, hence equivalent to $\forall_{i \leq |\mathcal{S}|, \mathcal{S} \subseteq \mathcal{P}(\mathcal{S}^0)}\ (|M_i(\theta_1|\mathcal{S}) - M_i(\theta_2|\mathcal{S})| \leq \epsilon)$ for any $\mathcal{S}^0$ as an m_0-size subset of $[m]$. Thus we have

$$\forall_{\mathcal{S}^0 \subseteq [m]}, \forall_{i \leq |\mathcal{S}|, \mathcal{S} \in \mathcal{P}(\mathcal{S}^0)}\ (|M_i(\theta_1|\mathcal{S}) - M_i(\theta_2|\mathcal{S})| \leq \epsilon) \Rightarrow \left(\| \theta_1 - \theta_2 \| < C(\epsilon)^{\frac{2}{M}} \right)$$

□

Lemma 2.1. *[Belkin and Sinha, 2015, Lemma 2.8] Let $\delta(\epsilon)$ be a positive root of the polynomial $q_\epsilon(x) = q_M(\epsilon)x^M + \ldots + q_0(\epsilon)$, where each $q_i(\epsilon)$ is a polynomial in ϵ. Assume also that $\lim_{\epsilon \to 0} \delta(\epsilon) = 0$. Then there exists a constant $C > 0$ such that $\delta(\epsilon) < C(\epsilon)^{\frac{1}{M}}$.*

The proof below is in general the same as in the original paper, but included here for the sake of completion.

Proof. We can write $q_\epsilon(x) = \epsilon P(x, \epsilon) + Q(x)$, where $Q(x)$ is a polynomial in x and $P(x, \epsilon)$ is a polynomial in x and ϵ. Since $\delta(\epsilon)$ is a root of $q_\epsilon(x)$, we have

$$\epsilon P(\delta(\epsilon), \epsilon) + Q(\delta(\epsilon)) = 0.$$

Note that $\delta(0) = 0$. It follows that from the above equation, $Q(0) = 0$. This allows us to write the polynomial $Q(x)$ as $Q(x) = x^j Q_1(x)$, $j > 0$, so that $Q_1(x) \neq 0$.

On the other hand, for x that is small enough, we have $|Q(x)| \geq \frac{1}{2}|Q_1(0)x^j|$. Since $\lim_{\epsilon \to 0} \delta(\epsilon) = 0$, we can thus write $|Q(\delta(\epsilon))| \geq C_1(\delta(\epsilon)^j)$ and $|P(\delta(\epsilon), \epsilon)| < C_2$ for some $C_1, C_2 > 0$. Combining these two inequalities we have

$$C_1(\delta(\epsilon))^j \leq |Q(\delta(\epsilon))| = \epsilon |P(\delta(\epsilon), \epsilon)| < C_2 \epsilon$$

Since $j \leq M$, we have $\delta(\epsilon) < C(\epsilon)^{\frac{1}{M}}$ for some constant $C > 0$. □

2.3 Polynomial Learnability

We first provide a high-level description of how we plan to establish the polynomial learnability for m_0-identifiable K-MNL models and then discuss the details in each subsections.

We will be working with three types of moments, namely

True moments : based on the true parameters, denoted by $M_i(\theta|\mathcal{S}) = \sum_{x \in \mathcal{S}} x^i p_\theta^{\mathcal{S}}$

Observed moments : empirical, computed by $\widehat{M_i}(\theta|\mathcal{S}) = \sum_{x \in \mathcal{S}} x^i \hat{p}_\theta^{\mathcal{S}}$ where $\hat{p}_\theta^{\mathcal{S}}$ is the observed aggregated distribution over the alternatives in $\mathcal{S}$; we will use θ_e to denote the parameter values corresponding to the empirical moment values

Estimated moments : computed moments based on a given set of estimates $\hat{\theta}$, i.e. $M_i(\hat{\theta}|\mathcal{S}) = \sum_{x \in \mathcal{S}} x^i p_{\hat{\theta}}^{\mathcal{S}}$

The idea is as follows:

1. Sample enough data to ensure $\widehat{M}_i(\theta|\mathcal{S})$ is close to $M_i(\theta|\mathcal{S})$

2. Use Theorem 2.2 to show θ_e is close to θ

3. Design an algorithm to make sure $\widehat{M}_i(\theta|\mathcal{S})$ is close to $M_i(\hat{\theta}|\mathcal{S})$ and $\hat{\theta}$ is close to θ_e

4. Achieve $M_i(\hat{\theta}|\mathcal{S})$ is close to $M_i(\theta|\mathcal{S})$ and $\hat{\theta}$ is close to θ

5. Return $\hat{\theta}$ as estimators

For the first step above, we quantify the notion of *enough data.*

Proposition 2.3 (Concentration of Moments). *Let $p_\theta^{\mathcal{S}} \in \mathbb{R}^s$, $\theta \in \Theta^s \subseteq \mathbb{R}^{K(s+1)}$, be a family of MMNL models over a given slate $\mathcal{S} \subseteq [m]$ of size s. Let $X_1, X_2, \ldots, X_n$ be i.i.d random variables sampled from $p_\theta^{\mathcal{S}}$. Let $\widehat{M}(\theta|\mathcal{S})$ be the empirical moments of the samples drawn. Then given sample size $n \geq \frac{s^{2s+1}}{\epsilon^2\delta}$, for any $\epsilon > 0$ and $0 < \delta < 1$, $\left|\widehat{M}_i(\theta|\mathcal{S}) - M_i(\theta|\mathcal{S})\right| \leq \epsilon$ with probability greater than $1 - \delta$ for $i \leq s$.*

Proof. Recall that the i-th raw moment $M_i(\theta|\mathcal{S}) = \mathbb{E}\left[X^i|\mathcal{S}\right] = \sum_{x\in\mathcal{S}} x^i p_\theta^{\mathcal{S}}$. The empirical moments $\widehat{M}_i(\theta|\mathcal{S}) = \sum_{x\in\mathcal{S}} x^i \hat{p}_\theta^{\mathcal{S}}(x)$, where $\hat{p}_\theta^{\mathcal{S}}$ is the empirical aggregated logit vector or distribution over all the alternatives in $\mathcal{S}$, which can be computed as $\hat{p}_\theta^{\mathcal{S}}(x) = \frac{1}{n}\sum_{j=1}^n \mathbb{1}_{[X_j=x]}$. Note that $\widehat{M}_i(\theta|\mathcal{S})$ is a random variable since it is a function of $X_1, X_2, \ldots, X_n$. In particular, we can write $\widehat{M}_i(\theta|\mathcal{S}) = \dfrac{1}{n}\displaystyle\sum_{j=1}^n X_j^i$.

The expectation of the empirical moments is equal to the true moments since $\mathbb{E}\left[\widehat{M}_i(\theta|\mathcal{S})\right] = \dfrac{1}{n}\displaystyle\sum_{j=1}^n \mathbb{E}\left[X_j^i\right] = M_i(\theta|\mathcal{S})$, for all i. On the other hand, the variance of the empirical moments is upper bounded:

$$\begin{aligned}
\operatorname{Var}\left(\widehat{M_i}(\theta|\mathcal{S})\right) &= \frac{1}{n}\operatorname{Var}\left(X_j^i\right) \\
&= \frac{1}{n}\left(\mathbb{E}\left[X_j^{2i}\right] - \mathbb{E}^2\left[X_j^i\right]\right) \\
&\leq \frac{1}{n}\mathbb{E}\left[X_j^{2i}\right] \\
&= \frac{1}{n}\sum_{x\in\mathcal{S}} x^{2i} p_\theta^{\mathcal{S}}(x) \\
&\leq \frac{s^{2i}}{n}
\end{aligned}$$

This means $\forall i \leq s$, we have $\operatorname{Var}\left(\widehat{M_i}(\theta|\mathcal{S})\right) \leq \frac{s^{2s}}{n}$.

Finally, we can use Chebyshev's inequality to compute an upper bound on the discrepancy of the empirical moments and the true moments:

$$\begin{aligned}
\mathbb{P}\left(\left|\widehat{M_i}(\theta|\mathcal{S}) - M_i(\theta|\mathcal{S})\right| > \epsilon\right) &= \mathbb{P}\left(\left|\widehat{M_i}(\theta|\mathcal{S}) - \mathbb{E}\left[\widehat{M_i}(\theta|\mathcal{S})\right]\right| > \epsilon\right) \\
&\leq \frac{\operatorname{Var}\left(\widehat{M_i}(\theta|\mathcal{S})\right)}{\epsilon^2} \\
&\leq \frac{s^{2s}}{n\epsilon^2}
\end{aligned}$$

Denote the event $\left|\widehat{M_i}(\theta|\mathcal{S}) - M_i(\theta|\mathcal{S})\right| > \epsilon$ as A_i.

$$\begin{aligned}
\mathbb{P}\left(\left|\widehat{M_i}(\theta|\mathcal{S}) - M_i(\theta|\mathcal{S})\right| \leq \epsilon, \forall i\right) &= 1 - \mathbb{P}\left(\exists i \text{ s.t. } \left|\widehat{M_i}(\theta|\mathcal{S}) - M_i(\theta|\mathcal{S})\right| > \epsilon\right) \\
&= 1 - \mathbb{P}\left(\bigcup_{1\leq i\leq s} A_i\right) \\
&\geq 1 - \sum_{i=1}^{s} \mathbb{P}(A_i) \\
&\geq 1 - \delta
\end{aligned}$$

where the last inequality follows from the fact that the number of data samples

$n > \frac{s^{2s+1}}{\epsilon^2 \delta}$. □

Proposition 2.3 first established the fact with a polynomial number of data points, we can approximate the true moments with the empirical moments with high accuracy and high confidence. The next question we will address is how to learn the parameters using the empirical moments. This is done using the grid-search algorithm as shown below.

2.3.1 The Grid Search Algorithm

In this section, we will present the grid-search algorithm with appropriate grid size for learning close-to-ground-truth parameters for MMNL models.

Algorithm 1: Grid Search Algorithm

```
in $\epsilon > 0$, $0 < \delta < 1$, grid size $\frac{1}{N} = O(\frac{\epsilon^t}{\sqrt{K(m+1)}})$, list of slates
  $\mathbb{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_T\}$
for $\tau \leftarrow 1$ to $T$ do
    Draw $x_1, x_2, \ldots, x_n$ samples from $p_\theta^{\mathcal{S}_t}$ and compute $\hat{p}^{\mathcal{S}_t}(j) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{[x_i=j]}$
      with $n \geq \frac{s}{\epsilon^2 \delta}$
end
Let $\theta = \{\frac{1}{N}, \frac{2}{N}, \ldots, \frac{N-1}{N}\}$ for $r = 1, \ldots, K(m+1)$
Random choose $\hat{\theta}$ from $\Theta^{\mathcal{S}}$
for $\theta \in \theta_1 \times \ldots \times \theta_{K(m+1)}$ do
    if $\max_\tau \|\hat{p}^{\mathcal{S}_\tau} - p_\theta^{\mathcal{S}_\tau}\|^2 < \max_\tau \|\hat{p}^{\mathcal{S}_\tau} - p_{\hat{\theta}}^{\mathcal{S}_\tau}\|^2$ then
        $\hat{\theta} = \theta$
end
Output: $\hat{\theta}$
```

Figure 2-1 provides an intuitive illustration of the idea we introduced at the beginning of this section and the grid search algorithm. With enough data sample, we can obtain a set of empirical moments that are close to the true moments. By Theorem

2.2, there exists some θ_e that is close to θ in the θ-space. Now assume that Algorithm 1 returns $\hat{\theta} = \theta_A$ as the output. If we have an appropriate grid size, we can ensure that $\hat{\theta}$ and θ_e is also close to each other. In turn, we can obtain the estimated moments using the estimated θ. The estimated moments will be close to the empirical moments, hence close to the true moments. Applying Theorem 2.2 again, we have the upper bound for $\|\hat{\theta} - \theta\|$.

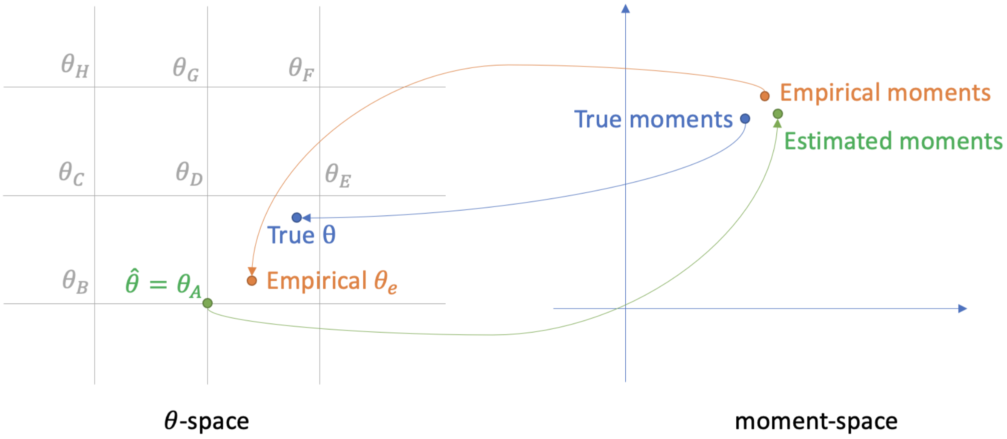


Figure 2-1: Grid search algorithm illustration

Next, we will discuss how to choose the appropriate grid size.

Proposition 2.4 (Moments Upper bound). *Let $p_{\theta\theta\in\Theta}$ be a family of MMNL models over an m-item universe. For a given slate $\mathcal{S} \subset [m]$ with $|\mathcal{S}| = s$, there exists some constant $C > 0$, bounded above, such that*

$$\sum_{i=1}^{s} |M_i(\theta_1|\mathcal{S}) - M_i(\theta_2|\mathcal{S})|^2 < C\|\theta_1 - \theta_2\|^2$$

Proof. We can show this using the mean value theorem:

$$\frac{|M_i(\theta_1|\mathcal{S}) - M_i(\theta_2|\mathcal{S})|}{|\theta_1 - \theta_2|} \leq \sup_{\theta\in\Theta} \|\text{grad}(M_i^{\mathcal{S}})(\theta)\|$$

We will analyze the element in $\text{grad}(M_i)(\theta|\mathcal{S})$. It contains the following three types of gradients and we will show each is bounded above by some constant:

- $$\begin{aligned}\frac{d\,\sum_{x\in\mathcal{S}}\sum_k w_k \frac{a_{kx}}{\sum_{i\in\mathcal{S}} a_{ki}} x^i}{d\,a_{kj}} &= \sum_{x\in\mathcal{S}}\left(\frac{w_k}{\sum_{i\in\mathcal{S}} a_{ki}} - \frac{w_k a_{kj}}{\left(\sum_{i\in\mathcal{S}} a_{ki}\right)^2}\right) x^i \\ &\leq \sum_{x\in\mathcal{S}} w_k \frac{1}{\sum_{i\in\mathcal{S}} a_{ki}} x^i \\ &\leq \sum_{x\in\mathcal{S}} \frac{w_k}{a_k^0} x^i\end{aligned}$$

  where $a_k^0 = \min_{j\in\mathcal{S}, a_{kj}>0} a_{kj}$, i.e., the smallest a_{kj} value in the slate for mixture k such that probability of choosing alternative j is non-zero. Note that $a_k^0 > 0$ for a given slate, otherwise, all alternatives in $\mathcal{S}$ will not be chosen, hence making it a meaningless slate.

- $$\begin{aligned}\frac{d\,\sum_{x\in\mathcal{S}} w_k \frac{a_{kj}}{\sum_{i\in\mathcal{S}} a_{ki}} x^i}{d\,a_{ki}} &= -\frac{w_k a_{kj}}{\left(\sum_{i\in\mathcal{S}} a_{ki}\right)^2} \\ &\geq -\sum_{x\in\mathcal{S}} \frac{w_k}{\left(\sum_{i\in\mathcal{S}} a_{ki}\right)^2} x^i \\ &\geq -\sum_{x\in\mathcal{S}} \frac{w_k}{(a_k^0)^2} x^i\end{aligned}$$

- $$\begin{aligned}\frac{d\,\sum_{x\in\mathcal{S}} w_k \frac{a_{kj}}{\sum_{i\in\mathcal{S}} a_{ki}} x^i}{d\,w_k} &= \sum_{x\in\mathcal{S}} \frac{a_{kj}}{\sum_{i\in\mathcal{S}} a_{ki}} x^i \\ &\leq \sum_{x\in\mathcal{S}} \frac{a_{kj}}{(a_k^0)} x^i\end{aligned}$$

Putting everything together, we have

$$\begin{aligned}\|\mathrm{grad}(M_i)(\theta)\| &\leq \sqrt{\sum_{k=1}^{K}\sum_{x\in\mathcal{S}}\left(\left(\frac{w_k}{a_k^0}\right)^2 + \left(\frac{w_k}{(a_k^0)^2}\right)^2 + \left(\frac{a_{kj}}{a_k^0}\right)^2\right) x^i} \\ &\leq \sqrt{KC'}\end{aligned}$$

where C' is an appropriate constant. □

Lemma 2.2 (Grid Search Accuracy). *Let p_θ be a family of MMNL models. Let θ^* be the ground truth parameter values. For any given $\epsilon > 0$, $0 < \delta < 1$, $\hat{\theta}$ generated from Algorithm 1 can achieve $\|\hat{\theta} - \theta^*\| < \epsilon$ with probability at least $1 - \delta$ for the same ϵ and δ used in Algorithm 1.*

Proof. We can show that

$$\begin{aligned}\left|M_i(\hat{\theta}|\mathcal{S}) - M_i(\theta|\mathcal{S})\right| &= \left|M_i(\hat{\theta}|\mathcal{S}) - \widehat{M}_i(\theta|\mathcal{S}) + \widehat{M}_i(\theta|\mathcal{S}) - M_i(\theta|\mathcal{S})\right| \\ &\leq \left|M_i(\hat{\theta}|\mathcal{S}) - \widehat{M}_i(\theta|\mathcal{S})\right| + \left|\widehat{M}_i(\theta|\mathcal{S}) - M_i(\theta|\mathcal{S})\right| \qquad (2.8) \\ &\leq C_1 \cdot \epsilon^t + C_2 \cdot \epsilon^t\end{aligned}$$

The second inequality holds because

- For the first component in Equation (2.8), it is a direct result of the grid search algorithm. With small enough grid size, we can achieve

  $$\begin{aligned}\|\theta_1 - \theta_2\| &= \sqrt{\sum_{i=1}^{K(m+1)} \frac{\epsilon^{2t}}{K(m+1)}} \\ &= \epsilon^t\end{aligned}$$

  By Proposition 2.4, this tells us that we

  $$\left|M_i(\hat{\theta}|\mathcal{S}) - \widehat{M}_i(\theta|\mathcal{S})\right| \leq C_1 \cdot \epsilon^t$$

- For the second component in Equation (2.8), it holds when we have enough sample as described in Proposition 2.3.

Finally, we apply Theorem 2.2 to get the desired bound on $\|\hat{\theta} - \theta^*\|$, with $t = \frac{M}{2}$. □

Complexity

Since the grid search algorithm divides the parameter space into $O(\frac{\epsilon^t}{\sqrt{K(m+1)}})$ grids, the number of operations needed to complete this process is a polynomial in $\frac{1}{\epsilon^t}$, i.e. poly$(\frac{1}{\epsilon})$.

2.3.2 Main Result

We putting all pieces discussed in this section together and present the main theorem.

Theorem 2.3 (Main result)**.** *Let p_θ be the family of mixed multinomial logit models. If p_θ is identifiable, then, $\forall \epsilon > 0, 0 < \delta < 1$, there exists an algorithm which can generate $\hat{\theta}$ such that $\left\|\hat{\theta} - \theta\right\| \leq \epsilon$ with probability $1 - \delta$ using a polynomial number of data samples $n = poly(\frac{1}{\epsilon}, \frac{1}{\delta})$ drawn from p_θ. The algorithm requires a polynomial number of operations in $\frac{1}{\epsilon}$.*

Proof. In order to learn the MMNL model, we will collect $n > \frac{s^{2s+1}}{\epsilon^{2t}\delta}$ data samples for a given slate $\mathcal{S}_\tau$ of size s. This allows us to achieve $|\widehat{M_i}(\theta|\mathcal{S}) - M_i(\theta|\mathcal{S})| < \epsilon^t$ with probability at least $1 - \delta$.

We then apply the grid search algorithm in the parameter space Θ to obtain an estimate $\hat{\theta}$. By setting the correct grid size, we have $\|\hat{\theta} - \theta_e\| < \epsilon^t$. Then by Proposition 2.4, we have $|M_i(\hat{\theta}|\mathcal{S}) - \widehat{M_i}(\theta_e|\mathcal{S})| < \epsilon^t$ with probability at least $1 - \delta$.

By triangular inequality, we have $|M_i(\hat{\theta}|\mathcal{S}) - M_i(\theta|\mathcal{S})| < C \cdot \epsilon^t$. Finally, according to Theorem 2.2, this gives $\|\hat{\theta} - \theta\| < \epsilon$ with probability at least $1 - \delta$.

□

Slate selection

Finally, we discuss how to choose the set of slates. Since we are assuming that p_θ is m_0-identifiable. This means if we look at a slate $\mathcal{S} = \{i_1, \ldots, i_{m_0}\}$ of size m_0, with enough sample, we can uniquely determine the parameters $\theta^{\mathcal{S}}$ that is ϵ-close to the true parameters with probability at least $1 - \delta$. Therefore, we can create such m_0-size slates like $\mathbb{S} = \{\{1, 2, \ldots, m_0\}, \{1, m_0 + 1, \ldots, 2m_0 - 1\}, \ldots, \{1, \ldots, m, 2, \ldots, \}\}$, each with alternative 1 as the common alternative in all. Note that if the last one is of size smaller than m_0 when we hit alternative m, we can add other alternatives despite overlapping. In worst case, it requires querying all 2^{m_0} subslates of each of these slates to learn the parameters $\theta^{\mathcal{S}}$. We need to do this for the $\left\lceil \frac{m-1}{m_0-1} \right\rceil$ elements in $\mathbb{S}$ elements in $\mathbb{S}$.

While we have shown that K-MNL models under the m_0 identifiability assumption are polynomial learnable the order of the polynomial is unknown. We tried implementing the grid search algorithm for the case where $K = 2, m = 4$ and found that it can be very slow. This leads us to develop a more practical algorithm for learning MMNL models, which will be discussed next.

Chapter 3

Learning MMNL with Provable Guarantees - A Data Driven Approach

In this chapter, we will present a new learning framework for the MMNL models. Our algorithm, called Stochastic Subregion Frank-Wolfe (`SSRFW`), aims to learn the parameters from a data-driven perspective while still providing provable guarantees to the estimators. The key difference between `SSRFW` and Chapter 2 lies in that this chapter makes assumptions on the data structure and is a more practical method that can be used in real-world applications.

In Section 3.1, we will formalize the learning problem and identify some key assumptions in our setting. In Section 3.2, we will describe the original Frank-Wolfe approach, analyze its underlying issues and present our framework, which is consisted of two parts: the `SSRFW` algorithm itself and a $\mathcal{Q}$-construction algorithm whose output will be taken as input to the former. Finally, in Section 3.3, we will prove the theoretical properties of `SSRFW` , including its provable guarantees on the parameters as well as the corresponding sample complexity.

3.1 Problem Formulation

Different from the previous chapter, we will resume using the standard MMNL notations as introduced in Chapter 1. Recall the setting where the decision makers need to choose among a set of $[M] = \{1, \ldots, M\}$ items, where each alternative $j \in [M]$ is associated with a feature vector $\boldsymbol{z}_j \in \mathbb{R}^d$. Assume the population is modeled by a MMNL model consisted of K individual MNL mixture components, with decision makers within each component exhibit the same logit parameters. Furthermore, we will focus on a linear utility function with respect to the observed attributes, i.e. $v_{kj} = \boldsymbol{\beta}_k^\top \boldsymbol{z}_j$, where $\boldsymbol{\beta}_k \in \mathbb{R}^d$ represents the decision makers' taste on different item attributes for decision maker of type k. Finally, the mixture weights are denoted as α_k, where $\sum_{k=1}^K \alpha_k = 1$.

Under these settings and RUM, the probability of choosing alternative j for a particular mixture k can be written as

$$q_j(\boldsymbol{\beta}_k) = \frac{\exp(\boldsymbol{\beta}_k^\top \boldsymbol{z}_j)}{\sum_{i=1}^M \exp(\boldsymbol{\beta}_k^\top \boldsymbol{z}_i)}, j \in [M]$$

and the aggregated counterpart for the entire population can be computed as

$$\mathrm{g}_j = \sum_{k=1}^K \alpha_k q_j(\boldsymbol{\beta}_k), j \in [M].$$

When there is no ambiguity, we will simplify the notation $q_j(\boldsymbol{\beta}_k)$ with q_{kj}. We also denote the individual and aggregated *logit vectors* as $\boldsymbol{q}_k = [q_{k0}, \cdots, q_{kM}]^\top \in \mathbb{R}^M, \forall k$ and $\mathbf{g} = [\mathrm{g}_0, \cdots, \mathrm{g}_M]^\top \in \mathbb{R}^M$, respectively. Since these logic vectors characterize categorical distributions among the choice set, we use the term "choice probability vectors" interchangeably.

As discussed in Chapter 1, the setting we adopt here formulates the MMNL into a *latent class model* and is non-parametric. In other words, $\boldsymbol{\beta}_k$'s are not drawn from a parametric prior $f(\boldsymbol{\beta})$ and we are not trying to estimate the parameters for the mixing distribution $f(\boldsymbol{\beta})$. Instead, we assume $\boldsymbol{\beta}$ can take K possible discrete values labeled $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K$, each with probability $\alpha_1, \ldots, \alpha_K$. Our goal is to develop an algorithm that can learn all α_k's and $\boldsymbol{\beta}_k$'s with some theoretical guarantees on the estimators.

3.1.1 Model Assumption

For the MMNL models to be learnable, we need to assume that none of the individual MNL is identical to any of the other mixtures. Let F and G be the cumulative distribution functions (CDF) for two different choice probability distribution $\boldsymbol{q}_k$ and $\boldsymbol{q}_{k'}$, i.e.

$$F(x) = \sum_{j=1}^{x} q_{kj}, \qquad G(x) = \sum_{j=1}^{x} q_{k'j}.$$

Define the Kolmogorov-Smirnov (KS) distance between the two distributions as

$$D_{\text{KS}}(\boldsymbol{q}_k, \boldsymbol{q}_{k'}) = \sup_x |F(x) - G(x)|$$

Formally, we assume the MMNL model to be learned is ϵ-standard, as defined below:

Definition 3.1. *We call a mixed multinomial logit model (MMNL)* $\mathbf{g}$ *ϵ-standard if for a given $\epsilon > 0$, we have*

1. $\min_k \alpha_k \geq \epsilon$
2. $\beta_{kd} \leq \frac{1}{\epsilon}, \forall k, \forall d$
3. $D_{KS}(\boldsymbol{q}_k, \boldsymbol{q}_{k'}) \geq \epsilon, \forall k \neq k'$

Definition 3.1 is adapted from a similar assumption introduced in Moitra and Valiant [2010] for learning Gaussian mixture models. Intuitively, it imposes some regularity to the MMNL model:

1. none of the mixtures has zero weight;
2. all parameters in the model are bounded;
3. each individual logit model has some level of separation from others.

We consider this as a very mild assumption in our setting, as later we will see that we will not predetermine the number of mixtures (K) and if a mixture/component is not present in the observed data (since it has zero weight), the algorithm will simply not learn it. This is also one of the advantages of our algorithm over the traditional approaches, such as EM.

3.1.2 Data Assumption

The SSRFW framework will assume to work with repeated choice data, such as consumer panel or survey data. This is a reasonable assumption in many real-world scenarios, for instance, see Revelt and Train [1998] and Brownstone et al. [2000]. In particular, we assume the data consists of historical choice decisions from the target population of size N. For each time period $t = 1, \ldots, T$, we consider each decision maker i's choice $X_i^{(t)}$ as i.i.d random variables drawn from a categorical distribution with parameters $\boldsymbol{q}_k$ if the decision maker is of type k. This means the support of X_i is $[M]$. Let $x_i^{(t)}$ be the realization of $X_i^{(t)}$ and Y_i^t be the one-hot encoding vector of $x_i^{(t)}$, i.e. $Y_i^t = [0, \cdots, 1, \cdots, 0]^\top \in \mathbb{R}^M$, where $Y_{ij}^t = 1$ if $x_i^{(t)} = j$.

The observed aggregated logit vector for period t is thus the average among all decision makers in the population:

$$\mathbf{y}^t = \frac{1}{N} \sum_{i=1}^{N} Y_i^t \in \mathbb{R}^M$$

Define $\mathcal{P} = \{\boldsymbol{q}(\boldsymbol{\beta}) | \boldsymbol{\beta} \in \mathbb{R}^d\}$, the set of all valid logit vectors for the given choice set $[M]$, and $\overline{\mathcal{P}}$ as its closure. Note that since $\mathbf{g} = \sum_{k=1}^{K} \alpha_k \boldsymbol{q}_k$, by definition we have $\mathbf{g} \in \mathrm{Conv}(\overline{\mathcal{P}})$.

Finally, we state our main learning objective as to minimize the difference between the observed (from data) and theoretical aggregated choice probability vector, as follows

$$\min_{\mathbf{g} \in \mathrm{Conv}(\overline{\mathcal{P}})} \mathcal{L}(\mathbf{g}; \mathbf{y}) \equiv \min_{\mathbf{g} \in \mathrm{Conv}(\overline{\mathcal{P}})} \frac{1}{2} \sum_{t=1}^{T} \left\| \mathbf{g} - \mathbf{y}^t \right\|^2 \tag{3.1}$$

To simplify notation, we omit the observed data terms in the expressions and only use

$\mathcal{L}(\mathbf{g})$ for the loss term. We can expand Eqn. (3.1) with the actual set of parameters:

$$\min_{\alpha_k, \boldsymbol{\beta}_k} \frac{1}{2} \sum_{t=1}^{T} \left\| \sum_{k=1}^{K} \alpha_k \frac{\exp(\boldsymbol{\beta}_k^\top \boldsymbol{z}_j)}{\sum_{i=1}^{M} \exp(\boldsymbol{\beta}_k^\top \boldsymbol{z}_i)} - \mathbf{y}^t \right\|^2$$

Note that this is a highly non-linear optimization problem where the decision variables α_k's and β_k's also entangle together with each other.

While various heuristic methods have been proposed for MMNL models, there does not exist an algorithm that can yield theoretical guarantees on the estimators. As one of the first attempts in this research problem, Jagabathula et al. [2020] designed a Frank-Wolfe framework that aims to establish some convergence properties as a direct result from the subgradient method, which was originally proposed by Marguerite Frank and Philipin Wolfe in 1956 with many additional nice properties proved by Jaggi [2013]. However, we will see that the convergence property in the *original FW* algorithm by Jagabathula et al. [2020] applies to the aggregated choice probabilities, $\mathbf{g}$, instead of the estimators α_k's and $\boldsymbol{\beta}_k$'s.

As many downstream applications require accurate estimations of these parameters and observing the abundant data that are available nowadays which exist in electronic forms, we design a new data-driven approach to learn the MMNL models. The Stochastic Subregion Frank-Wolfe (`SSRFW`) algorithm, inspired by the original FW, has an carefully constructed stochastic feasible region derived from the data, which restricts the search space for candidate choice probability vectors. As a consequence, the generated candidates from the algorithm can be shown to be within an ϵ-ball of the ground truth choice probability vectors with high probability. This allows us to establish provable convergence on the estimators. Roughly speaking, we can say the estimators from `SSRFW`, $\hat{\alpha}_k$ and $\hat{\boldsymbol{\beta}}_k$, satisfy that for any given $\epsilon > 0, 0 < \delta < 1$, there exists some mapping $\pi : [K] \rightarrow [K]$ such that $\mathbb{P}(|\hat{\alpha}_k - \alpha_{\pi(k)}| < \epsilon) \geq 1 - \delta$ and $\mathbb{P}(\|\hat{\boldsymbol{\beta}}_k - \boldsymbol{\beta}_{\pi(k)}\| < \epsilon) \geq 1 - \delta$, $\forall k$. In addition, we will also quantify the sample complexity of the learning algorithm. To the best of our knowledge, this is the first paper to prove these results for MMNL models with an arbitrary number of mixture types and minimal assumptions on the model parameters.

3.2 Stochastic Subregion Frank-Wolfe Algorithm (SSRFW)

3.2.1 Original FW

We will first present the original FW algorithm in Algorithm 2 and illustrate how it works. Then we will discuss the problems with this approach.

```
Algorithm 2: Original Frank-Wolfe (Fully Corrective Update)
  Input: data $\mathbf{y}$, $\mathcal{P}$
  Initialization: $k = 0$; $\boldsymbol{\alpha}^{(0)} = [1]$, a random $\mathbf{g}^{(0)}$
1 while stopping condition not met do
2     $k \leftarrow k + 1$
3     Compute $\boldsymbol{q} = \underset{\boldsymbol{v}\in\mathrm{Conv}(\overline{\mathcal{P}})}{\arg\min} \left\langle \nabla\mathcal{L}\left(\mathbf{g}^{(k-1)};\mathbf{y}\right), \mathbf{y} - \mathbf{g}^{(k-1)} \right\rangle$
4     Compute $\boldsymbol{\alpha}^{(k)} = \underset{\boldsymbol{\alpha}\in\Delta_k}{\arg\min}\, \mathcal{L}\left(\alpha_0^{(k)}\mathbf{g}^{(0)} + \sum_{s=1}^{k} \alpha_s^{(k)}\boldsymbol{q}^{(s)};\mathbf{y}\right)$
5     Update $\mathbf{g}^{(k)} := \alpha_0^{(k)}\mathbf{g}^{(0)} + \sum_{s=1}^{k} \alpha_s^{(k)}\boldsymbol{q}^{(s)}$
6 end
  Output: set of choice prob. $\{\boldsymbol{q}^{(0)}, \dots, \boldsymbol{q}^{(k)}\}$;
          mixture weights $\boldsymbol{\alpha}^{(k)} \in \Delta_k \subset \mathbb{R}^{k+1}$
```

Note on Line 4, Δ_k is the $(k+1)$-simplex, defined as $\Delta_k = \{\boldsymbol{\alpha} \in \mathbb{R}^{k+1} | \sum_{k'=1}^{k+1} \alpha_{k'} = 1\}$.

Each iteration k is consisted of two steps: 1) the *supporting finding step* (Line 3) which searches for a new direction represented by a choice probability vector $\boldsymbol{q}^{(k)}$ (as a new latent class k) via solving a linear optimization subproblem in the search space $\mathrm{Conv}(\overline{\mathcal{P}})$, and 2) the *proportion update step* (Line 4) which updates the mixture proportion $\alpha_i, i = 0, \dots, k$, assigned to all choice probability vectors obtained so far. At the end of each iteration, the aggregated logit vector $\mathbf{g}^k$ is updated with the new direction and new mixture weights, which is needed for computing the gradient for the next iteration. The stopping criteria to the iteration process can be set as a simple convergence checking of $\mathbf{g}^{(k-1)}$ and $\mathbf{g}^{(k)}$ or $\mathbf{g}^{(k)}$ and $\mathbf{y}$, which if occurred, we obtain the final learning outcome $\hat{\mathbf{g}}$ is $\sum_{i=0}^{k} \alpha_k \boldsymbol{q}^{(k)}$. The algorithm above uses a fully-corrective variant of the generic Frank-Wolfe algorithm, yet is still guaranteed to find the optimal solution to the optimization problem for the decision variable $\mathbf{g}$ [Jaggi, 2013].

Note in the above algorithm, after k iterations, we have generated k choice probability vectors $\boldsymbol{q}^{(s)}$, $s = 1, \ldots, k$, all of which will be the output of the learning algorithm. The mixture weight estimator is slightly different. $\boldsymbol{\alpha}^{(k)} \in \mathbb{R}^{k+1}$ represents the mixture weight vector for the k choice probability vectors we have generated so far, where $\alpha_s^{(k)}$ refers to the s-th element in the vector, corresponding to the mixture associated with $\boldsymbol{q}^{(s)}$. In other words, the vector length of $\boldsymbol{\alpha}$ increases by 1 for each iteration and only the final vector $\boldsymbol{\alpha}^{(k)}$ is outputted from the algorithm.

To help better understand the mechanism, we illustrate this process with a simple example as shown in Figure 3-1.

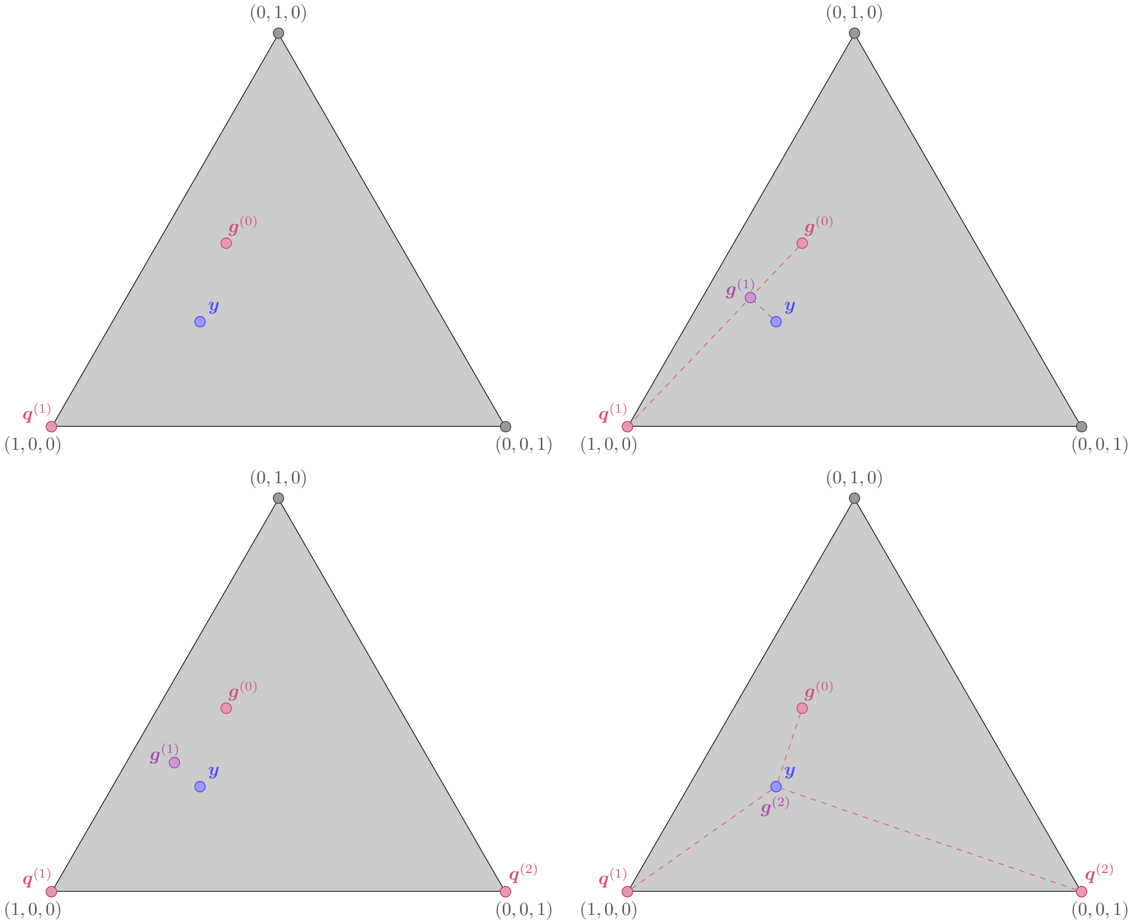


Figure 3-1: Illustration of the Frank-Wolfe algorithm

We start from a random point $\mathbf{g}^{(0)}$ in the feasible region (the shaded area). Enter the first iteration. In the *support finding step*, we find the gradient direction $\boldsymbol{q}^{(1)}$ by solving the linear optimization (first figure). In the *proportion update step*, we redistribute the weights $\alpha^{(0)}$ and $\alpha^{(1)}$ such that the distance between $\mathbf{g}^{(1)} = \alpha^{(0)}\mathbf{g}^{(0)} + \alpha^{(1)}\boldsymbol{q}^{(1)}$ and $\mathbf{y}$ is minimized (second figure). Next, enter the second iteration. We find a new direction again by solving the LP and locate $\boldsymbol{q}^{(2)}$ (third figure). The redistribution of weights will give us $\mathbf{g}^{(2)}$ (fourth figure). The iteration then stops, and the outputs from the algorithm are the logit vectors $\boldsymbol{q}^{(0)}(= \mathbf{g}^{(0)}), \boldsymbol{q}^{(1)}, \boldsymbol{q}^{(2)}$ and the mixture weights, $\alpha^{(0)}$, $\alpha^{(1)}$, $\alpha^{(2)}$.

The problem with the original FW approach is that the feasible region $\mathrm{Conv}(\overline{\mathcal{P}})$ contains the complete set of all logit vectors and their limiting points. The former corresponds to *non-boundary types*, whose choice model can be characterized by a standard MNL choice model, and the latter corresponds to *boundary types*, whose standard MNL parameters become unbounded, resulting infinite utility for some options and zero for the rest. As illustrated in Figure 3-2, the grey area is $\mathcal{P}$; the grey area together with the black boundaries form $\overline{\mathcal{P}}$.

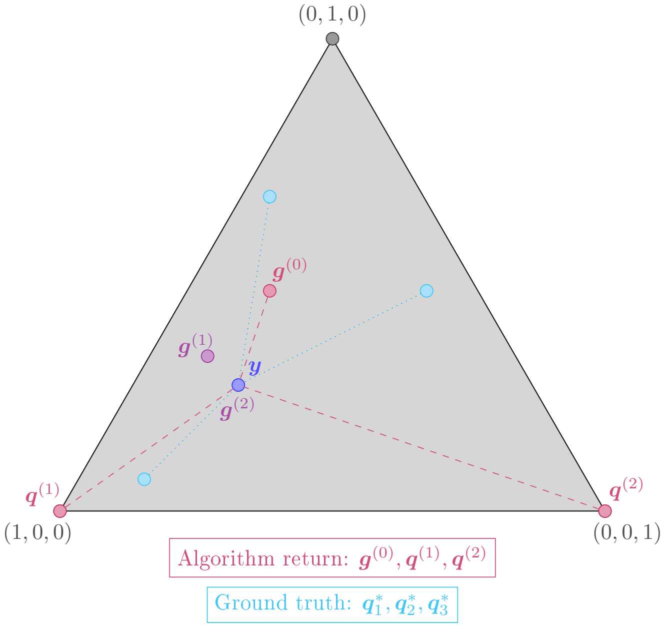

$(0,1,0)$
$\boldsymbol{g}^{(0)}$
$\boldsymbol{g}^{(1)}$
$\boldsymbol{y}$
$\boldsymbol{g}^{(2)}$
$\boldsymbol{q}^{(1)}$
$\boldsymbol{q}^{(2)}$
$(1,0,0)$
$(0,0,1)$
Algorithm return: $\boldsymbol{g}^{(0)}, \boldsymbol{q}^{(1)}, \boldsymbol{q}^{(2)}$
Ground truth: $\boldsymbol{q}_1^*, \boldsymbol{q}_2^*, \boldsymbol{q}_3^*$

Figure 3-2: Intuition: the original FW algorithm

Such broad search space $\mathrm{Conv}(\overline{\mathcal{P}})$ in FW generates mixture compositions with a considerable number of the boundary types. More specifically, this problem is rooted to the linear subproblem solved at each *support finding step*, whose optimal solution always lies on the extreme point (or the sides) of the feasible region which contains many of these limiting choice probability vectors $\in \overline{\mathcal{P}} \setminus \mathcal{P}$ as its extreme points, e.g. $\boldsymbol{q}^{(1)}$ and $\boldsymbol{q}^{(2)}$ in Figure 3-2. In addition, they can be very far from the ground truth. Often, each limiting choice probability vector corresponds to an unbounded $\boldsymbol{\beta}_k$ and cause the learning outcome unusable for downstream applications. Though significant effort is made to justify the legitimacy of these boundary types in the paper, the convergence result only applies to the population's aggregate choice probability, i.e. $\hat{\mathbf{g}} \to \sum_k \alpha_k \boldsymbol{q}_k$, which is an Frank-Wolfe property. In other words, their method fails to recover each of the individual logit models and the corresponding mixture weights accurately.

3.2.2 The SSRFW Algorithm

To remedy the issue present in the original FW and seek convergence guarantees for the actual MMNL estimation problem, we designed the Stochastic Subregion Frank-Wolfe (SSRFW), as shown in Algorithm 3.

Algorithm 3: Stochastic Subregion Frank-Wolfe

```
Input: data $\mathbf{y}$, $\mathcal{Q}$ from Algorithm 4
Initialization: $k = 0$; $\boldsymbol{\alpha}^{(0)} = [1]$, a random $\mathbf{g}^{(0)} = \boldsymbol{q}^{(0)}$ chosen from $\mathcal{Q}$
while stopping condition not met do
    $k \leftarrow k + 1$
    Compute $\boldsymbol{q} = \underset{\boldsymbol{v} \in \mathrm{Conv}(\mathcal{Q})}{\arg\min} \langle \nabla \mathcal{L}\left(\mathbf{g}^{(k-1)}; \mathbf{y}\right), \boldsymbol{v} - \mathbf{g}^{(k-1)} \rangle$
    Update $\mathbf{g}^{(k)} := \alpha_0^{(k)} \mathbf{g}^{(0)} + \sum_{s=1}^{k} \alpha_s^{(k)} \boldsymbol{q}^{(s)}$ Compute
        $\boldsymbol{\alpha}^{(k)} = \underset{\boldsymbol{\alpha} \in \Delta_k}{\arg\min} \mathcal{L}\left(\alpha_0^{(k)} \mathbf{g}^{(0)} + \sum_{s=1}^{k} \alpha_s^{(k)} \boldsymbol{q}^{(s)}\right)$
end
Output: choice prob. $\boldsymbol{q}^{(0)}, \ldots, \boldsymbol{q}^{(k)}$
        mixture weights. $\boldsymbol{\alpha}^{(k)} \in \Delta_k \subset \mathbb{R}^{k+1}$
```

It takes in an additional input $\mathcal{Q}$ from the $\mathcal{Q}$ *Construction Algorithm* (discussed in detail in Section 3.2.3), which is the key to eliminate the possibility of producing *boundary types*. In particular, the new feasible region $\text{Conv}(\mathcal{Q})$ (grey area in Figure 3-3) is a subset of $\text{Conv}(\overline{\mathcal{P}})$, where each element $\boldsymbol{q} \in \mathcal{Q}$ will be learned from data and is guaranteed to be within an ϵ-ball of the true choice probability vector for some mixture type with high probability, as shown in Figure 3-3.

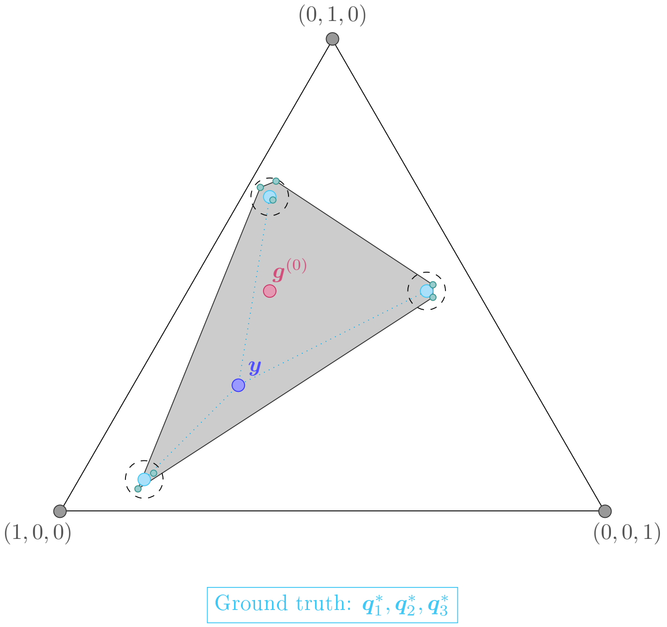

$(0,1,0)$
$\boldsymbol{g}^{(0)}$
$\boldsymbol{y}$
$(1,0,0)$
$(0,0,1)$
Ground truth: $\boldsymbol{q}_1^*, \boldsymbol{q}_2^*, \boldsymbol{q}_3^*$

Figure 3-3: Intuition: the `SSRFW` algorithm

Subsequently, this also ensures the `SSRFW` algorithm to recover the mixture weight for each latent class. Note that the extreme points of $\text{Conv}(\mathcal{Q})$ by definition is a subset of $\mathcal{Q}$. Since we only care about the extreme points of the feasible region, we can safely ignore any points $\in \text{Conv}(\mathcal{Q}) \setminus \mathcal{Q}$ and replace $\text{Conv}(\mathcal{Q})$ with $\mathcal{Q}$ in Algorithm 3.

Since Frank-Wolfe will converge to the optimal solution [Jaggi, 2013], and as we will see in Section 4 that the optimal solution to problem (3.1) is an interior point of the feasible region $\text{Conv}(\mathcal{Q})$, the stopping condition can simply be set as $\|\mathbf{g}^{(k)} - \mathbf{y}\| \leq \epsilon$.

3.2.3 The $\mathcal{Q}$ Construction Algorithm

Our primary goal in this section is to learn the input set $\mathcal{Q}$ for `SSRFW` (Algorithm 3). In particular for this set $\mathcal{Q} = \{\hat{\boldsymbol{q}}_\ell\}_{l=1,\ldots,L}$, we require that $\forall\ \hat{\boldsymbol{q}}_\ell$, there exists some mapping $\pi : [L] \to [K]$ such that $\left\|\hat{\boldsymbol{q}}_\ell - \boldsymbol{q}_{\pi(\ell)}\right\| \leq \epsilon$, where $\boldsymbol{q}_{\pi(\ell)}$ is one of the ground truth logit vectors. Once this property holds, each of the `SSRFW` 's $\boldsymbol{q}$ outputs will also be ϵ-close to some true choice probability vector as the algorithm always selects an extreme point of Conv($\mathcal{Q}$) in each iteration.

The two major components for the $\mathcal{Q}$ Construction Algorithm is first computing a distance score matrix and then creating subsamples that contain decision makers from only one mixture type through random seeding and non-uniform sampling. The final part in this subsection presents the algorithm and discusses the properties of the learning outcomes.

Distance Score Matrix We first define a distance score matrix S, whose element s_{ij} measures the dissimilarity between any two decision makers i and j in term of their choice decisions. Recall that each $x_i^{(t)}$ are realizations of i.i.d random variables with pmf $\boldsymbol{q}_k$ if i is of mixture type k. Denote its associated empirical cumulative distribution function (CDF) as $F_T(x;i)$ when a total of T decisions are observed. In particular,

$$F_T(x;i) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_{\{x_i^{(t)} \leq x\}}, x \in [M]$$

Define the distance score function for each pair of decision makers as

$$s_{ij} := s(i,j) = ||F_T(x;i) - F_T(x;j)||_\infty$$

where $||\cdot||_\infty$ represents the infinity norm.

Subsample Construction We now want to create a number of subsamples such that each subsample of decision makers contains only one mixture type with high probability. Each of these subsamples is generated following a procedure that contains a *random seeding* step, and a *subsampling* step. These steps are repeated L times to create a $\mathcal{Q}$ set of size L.

- Random Seeding.

  This step randomly samples a decision maker $i \in [N]$ from the population and can be done with simple uniform sampling technique. The selected decision maker is called a *seed.* We will generate L seeds and later investigate how large L needs to be in order to cover all mixture types in $\mathcal{Q}$.

- Subsampling Strategy.

  This step generates an index set $I(i)$ for a seed i such that, with high probability, decision makers whose indices belonging to this set are of the same type as i. $I(i)$ is called the *subsample originated from seed i.*

  Given a selected seed i, we first calculate an *accepting probability* $p_{j|i}$ which determines the likelihood of another decision maker j being accepted to the corresponding subsample $I(i)$. It is designed to be a monotonically decreasing function with respect to the distance score we defined earlier:

  $$p_{j|i} = f\left(s(i,j)\right)$$

  A simple example can be $p_{j|i} = 1 - s(i,j)$. Intuitively, when $s(i,j)$ is small, seed i and decision maker j's empirical CDF is close to each other, indicating there is a higher chance that they are of the same type and sharing the same choice probability $\boldsymbol{q}$. Subsequently, $p_{j|i}$ will be larger compared to a large value of $s(i,j')$ for another person j', resulting we accept j with higher probability than j'. This is consistent with our objective that we want each subsample to be composed of decision makers of the same type as the seed.

  For implementation, we first initiate an empty index set $I(i)$ for the selected seed i. Then we repeat the following two steps until we reach a desired subsample size n: 1) draw a random sample from set $[N] \setminus I(i)$, 2) accept this sample into the index set with probability $p_{j|i}$ and reject with $1 - p_{j|i}$.

Algorithm and Properties The remaining step is to obtain the set $\mathcal{Q}$ whose elements $\hat{\boldsymbol{q}}_\ell$ are computed as the average of historical choice decisions from decision makers in the subsamples I_ℓ, i.e. $\hat{\boldsymbol{q}}_\ell = \dfrac{1}{nT}\sum_{i\in I_\ell}\sum_{t=1}^{T} Y_i^{(t)}$. Finally, we present the $\mathcal{Q}$ Construction Algorithm in Algorithm 4.

Algorithm 4: The $\mathcal{Q}$ construction algorithm

```
Input: score matrix $S$, number of subsamples $L$, subsample size $n$
Initialization: $\mathcal{Q} = \texttt{set}()$
for $\ell \leftarrow 1$ to $L$ do
    Choose seed: $i \sim U(0, N)$
    Initiate: $I_\ell = \texttt{set}()$
    while $|I_\ell| \neq n$ do
        $j \leftarrow \texttt{random\_sample}([N] \setminus I_\ell)$
        Generate $u \sim U(0, 1)$
        if $u < p_{j|i}$ then
            $I_\ell.\texttt{add}(j)$
        end
    end
    Compute $\hat{\boldsymbol{q}}_\ell = \frac{1}{nT} \sum_{i \in I_\ell} \sum_{t=1}^{T} Y_i^{(t)}$
    $\mathcal{Q}.\texttt{add}(\hat{\boldsymbol{q}}_\ell)$
end
Output: $\mathcal{Q}$
```

Next, we discuss a few properties of the two algorithms. `SSRFW` returns the logit vectors $\boldsymbol{q}$ instead of the parameters $\boldsymbol{\beta}$. We can simply perform maximum likelihood estimation under the single MNL setting where, if the extreme point corresponding to subsample ℓ is selected during an `SSRFW` iteration, the log-likelihood is

$$l_\ell(\boldsymbol{\beta}) = -\sum_{t=1}^{T} \sum_{i \in I_\ell} \sum_{j \in [M]} Y_{ij}^{(t)} \left(\log \frac{\exp(\boldsymbol{\beta}^\top \boldsymbol{z}_j)}{\sum_{m \in [M]} \exp(\boldsymbol{\beta}^\top \boldsymbol{z}_m)} \right)$$

and we have $\hat{\boldsymbol{\beta}}_\ell = \arg\max_{\boldsymbol{\beta}} \; l_\ell(\boldsymbol{\beta})$. With the assumption of linear utility in $\boldsymbol{\beta}$, we can obtain the optimal solution using MLE (c.f. McFadden and Train [2000]).

In general, we prefer learning $\boldsymbol{\beta}_k$'s compared to $\boldsymbol{q}_k$'s. This makes the algorithm more robust when item attributes change over time which results in change in the $\boldsymbol{q}_k$'s whereas $\boldsymbol{\beta}_k$ values persist. To achieve this, we just need to add two additional steps after the index set is created:

- **MNL parameter estimation:** apply the MLE step outlined above; create $\mathcal{B} = \{\hat{\boldsymbol{\beta}}_\ell\}_{\ell=1,\ldots,L}$

- **Map $\mathcal{B}$ to $\mathcal{Q}$:** fix item attributes of current interest, calculate $\mathcal{Q} = \{\hat{\boldsymbol{q}} | \hat{\boldsymbol{q}} = \boldsymbol{q}(\hat{\boldsymbol{\beta}}_\ell), \hat{\boldsymbol{\beta}}_\ell \in \mathcal{B}\}$

Next, stochasticity in our algorithm comes from two parts, namely the realization of each decision makers' choices, as well as the randomness in the process of generating subsamples. Yet we will show that under appropriate assumptions, `SSRFW` can still recover all individual mixture parameters with high probability.

In addition, we do not impose any restriction on the subsamples created during $\mathcal{Q}$ construction to be mutually exclusive. As long as they only contain one single mixture type (i.e., with high purity), it creates a legitimate estimation of $\hat{\boldsymbol{q}}$ or $\hat{\boldsymbol{\beta}}$ with MLE.

Last but not least, this is a more robust strategy than clustering algorithms to segment the population. Not only do we not require the number of mixtures K as a hyper-parameter, each decision maker's data being used more than once for the estimation can be thought of as a special bootstrap mechanism that has custom weights tailored to our objective in creating homogeneous subsamples.

3.3 Theory of the `SSRFW` Algorithm

We first state the main result of our algorithm.

Theorem 3.1. *Let $\mathbf{g} = \sum_{k=1}^{K} \alpha_k \boldsymbol{q}_k$ be a mixed multinomial logit (MMNL) model over a set of M items. Assume $M \geq K$. For any $\epsilon > 0$, $0 < \delta < 1$, Algorithm 3 outputs an MMNL $\hat{\mathbf{g}} = \sum_{k=1}^{K'} \hat{\alpha}_k \hat{\boldsymbol{q}}_k$ where $K' \geq K$ such that, with probability $\geq 1 - \delta$, there exists a many-to-one mapping $\pi : k' \mapsto k, k' \in [K'], k \in [K]$ such that*

$$\left\| \hat{\boldsymbol{q}}_{k'} - \boldsymbol{q}_{\pi(k')} \right\| \leq \epsilon, \forall\ k',$$

and

$$\left| \sum_{k:\pi(k')=k} \hat{\alpha}_{k'} - \alpha_k \right| \leq \epsilon, \forall\ k.$$

The number of samples required by Algorithm 1 is $\mathcal{O}(\frac{1}{\epsilon^2} \log(\frac{1}{\delta}))$.

We will divide the proof into two parts: 1) how to derive the provable convergence from the properties of Frank-Wolfe and the stochastic subregion construction and 2) how to analyze the sample complexity.

3.3.1 Provable Convergence

Assume K' is the total number of iterations that the algorithm has performed before reaching the stopping criteria. To simplify notation, we denote the generated outcome from k-th iteration $\boldsymbol{q}^{(k)}$ in `SSRFW` as $\hat{\boldsymbol{q}}_k$ to indicate they are the estimators and the final updated mixture weight corresponding to $\boldsymbol{q}^{(k)}$, $\alpha_k^{(K')}$, as $\hat{\alpha}_k$. In other words, we have $\mathbf{g}^{\texttt{SSRFW}} = \sum_{k'=1}^{K'} \hat{\alpha}_{k'} \hat{\boldsymbol{q}}_{k'}$, We use a similar notation $\mathbf{g}^{\texttt{FW}}$ for the original FW approach.

We outline the proof sketch for Theorem 3.1 in Figure 3-4, which shows how we break down to smaller components and prove them individually. Note that all the statements hold in a provable way, meaning it happens with probability $1 - \delta$ given enough data, where the number of data points required is a function of both δ and ϵ, which we will discuss in the following section.

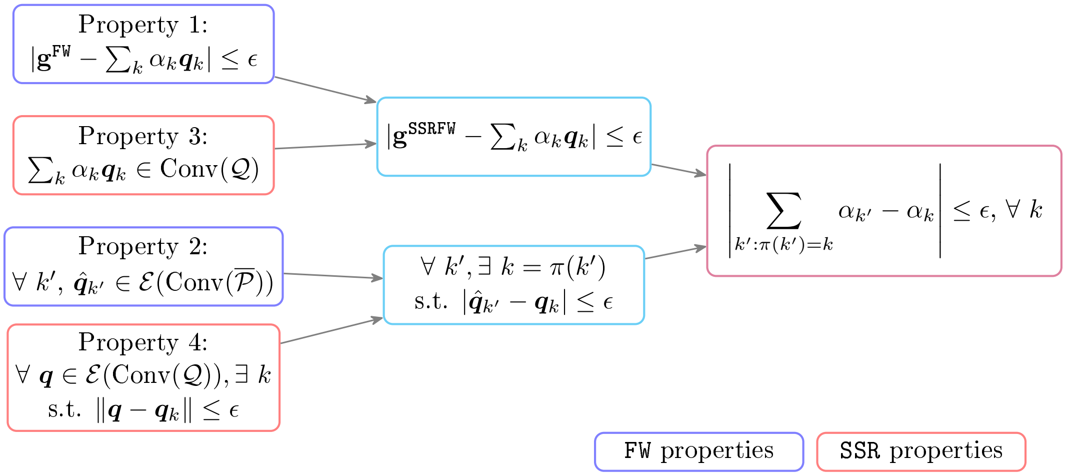

Property 1:
$|\mathbf{g}^{\texttt{FW}} - \sum_k \alpha_k \boldsymbol{q}_k| \leq \epsilon$
Property 3:
$\sum_k \alpha_k \boldsymbol{q}_k \in \mathrm{Conv}(\mathcal{Q})$
Property 2:
$\forall\ k',\ \hat{\boldsymbol{q}}_{k'} \in \mathcal{E}(\mathrm{Conv}(\overline{\mathcal{P}}))$
Property 4:
$\forall\ \boldsymbol{q} \in \mathcal{E}(\mathrm{Conv}(\mathcal{Q})), \exists\ k$ s.t. $\|\boldsymbol{q} - \boldsymbol{q}_k\| \leq \epsilon$
$|\mathbf{g}^{\texttt{SSRFW}} - \sum_k \alpha_k \boldsymbol{q}_k| \leq \epsilon$
$\forall\ k', \exists\ k = \pi(k')$ s.t. $|\hat{\boldsymbol{q}}_{k'} - \boldsymbol{q}_k| \leq \epsilon$
$\left| \sum_{k':\pi(k')=k} \alpha_{k'} - \alpha_k \right| \leq \epsilon, \forall\ k$
FW properties
SSR properties

Figure 3-4: Proof sketch of Thoerem 3.1

In Figure 3-4, Property 1 and 2 (purple boxes) are adapted from existing results of the Frank-Wolfe framework [Jaggi, 2013]. We prove Property 1 in Lemma 3.1. Property 2 follows from the fact that each of the $\hat{\boldsymbol{q}}_{k'}$, obtained by solving a linear optimization problem, has to be an extreme point (denoted by $\mathcal{E}(\cdot)$) of the feasible region.

On the other hand, Property 3 and 4 (red boxes) hold as a consequence of the stochastic subregion we constructed in Algorithm 4. Property 3 claims that with high probability, the ground truth aggregated choice probability vector is still an interior point of the shrunken feasible region Conv($\mathcal{Q}$). We will prove this using Corollary 3.2. Together with Property 1, this tells us that $|\mathbf{g}^{\texttt{SSRFW}} - \sum_k \alpha_k \boldsymbol{q}_k|\leq \epsilon$. That is to say the aggregated choice probability vector of `SSRFW` also converges to the ground truth. To achieve Property 4 from the $\mathcal{Q}$ construction algorithm, we need to supply enough of data, the details of which will be discussed in the sample complexity part. Combining Property 2 and 4, we have $\forall\ k', \exists\ k = \pi(k')$ s.t. $|\hat{\boldsymbol{q}}_{k'} - \boldsymbol{q}_k|\leq \epsilon$. Finally, by doing some algebraic manipulation, we can obtain $|\sum_{k':\pi(k')=k} \alpha_{k'} - \alpha_k|\leq \epsilon$.

In particular, Lemma 3.1 and Corollary 3.2 established Property 1 and 3 as shown in the figure, which in turn result in $|\mathbf{g}^{\texttt{SSRFW}} - \sum_k \alpha_k \boldsymbol{q}_k| \leq \epsilon$. On the other hand, recall that each of the $\hat{\boldsymbol{q}}_{k'}$ is an extreme point (denoted by $\mathcal{E}(\cdot)$) of the feasible region from the Frank-Wolfe framework (Property 2). Together with the property that each element in $\mathcal{Q}$ is ϵ-close to some true choice probability vector (Property 4) as a direct result of the $\mathcal{Q}$ construction algorithm, we have $\forall\ k', \exists\ k = \pi(k')$ s.t. $|\hat{\boldsymbol{q}}_{k'} - \boldsymbol{q}_k| \leq \epsilon$. Finally, by doing some algebraic manipulation, we can obtain $\left|\sum_{k':\pi(k')=k} \alpha_{k'} - \alpha_k\right| \leq \epsilon$.

Lemma 3.1. *Denote* $\mathbf{g}^*$ *as the optimal solution to the Stochastic Subregion Frank-Wolfe algorithm and* $\mathbf{g}^{(k)}$ *denote the* k*-th iterate generated by Algorithm 1. Then*

$$\mathcal{L}(\mathbf{g}^{(k)}) - \mathcal{L}(\mathbf{g}^*) \leq \frac{4}{k+2}$$

for all $k \geq K$.

Proof. This Lemma follows directly from the existing results of the original Frank-Wolfe algorithm and its variants [Jaggi, 2013], which states that for an optimization problem $\min_{x\in\mathcal{D}} f(\mathbf{x})$ where f is a convex and continuously differentiable function and that the domain $\mathcal{D}$ is a compact convex set of any vector space, then for each

$k \geq 1$, the iterates $\mathbf{x}^{(k)}$ of the fully-corrective Frank-Wolfe algorithm satisfy:

$$f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^*) \leq \frac{2 \cdot C_f}{k+2} \tag{3.2}$$

where C_f, defined as

$$C_f := \sup_{\substack{\boldsymbol{x},\boldsymbol{s} \in \mathcal{D} \\ \gamma \in [0,1] \\ \boldsymbol{r} = \boldsymbol{x} + \gamma(\boldsymbol{s} - \boldsymbol{x})}} \frac{2}{\gamma^2} \left(f(\boldsymbol{r}) - f(\boldsymbol{x}) - \langle \nabla f(\boldsymbol{x}), \boldsymbol{r} - \boldsymbol{x} \rangle \right),$$

is the *curvature constant*, which measures the "non-linearity" of function f over domain $\mathcal{D}$. The type of the Frank-Wolfe we use in Algorithm 3 is precisely the fully-corrective variant in that we optimize for α's in each iteration.

Claim 3.1. $\mathcal{L}(\mathbf{g}; \mathbf{y}) = ||\mathbf{g} - \mathbf{y}||^2$ *is a twice differentiable convex function. Conv*$(\mathcal{Q})$ *is a compact convex set.*

The first statement in Claim 3.1 is true by definition. The second statement can be shown by observing that $\mathcal{Q}$ is a finite set, hence compact, followed by the fact that convex hulls of compact set are compact.

For squared loss function $\mathcal{L}$ used in our model, Jagabathula et al. [2020] proved that $C_{\mathcal{L}} \leq 2$. The result of Lemma 3.1 follows by plugging $C_{\mathcal{L}}$ into Equation 3.2.

□

Note that with enough sample, we also have $\|\mathbf{y} - \sum_{k=1}^{K} \alpha_k \boldsymbol{q}_k\| \leq \epsilon$ by the law of large numbers, which leads to $\mathbf{g}^{\texttt{SSRFW}}$ converges to $\sum_{k=1}^{K} \alpha_k \boldsymbol{q}_k$ with high probability. This shows that Frank-Wolfe can reach any tolerance level ϵ with enough number of iterations by setting appropriate stopping criteria.

Lemma 3.2 (Wendel [1962])**.** *If $X_1, \ldots, X_n$ are i.i.d. random points in $\mathbb{R}^d$ whose distribution is symmetric with respect to the center O and assigns measure zero to every hyperplane through O, then*

$$P_n^{(d)}(O \in Conv\{X_1, \ldots, X_n\}) = 1 - \frac{1}{2^{n-1}} \sum_{k=0}^{d-1} \binom{n-1}{k}$$

Lemma 3.2 is an interesting result from stochastic geometry, which states that if we randomly sample n points in a d-dimensional ball, the probability that the convex hull formed using these points contains the center point can be computed using the above formula.

Corollary 3.1.
$\lim_{n\to\infty} P_n^{(d)}(O \in \mathit{Conv}\{X_1, \ldots, X_n\}) = 1$

Proof. When $n \geq 2d-1$, $\binom{n-1}{k}$ is a monotonically increasing function of k for $k = 0, \ldots, d-1$. We then have

$$\sum_{k=0}^{d-1} \binom{n-1}{k} \leq d\binom{n-1}{d-1} \leq d\frac{(n-1)^d}{(d-1)!}$$

Therefore, when $n \geq 2d-1$,

$$P_n^{(d)}(O \in \mathrm{Conv}\{X_1, \ldots, X_n\}) = 1 - \frac{1}{2^{n-1}} \sum_{k=0}^{d-1} \binom{n-1}{k} \geq 1 - \frac{d}{(d-1)!} \frac{(n-1)^d}{2^{n-1}}$$

Since $\lim_{n\to\infty} \frac{(n-1)^d}{2^{n-1}} = 0$, we get

$$\lim_{n\to\infty} P_n^{(d)}(O \in \mathrm{Conv}\{X_1, \ldots, X_n\}) = 1$$

□

Corollary 3.2.
With high probability, $\mathit{Conv}(\{\boldsymbol{q}_k\}_{1,\ldots,K}) \subseteq \mathit{Conv}(\mathcal{Q})$

The proof of Corollary 3.2 will be included in the proof of Theorem 3.1 in Subsection 3.3.3. We will also quantify what "high probability" it is referring to. Corollary 3.2 establishes the fact that $\mathbf{g} = \sum_{k=1}^{K} \alpha_k \boldsymbol{q}_k \in \mathrm{Conv}(\{\boldsymbol{q}_k\}_{k=1,\ldots,K}) \subseteq \mathrm{Conv}(\mathcal{Q})$ with high probability. We illustrate this idea in Figure 3-5 for an intuitive understanding.

For each mixture type k, with enough data points, we have with probability $1-\delta$, $\forall \ell$ such that $\pi(\ell) = k$, $\hat{\boldsymbol{q}}_\ell$ is within an ϵ-ball of $\boldsymbol{q}_k$ given sufficient number of samples. In addition, according to Lemma 3.2, with high probability, such convex hull (small

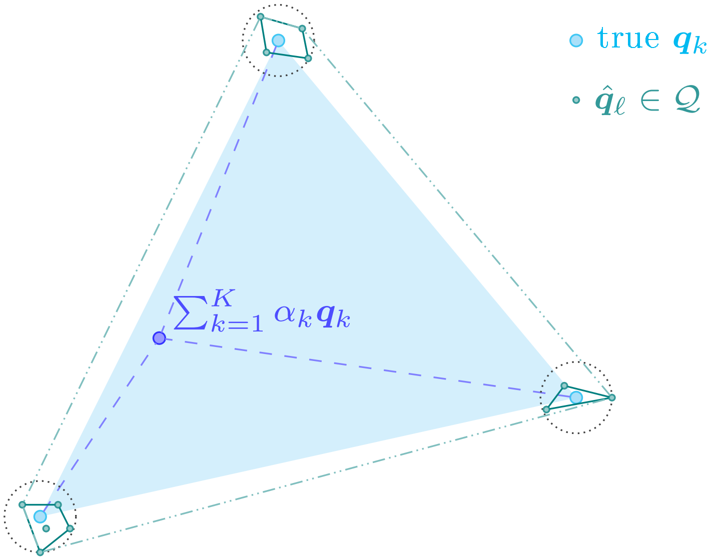

true $\boldsymbol{q}_k$
$\hat{\boldsymbol{q}}_\ell \in \mathcal{Q}$
$\sum_{k=1}^K \alpha_k \boldsymbol{q}_k$

Figure 3-5: Constructed convex hull using logit vectors generated from subsamples

regions with green firm lines for each k) contains the ground truth choice probability vector $\boldsymbol{q}_k$, i.e. $\boldsymbol{q}_k \in \text{Conv}(\{\boldsymbol{q}_\ell \in \mathcal{Q} | \pi(\ell) = k\})$. Since the true aggregated choice probability $\sum_{k=1}^K \alpha_k \boldsymbol{q}_k$ is a convex combination of $\boldsymbol{q}_k$'s (so it is in the blue shaded region) and $\bigcup_k \text{Conv}(\{\boldsymbol{q}_\ell \in \mathcal{Q} | \pi(\ell) = k\}) \subset \text{Conv}(\mathcal{Q})$, we have $\text{Conv}(\mathcal{Q})$ encloses the blue region and $\sum_{k=1}^K \alpha_k \boldsymbol{q}_k$ is an interior point of $\text{Conv}(\mathcal{Q})$.

Next, we move on to discuss the sample complexity and how that establishes Property 4, before we prove the main result.

3.3.2 Sample Complexity

Recall we use T as the number of choice data records for each decision maker. Denote $T_{\min}$ as the minimum number required and $T_{\min} \geq 1$.

Define *in-types* to be the decision makers who share the same type as the seed, and *out-types* to be the ones of different types. We focus on the following three properties:

- probability of accepting the in-types,
- probability of rejecting the out-types,
- probability of covering all mixture types.

Consider a seed i, with type $k_i = k$. Assume there is a total of m_k decision makers in the population that is of type k.

Accepting In-Types. Denote the empirical choice probability for the seed i as F_n and that for another decision maker j as G_m, with n and m representing the number of independent price experiments from i and j, respectively. Recall that $s(i,j) = ||F_n(x) - G_m(x)||_\infty = \sup_x |F_n(x) - G_m(x)|$. For exposition simplicity, we use F and G to denote $F(x)$ and $G(x)$ in the subsequent presentation if no confusion incurred.

We first extend the Dvoretzky-Kiefer-Wolfowitz (DKW) inequality with the following Lemma, where both CDFs in the inequality are empirical.

Lemma 3.3. *Let F_n and G_m be independent empirical distribution based on m and n i.i.d. samples drown from a common cumulative distribution $F(\cdot)$. Denote $min\{m,n\}$ as T_{min}. We have*

$$\mathbb{P}\left(\sup_x |F_n(x) - G_m(x)| > \epsilon\right) \leq 4\exp\left(-\frac{1}{2}T_{\min}\epsilon^2\right)$$

Proof. Lemma 3.3 differs from the Dvoretzky-Kiefer-Wolfowitz (DKW) inequality in that it investigates the tail probability of the maximum difference between *two empirical* distributions. By DKW, we know that

$$\mathbb{P}\left(\sup_x |F_n(x) - F(x)| > \epsilon\right) \leq 2\exp\left(-2n\epsilon^2\right)$$

$$\mathbb{P}\left(\sup_x |G_m(x) - F(x)| > \epsilon\right) \leq 2\exp\left(-2m\epsilon^2\right)$$

if F_n and G_m are empirical distributions of samples drawn from their true distribution function F and G respectively. We can show

$$\begin{aligned} & \mathbb{P}\left(\sup_x |F_n - F| + \sup_x |G_m - F| > \epsilon\right) \qquad (3.3) \\ \leq\ & 1 - \mathbb{P}\left(\sup_x |F_n - F| \leq \frac{\epsilon}{2} \cap \sup_x |G_m - F| \leq \frac{\epsilon}{2}\right) \\ \leq\ & 1 - \left(1 - 2\exp\left(-\frac{1}{2}n\epsilon^2\right)\right)\left(1 - 2\exp\left(-\frac{1}{2}m\epsilon^2\right)\right) \\ \leq\ & 4\exp\left(-\frac{1}{2}T_{\min}\epsilon^2\right) \end{aligned}$$

where the first inequality makes use of the fact that $\sup_x |F_n - F| + \sup_x |G_m - F| > \epsilon$ implies that either $\sup_x |F_n - F| > \frac{\epsilon}{2}$ or $\sup_x |G_m - F| > \frac{\epsilon}{2}$. The second inequality comes from the independent assumption between F_n and G_m. On the other hand, we also have

$$\begin{aligned} & \sup |F_n - F| + \sup |F - G_m| \qquad (3.4) \\ \geq\ & \sup |F_n - F| + |F - G_m| \\ \geq\ & \sup |F_n - F + F - G_m| \\ =\ & \sup |F_n - G_m| \end{aligned}$$

Combining (3.3) and (3.4), we can obtain

$$\begin{aligned} & \mathbb{P}(\sup_x |F_n(x) - G_m(x)| > \epsilon) \\ \leq\ & \mathbb{P}(\sup_x |F_n(x) - F(x)| + \sup_x |G_m(x) - F(x)| > \epsilon) \\ \leq\ & 4\exp\left(-\frac{1}{2}T_{\min}\epsilon^2\right). \end{aligned}$$

□ □

Theorem 3.2 (Sample Complexity I). *Assume i and j are of the same type. Define $p_{j|i} = 1 - s(i,j)$. $\forall \delta > 0, \epsilon > 0$, we can achieve $p_{j|i} > 1 - \epsilon$ with probability at least $1 - \delta$ with sample size $T_{\min} = \mathcal{O}(\frac{1}{\epsilon^2}\log\left(\frac{1}{\delta}\right))$.*

Proof. For simpler notation, denote $s(i,j) = s$. Since i and j are of the same consumer type k, F_n and G_m are empirical distributions based on n and m samples drawn from the same distribution $\boldsymbol{q}_k$. According to Lemma 3.3,

$$\mathbb{P}(p_{j|i} > 1 - \epsilon) = \mathbb{P}(s < \epsilon) \geq 1 - 4\exp(-\frac{1}{2}T_{\min}\epsilon^s)$$

Let $\delta = 4\exp(-\frac{1}{2}T_{\min}\epsilon^2)$, we have $T_{\min} = \mathcal{O}(\frac{1}{\epsilon^2}\log\frac{1}{\delta})$. □

Theorem 3.2 indicates that with sufficient amount of data, there is a high probability to accept a in-type—who shares the same type as the seed—into the subsample if it is chosen from the population after the random draw.

Rejecting Out-Types. Assume a decision maker j is of a different type from the seed i. We use F and G to denote their CDFs. Let F_n and G_m represent the corresponding empirical CDFs based on samples drawn from the two different distributions with sample size n and m, respectively. $s(i,j)$ is defined the same as before.

Theorem 3.3 (Sample Complexity II). *Assume F and G correspond to the choice CDFs of two different types and $\sup_x |F(x) - G(x)| \geq \xi$. Let F_n and G_m be independent empirical distribution based on m and n i.i.d. samples drawn from F and G, respectively. Denote $T_{min} = min\{m, n\}$. Define $p_{j|i} = 1 - s(i,j)$. $\forall \delta > 0, \epsilon > 0$, we can achieve $p_{j|i} < 1 - \xi + \epsilon$ with probability at least $1 - \delta$ with $T_{min} = \mathcal{O}(\frac{1}{\epsilon^2}\log\frac{1}{\delta})$.*

Proof. By definition,

$$\begin{aligned} \xi \leq \quad & \sup_x |F - G| \\ \leq \quad & \sup_x |F - F_n + F_n - G_m + G_m - G| \\ \leq \quad & \sup_x |F - F_n| + \sup_x |F_n - G_m| + \sup_x |G_m - G| \end{aligned} \tag{3.5}$$

Therefore,

$$\begin{aligned}
& \mathbb{P}(\sup_x |F_n - G_m| \leq \epsilon) & (3.6) \\
\leq \; & \mathbb{P}(\sup_x |F - F_n| + \sup_x |G_m - G| > \xi - \epsilon) \\
\leq \; & \mathbb{P}(\sup_x |F - F_n| > \frac{\xi - \epsilon}{2} \cup \sup_x |G - G_m| > \frac{\xi - \epsilon}{2}) \\
= \; & 1 - \mathbb{P}(\sup_x |F - F_n| \leq \frac{\xi - \epsilon}{2})\mathbb{P}(\sup_x |G - G_m| \leq \frac{\xi - \epsilon}{2}) \\
\leq \; & 4 \exp\left(-2T_{min}\left(\frac{\xi - \epsilon}{2}\right)^2\right)
\end{aligned}$$

where the first inequality is based on (3.5) and the second inequality makes use of the fact that $\sup_x |F_n - F| + \sup_x |G_m - F| > \xi - \epsilon$ implies that either $\sup_x |F_n - F| > \frac{\xi-\epsilon}{2}$ or $\sup_x |G_m - F| > \frac{\xi-\epsilon}{2}$. The equality comes from the independent assumption between F_n and G_m. The last inequality is from the DWK inequality.

We want to restrain the sampling probability such that $p_{j|i} = 1 - s$ is within ϵ-radius of the smallest possible sampling probability, which is $1 - \xi$, i.e.

$$\begin{aligned}
\mathbb{P}(1 - s < 1 - \xi + \epsilon) &= \mathbb{P}(s > \xi - \epsilon) \\
&= 1 - \mathbb{P}(s \leq \xi - \epsilon) \\
&\geq 1 - 4 \exp\left(-2T_{min}\left(\frac{\epsilon}{2}\right)^2\right)
\end{aligned}$$

as a result of Equation 3.6.

Let $\delta = 4\exp(-\frac{1}{2}T_{min}\epsilon^2)$. Then we have $T_{min} = \mathcal{O}(\frac{1}{\epsilon^2}\log\frac{1}{\delta})$. □

Theorem 3.3 complements Theorem 3.2 to rule out the out-types from being selected with a high probability. The more separable (larger ξ) their underlying distributions are, the lower accepting probabilities are.

Lemma 3.4. *For $\epsilon > 0$ and $0 < \delta < 1$, with minimum of choice records T required by Theorem 3.2 and Theorem 3.3, $\forall$ $\boldsymbol{q}_\ell \in \mathcal{Q}$, $\left\|\boldsymbol{q}_\ell - \boldsymbol{q}_{\pi(\ell)}\right\| \leq \epsilon$ with probability at least $1 - \delta$, where $\hat{\boldsymbol{q}}_\ell = \frac{1}{nT}\sum_{i \in I_\ell}\sum_{t=1}^{T} Y_i^{(t)}$.*

Proof. Based on Theorem 2, for in-type decision makers, we can achieve $\mathbb{P}(\sup_x |F_T(x) - F(x)| \leq \epsilon) > 1 - \delta$ with enough samples, where $F(x)$ is the CDF of $\boldsymbol{q}_k$ and $F_T(x)$ is the empirical CDF. In addition, the sampling probability for in-types is at least $1-\epsilon$. Subsequently, we can show, $\forall x$,

$$\mathbb{P}\left(\left|\frac{1}{T}\sum_{t=1}^{T} Y_{ix}^{(t)} - q_{kx}\right| \leq \epsilon\right) > 1 - \delta$$

if i is in-type.

For out-type decision makers, Theorem 3 states, if i' is of mixture k' and with enough samples, we have

$$\mathbb{P}\left(-\epsilon + \xi < \left|\frac{1}{T}\sum_{t=1}^{T} Y_{i'x}^{(t)} - q_{kx}\right| < \epsilon + \xi\right) \geq 1 - \delta, \forall x,$$

where $\sup_x |F(x) - G(x)| = \xi \geq \epsilon$ (Definition 1) and $F(x)$ and $G(x)$ are CDFs for mixture k and k' respectively.

On the other hand, note that the sampling probability is upper bounded by $1-\xi+\epsilon$ for out-type. This implies that if ξ is large, then it is unlikely that an out-type will be added to the subsample while if ξ is small, $\left|\frac{1}{T}\sum_{t=1}^{T} Y_{i'x}^{(t)} - q_{kx}\right|$ becomes negligible. In particular, setting ϵ such that $\epsilon < \frac{\xi}{2}$ can make sure that out-type sampling probability is less than $1-\epsilon$ while for in-type, it is least $1-\epsilon$ with high probability. By doing so, we can distinguish between in-type and out-type and obtain I_ℓ such that it contains only in-type with high probability given enough samples. □

Lemma 3.4 states that with enough data samples, the $\mathcal{Q}$-construction algorithm achieves Property 4, as shown in Figure 3-4, which SSRFW's desired provable convergence builds upon.

Sample complexity vs computational complexity. As a final remark, we discuss the number of subsamples (i.e. L) needed in the $\mathcal{Q}$-construction process. We distinguish this from sample complexity as we are not requesting more data points with a larger L. Instead, we should view L as a computation complexity factor. Recall that each moving direction derived from the *support finding step* is viewed as a learned logit vector of one mixture component. Therefore, we want the $\mathcal{Q}$-construction algorithm to include at least one $\hat{\boldsymbol{q}}_\ell$ for each mixture k in order for it to be picked in SSRFW . This impose a requirement on L. Theorem 3.4 states that the expected number of the number of subsamples is controlled by the smallest mixture weight.

Theorem 3.4. *Assume $\alpha_1 \leq \alpha_2 \leq \cdots \alpha_K$. The expected number of subsamples L we need to construct is bounded by $\frac{1}{\alpha_1} \log \frac{1}{\alpha_1}$.*

Intuitively, this means if we have one mixture component that is very underrepresented, then we need to create more subsamples to ensure that one of the seeds belongs to this mixture.

Proof. As in Algorithm 3, we use L as the number of subsamples we need to construct. Let L_k be the number of subsamples needed to hit the k-th mixture type after $k-1$ types of seeds have been selected. We have $L = L_1 + \cdots + L_K$.

We first construct a simple and fake scenario where we have K' mixture types with each mixture weight equal to α_1, i.e. $K' = \frac{1}{\alpha_1}$. Similarly, we can define L' and L'_k as above and also have $L' = L'_1 + \cdots + L'_{K'}$. Think of L' and $L'_k, k = 1, \ldots, K'$ as random variables and we know the probability of selecting a seed from a new type k is $p_k = \frac{K'-k+1}{K'}$ since in the fake scenario, each type has the same probability α_1 of being chosen. This tells us that L'_k has a geometric distribution with expectation $\frac{1}{p_i} = \frac{K'}{K'-k+1}$.

By the linearity of expectations we have

$$\begin{aligned}
\mathbb{E}[L'] &= \mathbb{E}[L'_1 + L'_2 + \cdots + L'_{K'}] \\
&= \mathbb{E}[L'_1] + \mathbb{E}[L'_2] + \cdots + \mathbb{E}[L'_{K'}] \\
&= \frac{K'}{K'} + \frac{K'}{K'-1} + \cdots + \frac{K'}{1} \\
&= K' \cdot \left(\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{K'} \right) \\
&= K' \cdot H_{K'}
\end{aligned}$$

where H'_K is the K'-th harmonic number. Using the asymptotics of the harmonic numbers, we get

$$\mathbb{E}[L'] \approx K' \log(K') = \frac{1}{\alpha_1} \log(\frac{1}{\alpha_1})$$

Since $\alpha_1 \leq \cdots \leq \alpha_K$, we know $K' \geq K$. On the other hand, we know the $\mathbb{E}[L_k] \leq \mathbb{E}[L'_k]$ since there is a higher probability of choosing any mixture type $k \geq 2$, due to the same reason, i.e. $\alpha_k \geq \alpha_1$. Therefore, we have $\mathbb{E}[L] \leq \frac{1}{\alpha_1} \log(\frac{1}{\alpha_1})$

We can further characterize the probability of event $\mathcal{H}$, which describes the event that all mixture types are included in the constructed set $\mathcal{Q}$ by creating L subsamples.

□

Claim 3.2. *For any $\delta > 0$, we have $\mathbb{P}(\mathcal{H}_L) \geq 1 - \delta$ with L chosen according to the criteria described below.*

Proof. Denote Z_k^L as the event that k-th mixture type is not being chosen as seed in the L trials. Similarly, we can define $Z_k'^L$ for the fake scenario as described above. We then have

$$\mathbb{P}(Z_k^L) = (1 - \alpha_k)^L \leq \left(1 - \frac{1}{K'}\right)^L = \mathbb{P}(Z_k'^L) \leq e^{-\frac{L}{K'}}$$

Denote W_k as the event that the convex hull formed by the set $\{\hat{\boldsymbol{q}}_\ell = \frac{1}{nT} \sum_{i \in I_\ell} \sum_{t=1}^T Y_i^{(t)} | \pi(i) = k\}$ for mixture type k covers the true choice probability vector. Note that each $\frac{1}{T} \sum_{t=1}^T Y_i^{(t)}$ can be viewed as a sample mean of $\boldsymbol{q}_k$ and by central limit theorem, it is symmetric with respect to $\boldsymbol{q}_k$, hence so are the $\hat{\boldsymbol{q}}_\ell$'s. According to Lemma 3.2 and Corollary 3.1,

$$\begin{aligned} \mathbb{P}(W_k) &= 1 - \frac{1}{2^{L_k - 1}} \sum_{i=0}^{d-1} \binom{L_k - 1}{i} \\ &\geq 1 - \frac{d}{(d-1)!} \frac{(L-1)^d}{2^{L-1}} \qquad\qquad \text{when } L \geq 2d - 1 \end{aligned}$$

Putting everything together, we have

$$\mathbb{P}(\mathcal{H}) = \left(1 - \mathbb{P}(\cup_{k=1}^{K} Z_k^L)\right) \mathbb{P}(\cap_{k=1}^{K} W_k) \tag{3.7}$$

$$\geq \left(1 - \mathbb{P}(\cup_{k=1}^{K'} Z_k^L)\right) \left(1 - \frac{1}{2^{L-1}} \sum_{i=0}^{d-1} \binom{L-1}{i}\right)^K \tag{3.8}$$

$$\geq \left(1 - \frac{1}{\alpha_1} e^{-L\alpha_1}\right) \left(1 - \frac{d}{(d-1)!} \frac{(L-1)^d}{2^{L-1}}\right)^K \tag{3.9}$$

For any $\delta > 0$, we can then choose L such that $1 - \delta \leq$ RHS of Equation (11) and $L \geq 2d - 1$. □

We can now conclude that according to Theorem 3.2, Theorem 3.3, the sample complexity is $\mathcal{O}(\frac{1}{\epsilon^2} \log(\frac{1}{\delta}))$ for the `SSRFW` algorithm.

3.3.3 Proof of the Main Theorem

We first discuss Corollary 3.2 and what "high probability" refers to.

Denote W_k as the event that the convex hull formed by the subsamples for a mixture type k covers the true choice probability vector. If we have subsampled all mixture types and for each type k, event W_k occurs, we can obtain $\boldsymbol{q}_k \in \mathrm{Conv}(\mathcal{Q})$. Subsequently, we have $\mathrm{Conv}(\{\boldsymbol{q}_k\}_{1,\dots,K}) \subseteq \mathrm{Conv}(\mathcal{Q})$.

On the other hand, we have already analyzed the probability for event that *sub-sampled all mixture types and* Y_k *occurs* $\forall k$ to occur, which is precisely $\mathcal{H}_L$ as defined above. Specifically, it happens with probability $\geq \left(1 - \frac{1}{\alpha_1} e^{-L\alpha_1}\right) \left(1 - \frac{d}{(d-1)!} \frac{(L-1)^d}{2^{L-1}}\right)^K$. As L increases, this number quickly increases to 1. This completes the proof of Corollary 3.2. Finally, we combine all the results above and prove the provable convergence part in Theorem 3.1.

Proof. As illustrated in Figure 3-4, we want to show with probability $\geq 1 - \delta$ we have

S.1 $|\mathbf{g}^{\texttt{SSRFW}} - \sum_k \alpha_k \boldsymbol{q}_k| \leq \epsilon$

S.2 $\forall\, k', \exists\, k = \pi(k')$ s.t. $|\hat{\boldsymbol{q}}_{k'} - \boldsymbol{q}_k| \leq \epsilon$

S.3 $|\sum_{k':\pi(k')=k} \alpha_{k'} - \alpha_k| \leq \epsilon$

First, **S.1** is proved by Lemma 3.1 (Property 1: $|\mathbf{g}^{\texttt{FW}} - \sum_k \alpha_k \boldsymbol{q}_k| \leq \epsilon$) and Corollary 3.2 (Property 3: $\sum_k \alpha_k \boldsymbol{q}_k \in \text{Conv}(\mathcal{Q})$), together with the fact that $\mathbb{P}\left(|\frac{1}{T}\sum_{t=1}^T \mathbf{y}^t - \sum_k \alpha_k \boldsymbol{q}_k| \leq \epsilon\right) > 1 - \delta$ by central limit theorem. Since $\sqrt{T}(\frac{1}{T}\sum_{t=1}^T \mathbf{y}^t - \sum_k \alpha_k \boldsymbol{q}_k) \xrightarrow{d} \mathcal{N}(0, \sigma^2)$, number of samples required is also in the order of $\frac{1}{\epsilon^2}\log(\frac{1}{\delta})$.

Second, **S.2** is also a combined result by Frank-Wolfe's solving a linear program as an intermediate step (Property 2: $\forall\ k'$, $\hat{\boldsymbol{q}}_{k'} \in \mathcal{E}(\text{Conv}(\overline{\mathcal{P}}))$, $\mathcal{E}(\cdot)$ denoting the extreme point set of the input region) and by construction using Algorithm 4 (Property 4: $\forall\ \boldsymbol{q} \in \mathcal{E}(\text{Conv}(\mathcal{Q})), \exists\ k$ s.t. $\|\boldsymbol{q} - \boldsymbol{q}_k\| \leq \epsilon$).

Subsequently, we can show **S.3**. Denote K' as the number of mixtures output by the `SSRFW` algorithm. Using **S.1**, we first write

$$|\sum_{k'=1}^{K'} \hat{\alpha}_{k'} \hat{\boldsymbol{q}}_{k'} - \sum_{k=1}^{K} \alpha_k \boldsymbol{q}_k| \leq \epsilon' \tag{3.10}$$

According to **S.2**, $\exists\ \pi$ such that $\pi(k') = k$ and we can write $\hat{\boldsymbol{q}}_{k'} = \boldsymbol{q}_{\pi(k')} + \boldsymbol{\epsilon}'$ where $|\boldsymbol{\epsilon}'| \leq \epsilon'$. Rearranging Eq. (3.10) gives

$$|\sum_{k=1}^{K} \boldsymbol{q}_k \left(\sum_{k':\pi(k')=k} \hat{\alpha}_{k'} - \alpha_k\right) + \sum_{k'=1}^{K'} \hat{\alpha}_{k'} \boldsymbol{\epsilon}'| \leq \epsilon' \tag{3.11}$$

By triangle inequality, we get

$$|\sum_{k=1}^{K} \boldsymbol{q}_k \left(\sum_{k':\pi(k')=k} \hat{\alpha}_{k'} - \alpha_k\right)| - \epsilon' \leq \epsilon'$$

Since $\boldsymbol{q}_k$ is some arbitrary non-zero vector, we must have $\left(\sum_{k':\pi(k')=k} \hat{\alpha}_{k'} - \alpha_k\right) \leq 2\epsilon'$, $\forall\ k$, which completes **S.3** by letting $\epsilon = 2\epsilon'$.

Note the above result holds assuming $K' \geq K$. To see why this is always the case, consider the linear system $Q\boldsymbol{x} = [\boldsymbol{q}_1 \boldsymbol{q}_2 \ldots \boldsymbol{q}_K][x_1, x_2, \ldots, x_K]^\top = \mathbf{g}$, where $Q \in \mathbb{R}^{M \times K}$ and $\mathbf{g} = \sum_{k=1}^K \alpha_k \boldsymbol{q}_k$. According to Definition 1, $D_{KS}(\boldsymbol{q}_k, \boldsymbol{q}_{k'}) \geq \epsilon$, we know that all $\boldsymbol{q}_k$'s are linearly independent. Since $M \geq K$, $\text{rank}(Q) = \text{rank}(Q|\mathbf{g}) = K$. The linear system has a unique solution that $\boldsymbol{x} = \boldsymbol{\alpha}$, where all x_k's are non-zero. On the other hand, we have $\mathbf{g}^{\texttt{SSRFW}} = \sum_{k'=1}^{K'} \hat{\alpha_{k'}} \hat{\boldsymbol{q}}_{k'}$ and $\|\mathbf{g}^{\texttt{SSRFW}} - \mathbf{g}\| \leq \epsilon$. Assume $K' < K$, then upto a difference of ϵ, the linear system $\hat{Q}\boldsymbol{x} = \mathbf{g}$, where $\hat{Q} = \begin{bmatrix} \\ \hat{\boldsymbol{q}}_1 \hat{\boldsymbol{q}}_2 \dots \hat{\boldsymbol{q}}_{K'} \end{bmatrix} \in \mathbb{R}^{M \times K'}$, is inconsistent. In other words, we will not be able to obtain a $\mathbf{g}^{\texttt{SSRFW}}$ that is ϵ-close to $\mathbf{g}$, making it impossible to reach the stopping condition in the `SSRFW` algorithm. Therefore, the algorithm will keep going for more iterations, until we have at least $K' = K$.

Finally, according to Theorem 3.2, Theorem 3.3, the sample complexity is $\mathcal{O}(\frac{1}{\epsilon^2} \log(\frac{1}{\delta}))$.

□

Additional discussion. In the proof, we showed that number of mixtures returned by `SSRFW` , K', is at least the ground truth number of mixtures, K. A natural question to ask is that how the misaligned number of mixtures affect the learning result, if $K' \neq K$. In many situations, this would not be a problem.

Consider the case that $\exists k_1, k_2$, such that $\pi(k_1) = \pi(k_2) = k$ while the rest are all one-to-one mapping. According to Theorem 3.1, we have $\mathbb{P}(|\boldsymbol{q}_{k_i} - \boldsymbol{q}_k| < \epsilon) \geq 1 - \delta$, for $i = 1, 2$ and $\mathbb{P}(|\alpha_{k_1} + \alpha_{k_2} - \alpha_k| < \epsilon) \geq 1 - \delta$. We can view the ground-truth MMNL model as an $(M+1)$-MNL model, where the original k-th mixture is now divided into two MNL components which share the identical logit parameters, where one of them has mixture weight α_{k_1} and the other one $\alpha_k - \alpha_{k_1}$. It is not hard to see that for the first component, we learned the correct mixture weight with an ϵ-close logit vector $\hat{\boldsymbol{q}}_{k_1}$ while for the second, the mixture weight is off by at most ϵ with an ϵ-close logit vector $\hat{\boldsymbol{q}}_{k_2}$.

Finally, we give some comment on the mapping function π. Note that we do not need this information other than using it as a tool in the proofs, though we can design heuristics to learn the mapping. In real world applications, we do not know the ground truth parameters, so we cannot derive such mapping anyways. On the other hand, if we run the algorithm multiple times, we will get different results due to the stochasticity embedded in the algorithm. In general, the lexicographic order of the mixtures is not important and can be reordered arbitrarily. This is referred to as "label swapping" for mixture model learning problems but in general it can be safely ignored.

To summarize, SSRFW mainly benefits from making better utilization of the personal level choice data. Such data are called panel data in statistics and econometrics and commonly used for longitudinal studies. In the next chapter, we will demonstrate how to apply our algorithm in both simulation studies and real world panel datasets.

Chapter 4

Numerical Experiments and Case Studies

In this chapter, we will demonstrate the advantage of SSRFW over the original FW algorithm in several settings. First, we will conduct simulation studies, which allows us to compared the learning outcome to the ground truth values. Second, we will apply both algorithms to the Nielsen Consumer Panel Data and report different statistics for performance comparison.

4.1 Simulation Studies

4.1.1 Data Generation

We generate the data using the following set of hyperparameters:

- Choice set size: $M = 10$ with an offset option 0
- Population size: $N = 2000$
- Number of mixtures: $K = 5$, denoted as "A", "B", "C", "D", "E"
- Feature vector dim: $d = 10$
- Time periods: $T \in [5, 300]$ with increment of 5

- Number of seeds: $L = 75$ (cardinality of $\mathcal{Q}$)

Note that we also include the offset option to allow the possibility of choosing nothing from the option set. This is a more realistic case in many real world applications.

Attributes of different options and preference vector β_k's are randomly generated in the interval $[-1, 1]$. α_k's are randomly generated such that $\sum_k \alpha_k = 1$ and the minimum mixture proportion $\min_k \alpha_k \geq \frac{1}{K+3}$ to ensure that not a particular type is under-represented. We set $L = 75$.

Sample Purity

After running Algorithm 4, we obtain a candidate set $\mathcal{Q}$ of choice probability vectors $\boldsymbol{q}$, corresponding to each subsample. This will then be fed into the `SSRFW` algorithm. Since the `SSRFW` algorithm chooses moving direction from this candidate set, a high quality $\mathcal{Q}$ will directly affect the learning outcome of `SSRFW` . Therefore we report the subsample purity as an intermediate performance measure. The intuition lies in that if a subsample has higher purity, i.e., more decision makers in the set are of the same type, the higher the chance that it is closer to the true value associated with the majority type within this sample.

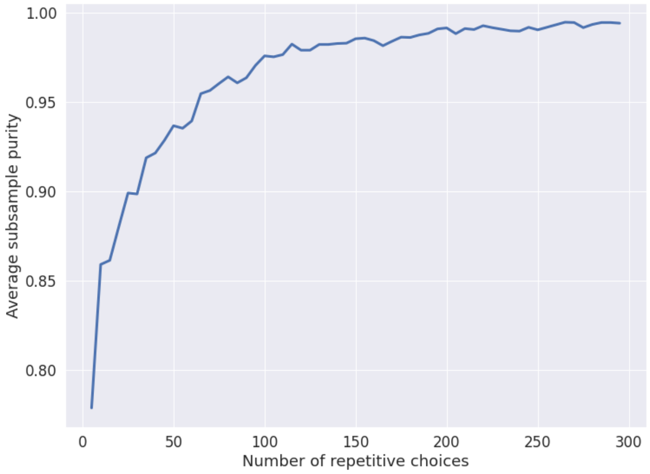


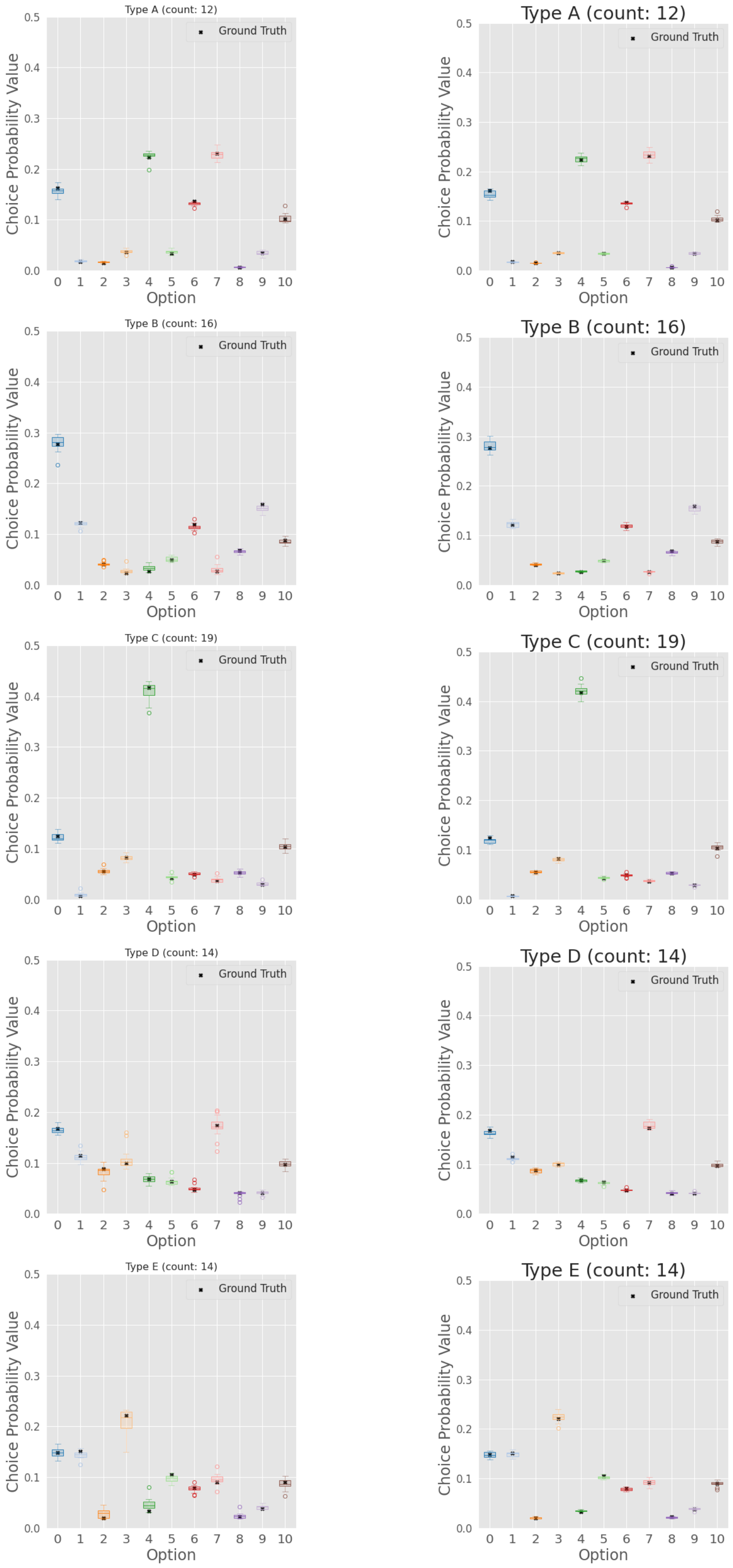
Figure 4-1: Average subsample purity

Figure 4-1 shows the average subsample purity with respect to the number of repetitive choices a decision maker has made. We can see that the constructed subsamples can achieve 90% purity (i.e. 90% of the consumers in the subsample are of the same type) with as few as 30 experiment epochs, and quickly reaches 99% around $T = 150$.

Quality of Set $\mathcal{Q}$

Next we evaluate the quality of $\mathcal{Q}$. According to SSRFW, the estimated choice probability outcome is essentially a subset of these candidate vectors in $\mathcal{Q}$—the ones emitted by SSRFW at each iteration. Therefore, the higher the quality of $\mathcal{Q}$—in the sense it is concentrated near the ground truth—the better mixture estimation we can obtain using the learning algorithm SSRFW. We categorize each subsample to each mixture type in the ground truth with the closest choice probability. For instance, we find that 19 out of 75 subsamples are categorized as generated from Type A. We plot the distribution of choice probability values for all mixture types in Figure 4-2, with Figure 4-2a showing the result with 50 repetitive choices and Figure 4-2b with 300 repetitive choice.

These figures show that the estimated choice probability values are very concentrated near the ground truth when $T = 300$ and are also reasonably good even with a small $T = 50$. This suggests that when use our algorithm in real world applications, the number of choices that the algorithm requires for each decision makers is within a reasonable range, depending on the application. For instance, as we will see later in the Nielsen consumer panel data, the average number of purchases (choices) for grocery items is close to the hundred.



(a) $T = 50$

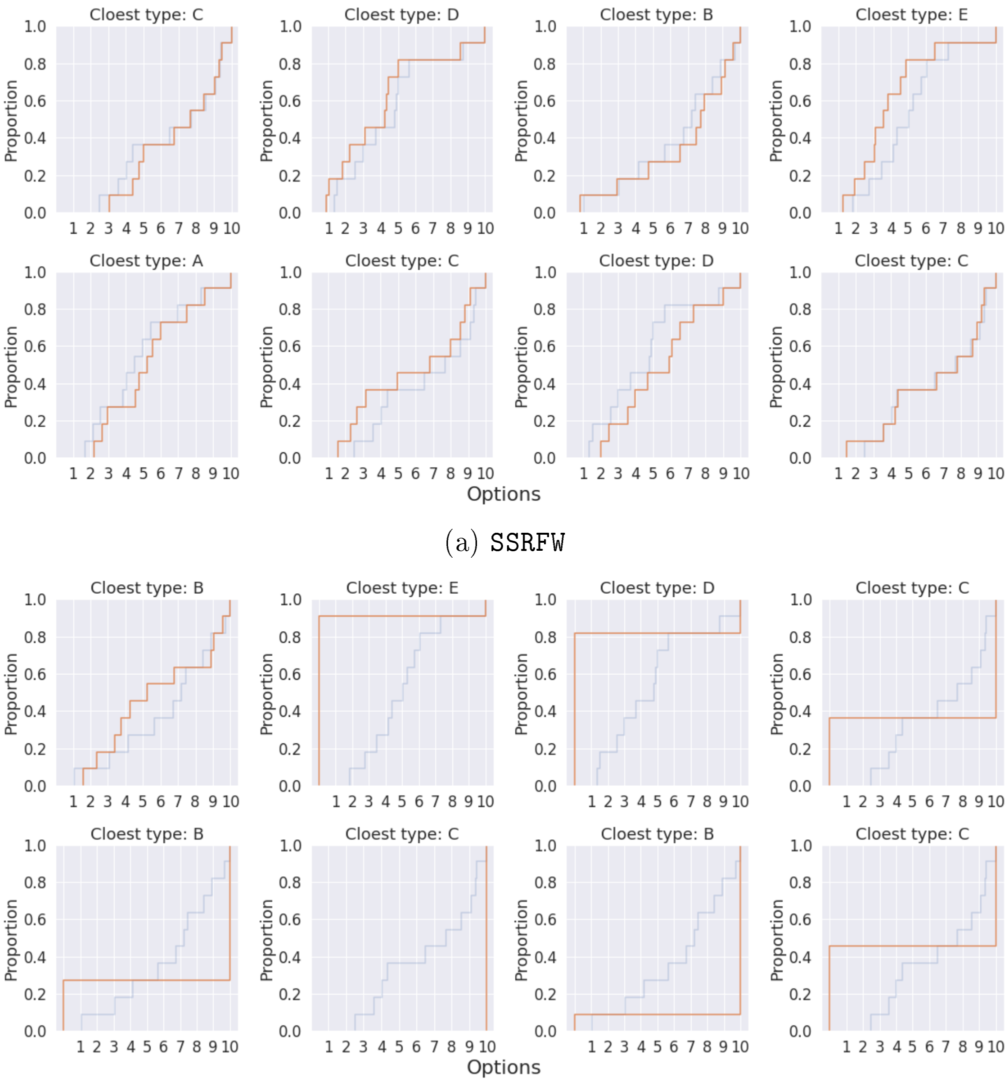
(b) $T = 300$

Figure 4-2: Choice Probability Estimation Result for All Consumer Types

4.1.2 Mixture Type Recovery

Since we have obtain good estimates with $T = 50$, we will keep this in the rest of the experiments.

After feeding the $\mathcal{Q}$ above to Algorithm SSRFW, it generated 8 mixture types (as expected, larger than the ground truth number of mixtures) and we would like to compare these generated segments against the ground truth. We refer to the mapping $\pi(j) = i$ as the "closest type" if the majority in the subsample which generated this $\hat{\boldsymbol{q}}$ is of that type.



(a) SSRFW

(b) Original FW

Figure 4-3: Comparison of empirical cumulative distribution of logit vectors

Figure 4-3 compares j and $\pi(j)$ using the cumulative distribution of the choice probability vectors. The orange lines are the learning outcome from the two algorithms while the light blue line in the background depicts the ground truth for the closest type. We can see that the results from `SSRFW` are very close to the true mixture CDFs (Figure 4-3a). In contrast, the original `FW` algorithm is incapable of recovering the true choice probabilities (Figure 4-3b). We further verify the validity of this outcome by checking the mixture proportion estimates α's. Table 4.1 shows that if we have a one-to-one mapping (such as A and E), the α estimates from the algorithm is close to the true values. If we have many-to-one mappings, i.e., the algorithm outputs multiple mixtures to the same latent class, the sum of the estimated mixture proportions is also close to the true values of each mixture.

Table 4.1: Mixture Proportion α Estimation

Type	Ground Truth α	`SSRFW` Generated $\hat{\alpha}$	Type-wise summation of $\hat{\alpha}$
A	0.2000	$\hat{\alpha}_5$: 0.1904	0.1904
B	0.2364	$\hat{\alpha}_1$: 0.0713 $\hat{\alpha}_7$: 0.1607	0.2320
C	0.1636	$\hat{\alpha}_3$: 0.0741 $\hat{\alpha}_8$: 0.0757	0.1498
D	0.2182	$\hat{\alpha}_4$: 0.0387 $\hat{\alpha}_6$: 0.2143	0.2530
E	0.1818	$\hat{\alpha}_2$: 0.1748	0.1748

4.1.3 Comparison with the Original FW Algorithm

We define another performance measure as the weighted average of distance from algorithm generated choice probabilities to its closest ground truth choice probabilities, with weights equal to the corresponding mixture proportion:

$$\sum_k \frac{\alpha_k}{\#(k')} \sum_{k':\pi(k')=k} \|\hat{\boldsymbol{q}}_{k'} - \boldsymbol{q}_k\|$$

which looks at the aggregated discrepancy of the learned logit vectors from ground truth. Figure 4-4 plots this quantity with respect to the number of repetitive choices, from which we can see that the total discrepancy is much smaller using the `SSRFW` algorithm regardless the number of choice repetitions. This is exactly due to the fact that the original Frank-Wolfe approach only aims to minimize the distance between the aggregated choice probability of the entire population and its estimators for each individual mixture can be very different from the ground truth. It also suggests that even we have a small amount of the repetitive choice data, we could still benefit from using that info as well as the `SSRFW` algorithm.

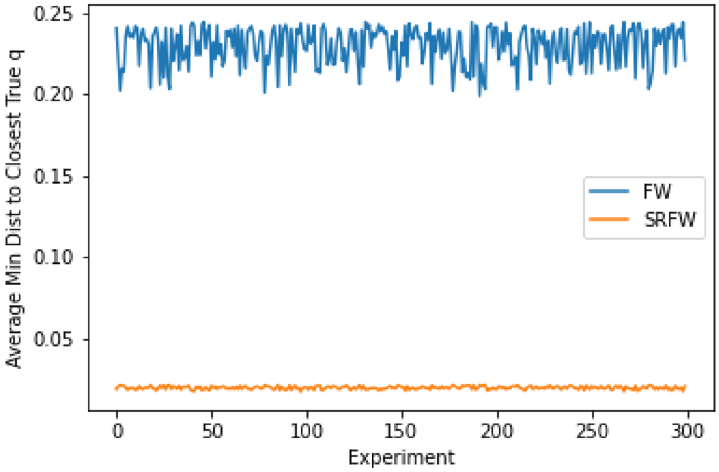


Figure 4-4: Comparison with the original `FW`

4.1.4 Comparison with the EM Algorithm

Finally, we compare our algorithm to the most widely applied algorithm to estimate mixed MNL choice models, the EM algorithm. EM algorithm works well when the true number of consumer types is known, however, in general we do not have this information.

We first compare the performance of SSRFW to EM with various different hyperparameter $\tilde{K}$ values, as shown in Figure 4-5.

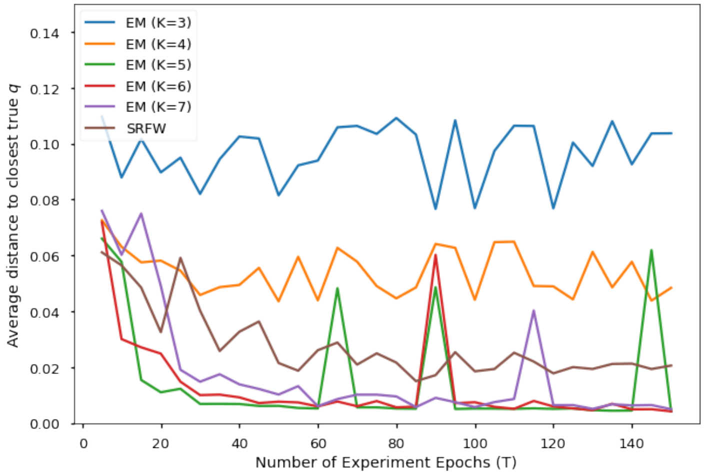


Figure 4-5: Comparison with EM

We can see from the plot that when $\tilde{K} < K$, SSRFW usually outperforms the EM algorithm. On the other hand, when $\tilde{K} >= K$, while EM can achieve a marginally better performance, it suffers from its instability. Then, we adopted the conventional strategy (c.f. Train [2008]) to use AIC/BIC to determine the best $\tilde{K}$, which corresponds to the lowest of these two criteria respectively. Both AIC/BIC measure the relative quality of statistical models for a given dataset, by balancing the trade-off between goodness of fit and the simplicity of the model and are commonly used in model selection. In our experiment, $K = 3$ gives the lowest value for both AIC and BIC. This indicates we should choose $K = 3$, yet it will result in worse estimation quality as shown in Figure 4-5.

4.2 Case Study: The Nielsen Consumer Panel Data

In this section, we demonstrate how we have applied `SSRFW` to the Nielsen Consumer Panel data. This comprehensive dataset is provided by the Kilts Center for Marketing at the University of Chicago Booth School of Business, NielsenIQ, and Nielsen. It contains panelists (i.e., households) purchase decisions on grocery items included in the NielsenIQ food and nonfood departments (roughly 1.4M UPC codes) dated back to 2004 with regular annual updates The panel size varies from 40K to 60K and the characteristics include product description, brand, multipack, size, etc. This panel data is widely used for longitudinal studies in marketing science.

4.2.1 Data Curation

We consider applying the algorithm to a substitute set of products under a particular category. This is a realistic setting as consumers usually choose one item from the substitution set. We curated data for six different categories, including yogurt, cereal, snack, candy, soft drinks and pet food and provide some summary statistics in Table 4.2.

Table 4.2: Nielsen case study: categories and data information

Category	Panel size	Number of features	Average # purchases
yogurt	1443	9	178
pet-food	2451	8	403
candy	1499	14	127
cereal	1085	13	96
snack	665	16	61
soft-drinks	412	12	209

4.2.2 Experiment Setup

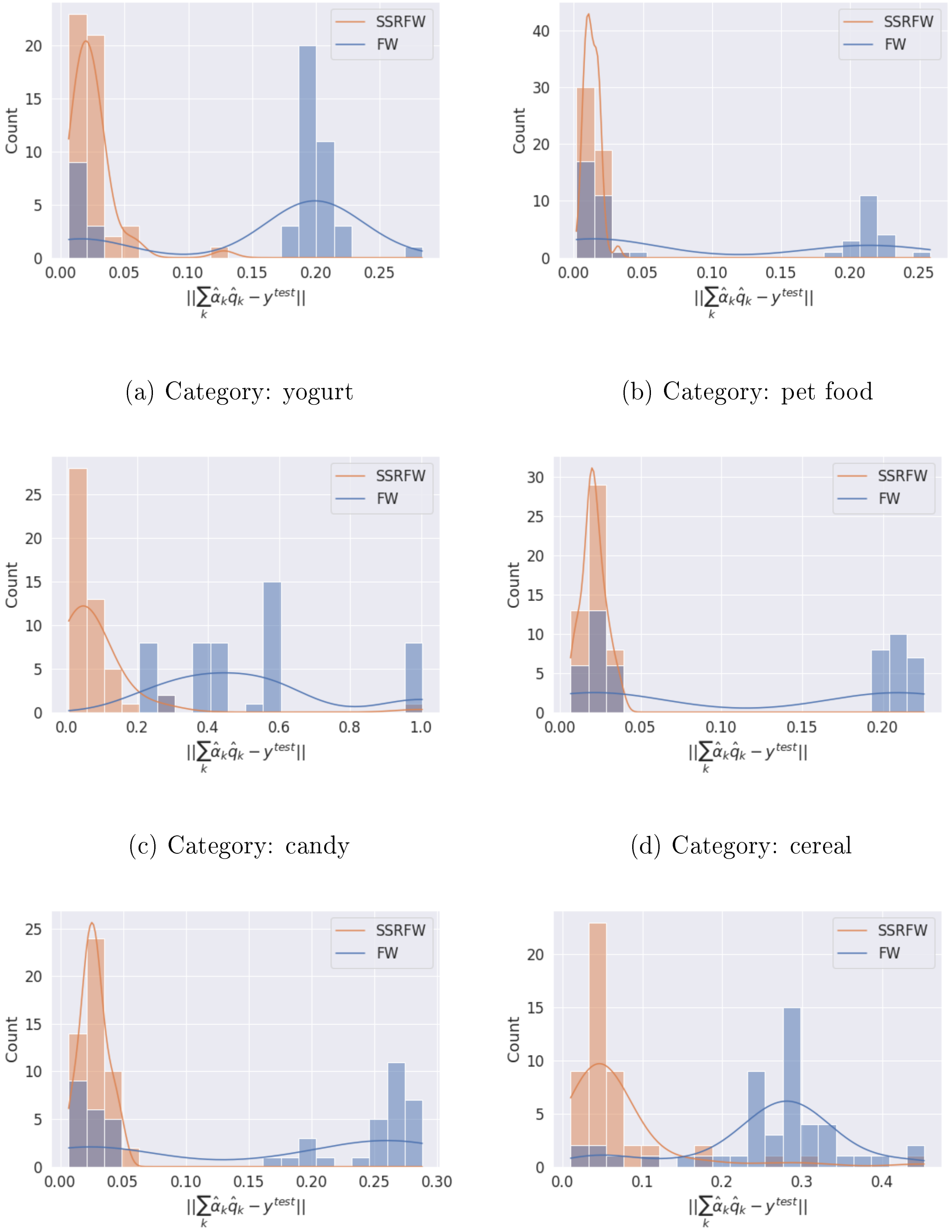
We cannot evaluate the model performance the same as in simulation studies since we no longer have the ground truth knowledge. Instead, we split the data into a training and test set, with the former used for learning the model parameters and the latter for evaluation. Specifically, we apply our algorithm to the training set, and use the learned parameters to compute the theoretical aggregated market share $\sum_k \hat{\alpha}_k \hat{\boldsymbol{q}}_k$. Then we compute its distance to the aggregated market share of the test set, i.e. $\|\sum_k \hat{\alpha}_k \hat{\boldsymbol{q}}_k - y^{\text{test}}\|$. The assumption is, if we have similar mixture composition in the training and test set, then the estimated parameter values from the training set should yield aggregated choice probability values close to that of the test set. To avoid randomness in the data split, we used a 10-fold cross validation, with the entire process repeated for five times.

The stopping criteria was set to $\|\boldsymbol{g} - \sum_k \hat{\alpha}_k \hat{\boldsymbol{q}}_k\| \leq 1e - 3$.

4.2.3 Results

Figure 4-6 plots the distribution of $\|\sum\limits_k \hat{\alpha}_k \hat{\boldsymbol{q}}_k - y^{\text{test}}\|$ from the repeated runs of both algorithms. We can see that `SSRFW` in general outperforms the original FW algorithm in that the discrepancy is close 0.

Figure 4-7 plots the deviation of product-level choice probability values from the test set, $\frac{|\sum_k \hat{\alpha}_k \hat{q}_{kj} - y_j^{\text{test}}|}{y_j^{\text{test}}}$, for $j = 1, 2, 3, 4, 5$. The light orange horizontal line indicates a zero deviation and we observe that the predicted aggregated choice probability per product level from `SSRFW` is more concentrated around zero than the original FW. In addition, there is a smaller variance with respect to different runs.



(a) Category: yogurt

(b) Category: pet food

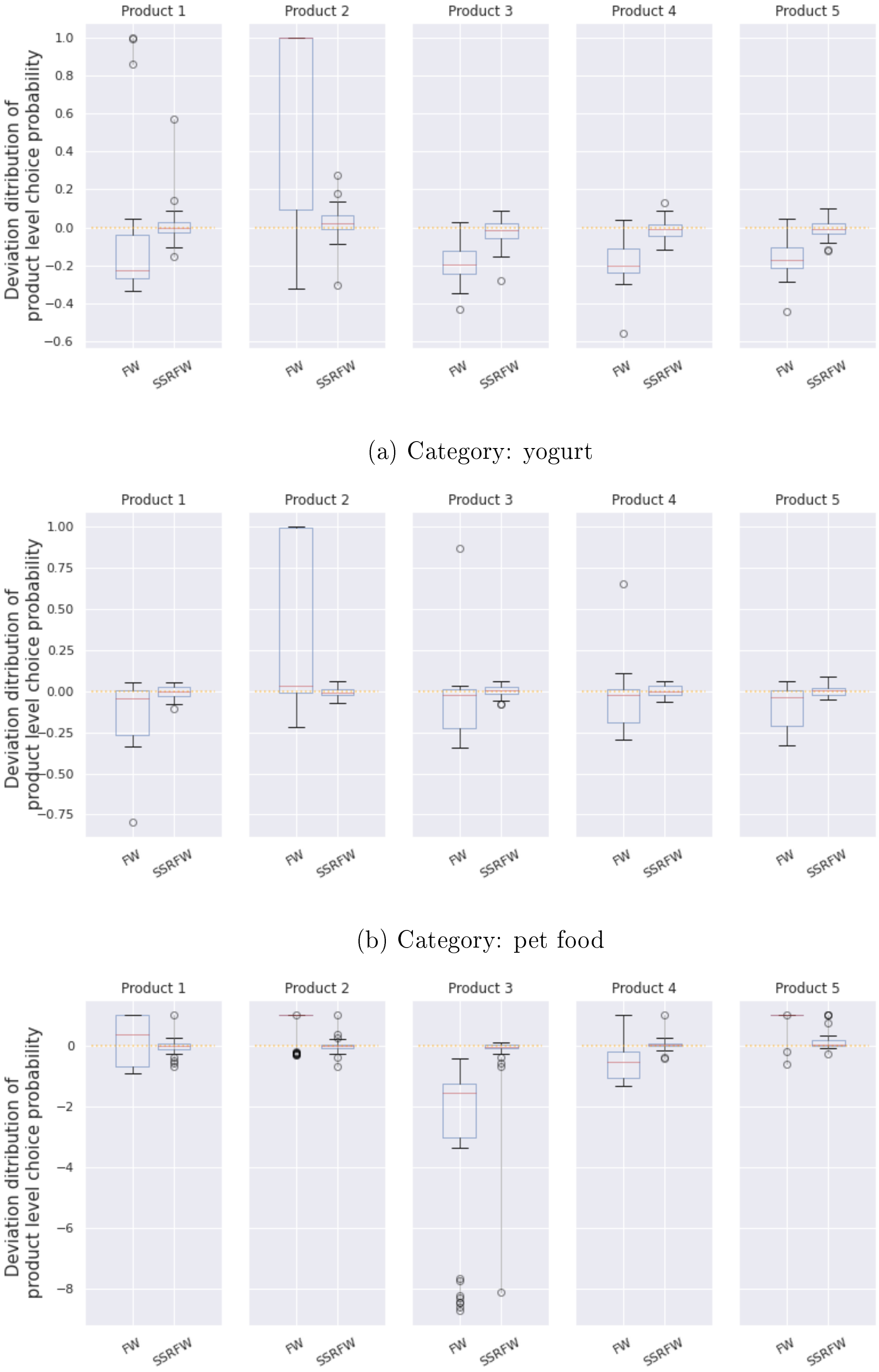
(c) Category: candy

(d) Category: cereal

(e) Category: snack

(f) Category: soft drinks

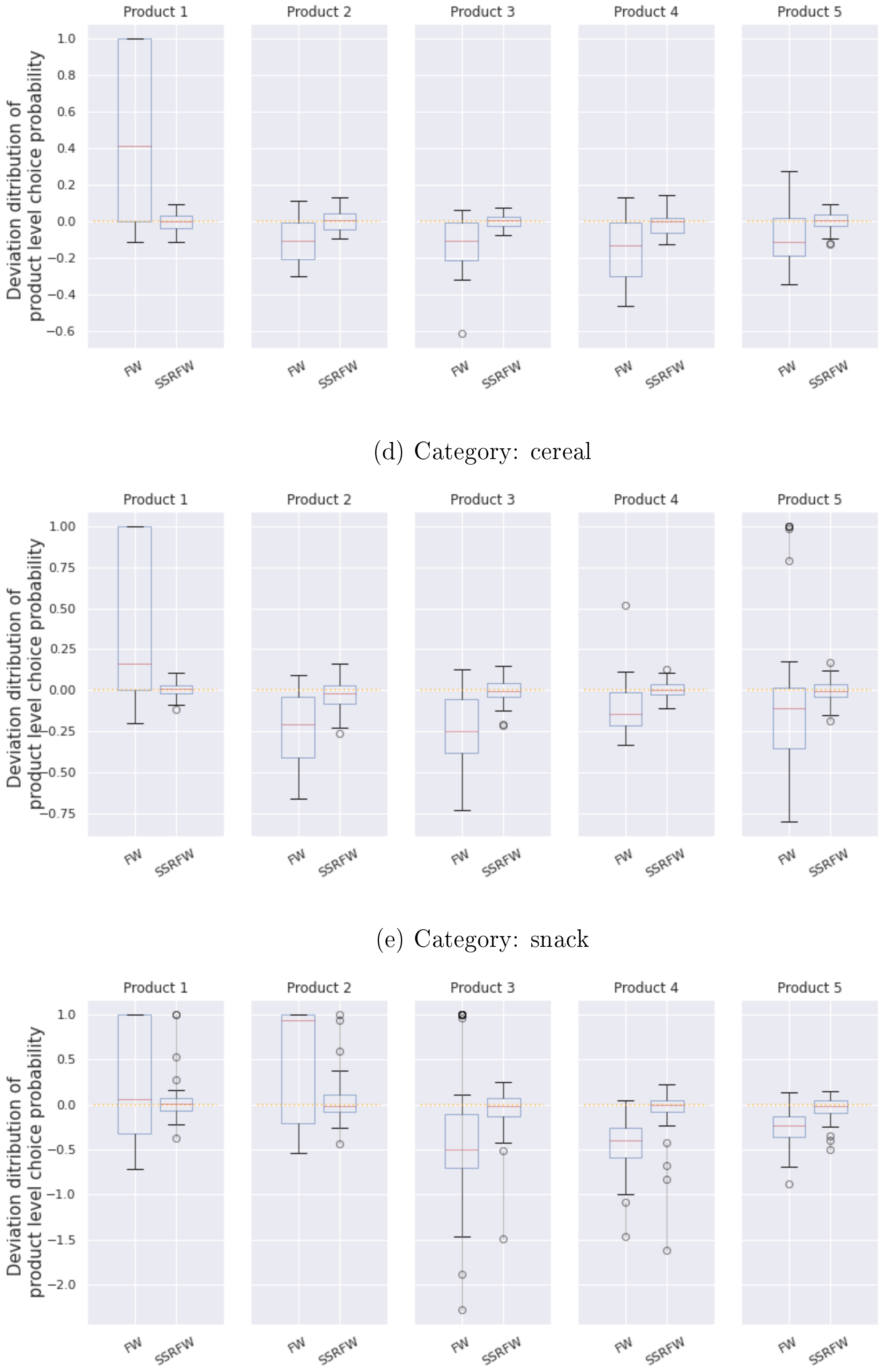
Figure 4-6: $\|\sum_k \hat{\alpha}_k \hat{\boldsymbol{q}}_k - y^{\text{test}}\|$ for the six product categories



(a) Category: yogurt

(b) Category: pet food

(c) Category: candy



(d) Category: cereal

(e) Category: snack

(f) Category: soft drinks

Figure 4-7: Deviation-from-test distribution of product-level choice probability values

Next, we examine the number of iterations required until convergence. We also report the percentage of active directions for the given number of iterations. We think these two metrics measures the effectiveness of each direction being chosen during the learning process.

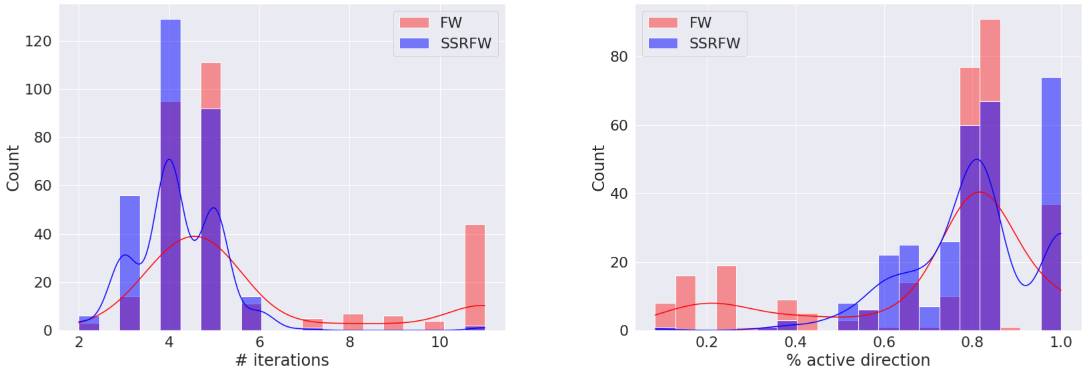
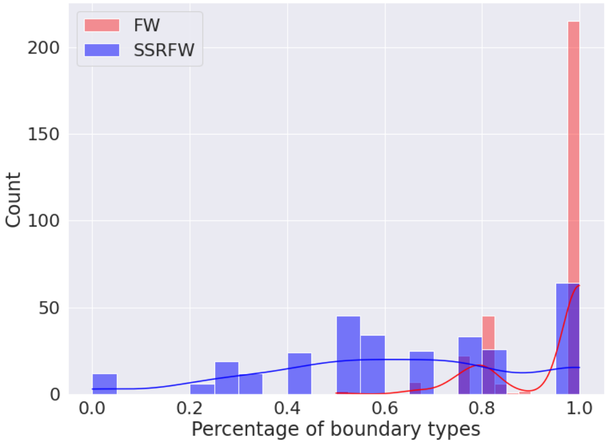


Table 4.3: Other Statistics

	# iteration		%active direction	
	FW	SSRFW	FW	SSRFW
yogurt	4.78	4.24	0.877	0.74
	(0.18)	(0.12)	(0.017)	(0.020)
pet food	4.60	4.00	0.817	0.863
	(0.10)	(0.09)	(0.002)	(0.015)
candy	10.42	3.74	0.239	0.675
	(0.21)	(0.19)	(0.020)	(0.024)
cereal	4.48	4.06	0.848	0.871
	(0.07)	(0.12)	(0.012)	(0.015)
snack	4.38	4.46	0.642	0.808
	(0.09)	(0.10)	(0.026)	(0.021)
soft drinks	5.70	4.88	0.835	0.866
	(0.33)	(0.17)	(0.010)	(0.014)

The last metric we look at is the percentage of *boundary types* in the learning result. Figure 4-9 shows that the original FW still exhibits the same problem of generating boundary-type logit vectors while `SSRFW` is much less likely to suffer from this problem.

Figure 4-9: Percentage of boundary types



Finally, as requested by Kilts Center for Marketing at the University of Chicago School of Business, we make the following disclaimers:

- Researcher(s)' own analyses calculated (or derived) based in part on data from Nielsen Consumer LLC and marketing databases provided through the NielsenIQ Datasets at the Kilts Center for Marketing Data Center at The University of Chicago Booth School of Business.

- The conclusions drawn from the NielsenIQ data are those of the researcher(s) and do not reflect the views of NielsenIQ. NielsenIQ is not responsible for, had no role in, and was not involved in analyzing and preparing the results reported herein.

Chapter 5

Conclusion

5.1 Summary

In this thesis, we focus on the topic of learning mixed multinomial logit models with theoretical guarantees, which is the process of estimating the model parameters using data and making sure that the estimators possess desirable statistical properties.

Chapter 2 studies the polynomial learnability of MMNL models with K number of mixtures, extending the current literature in 2-MNL setting. In particular, we show that there exists an algorithm that can learn a more general set of K-MNL models, if identifiable, using polynomial number of data points and polynomial number of operations under some mild assumptions. On the other hand, there is still many more directions one can explore under this regime of work. For instance, what is the identification condition for the general K-MNL models? Essentially, this aims to explore the relationship between m_0 and K. Another good research question is once m_0 can be precisely defined, what will be an efficient algorithm to learn a K-MNL model over the m_0-item universe without query the entire power set.

In Chapter 3, we propose the `SSRFW` algorithm, which provides an end-to-end solution for learning MMNL models using historical choice data. This novel approach utilizes a carefully designed sampling method to construct a meaningful search space. Not only does it resolves the drawback of the original Frank-Wolfe approach with boundary-type issues but also enables us to obtain provable guarantees for the model estimates and sample complexity. It is also more robust than traditional unsupervised approaches such as clustering method and EM methods.

We then conduct numerical experiments in Chapter 4 to evaluate the performance and demonstrate the advantage of the `SSRFW` algorithm in various settings. Simulation studies show that `SSRFW` is capable of recovering the ground truth parameter values while the original FW fails to do so. We then present how to apply our algorithm in real case studies using the Nielsen Consumer Panel data, where we investigate a few metrics to compare more aspects between the two models' learning outcomes.

It does not only remove some of the restrictive assumptions present in standard MNL models, more importantly, it is more suitable for modeling real world scenarios. In addition, the learning outcome can be used in a variety of ways in many downstream tasks in areas such as marketing, operations research, urban planning, etc. We will describe some sample use cases in the next section.

5.2 Use Cases

- Consumer Segmentation

  Under the linear utility model, the utility for choosing option j for the k-th mixture is $\sigma(\mathbf{z}_j; \boldsymbol{\beta}_k) = \boldsymbol{\beta_k} \cdot \mathbf{z}_j$, where $\boldsymbol{\beta}_k, \mathbf{z}_j \in \mathbb{R}^d$. We can think of these $\boldsymbol{\beta}_k$ as a unique "preference vector" associated with each consumer type. It represents the taste of the consumers over different attributes of the options. Such information can then be utilized by manufacturers and retailers to better design their product design and marketing strategies. As an example, Kamakura and Russell [1989] used MMNL to model brand preferences and created market structure that links the pattern of brand switching with price elasticities. The result provides a 'managerially useful description of brand competition", which then in turn allows them to explore the characteristics of competition between national brands and private labels.

  From another point of view, we can also think of $\boldsymbol{\beta}_k$'s as interpretable *user embeddings*. The concept of embedding is widely used in a variety of machine learning settings, including natural language processing and computer vision, which is a relatively low-dimensional vector that can be used to quantify similarities and distances between complex and/or non-numerical objects.

- Multi-product pricing problem

  While MMNL are discrete choice models, they can also be utilized as tools to model relative consumer demands. In particular, MMNL models provide an accurate demand model thanks to its capability of capturing the heterogeneity in the consumer population, making it a fundamental tool for demand prediction in revenue management and supply chain management.

  An important application that benefits from MMNL demand models is the *multi-product pricing problem.* The objective is to maximize the total revenue by finding the optimal price for a set of M products:

$$\begin{aligned} \max_{\boldsymbol{p}} \quad & \sum_{j=1}^{M} p_j \sum_{k=1}^{K} \alpha_k \frac{\exp \sigma(\mathbf{z}_j, \boldsymbol{p}; \boldsymbol{\beta}_k)}{1 + \sum_i \exp \sigma(\mathbf{z}_j, \boldsymbol{p}; \boldsymbol{\beta}_k)} \\ \text{s.t.} \quad & p_j \geq 0 \qquad \forall j \in [M] \end{aligned}$$

  First note that the decision variables $\boldsymbol{p} \in \mathbb{R}^M$ are also part of the utility function σ. This is due to the fact that in the multi-product pricing setting, prices — not only the price of a product itself, but also prices of other products in the same set — are often an important factor that will impact people's choice behavior. Because of this entanglement, it does not suffice if we only learn the aggregated choice probabilities of the population. Instead, we have to accurately estimate the parameters for each individual MNL mixture before we can solve this optimization problem.

  Note in the above formulation, we also include an offset option, which allows the consumer to choose not to purchase anything from the set. The probability of the offset can be expressed as $\dfrac{1}{1 + \sum_i \exp \sigma(\mathbf{z}_j, \boldsymbol{p}; \boldsymbol{\beta}_k)}$.

Bibliography

E. S. Allman, C. Matias, and J. A. Rhodes. Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics*, 37(6A): 3099 – 3132, 2009.

A. Anandkumar, D. Hsu, and S. M. Kakade. A method of moments for mixture models and hidden markov models. In S. Mannor, N. Srebro, and R. C. Williamson, editors, *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *Proceedings of Machine Learning Research*, pages 33.1–33.34, Edinburgh, Scotland, 25–27 Jun 2012. PMLR.

F. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning*, 6(2-3):145–373, 2013.

M. Belkin and K. Sinha. Polynomial learning of distribution families. *SIAM Journal on Computing*, 44(4):889–911, 2015.

S. Berry, J. Levinsohn, and A. Pakes. Automobile prices in market equilibrium. *Econometrica: Journal of the Econometric Society*, pages 841–890, 1995.

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

J. Boyd and R. E. Mellman. The effect of fuel economy standards on the u.s. automotive market: An hedonic demand analysis. *Transportation Research Part A: General*, 14(5):367–378, 1980. ISSN 0191-2607. doi: https://doi.org/10.1016/0191-2607(80)90055-2. URL `https://www.sciencedirect.com/science/article/pii/0191260780900552`.

D. Brownstone and K. Train. Forecasting new product penetration with flexible substitution patterns. *Journal of Econometrics*, 89(1):109–129, 1998. ISSN 0304-4076. doi: https://doi.org/10.1016/S0304-4076(98)00057-8. URL `https://www.sciencedirect.com/science/article/pii/S0304407698000578`.

D. Brownstone, D. S. Bunch, and K. Train. Joint mixed logit models of stated and revealed preferences for alternative-fuel vehicles. *Transportation Research Part B: Methodological*, 34(5):315–338, 2000.

N. Cardell and F. C. Dunbar. Measuring the societal impacts of automobile downsizing. *Transportation Research Part A: General*, 14(5):423–434, 1980. ISSN 0191-2607. doi: https://doi.org/10.1016/0191-2607(80)90060-6. URL https://www.sciencedirect.com/science/article/pii/0191260780900606.

F. Chierichetti, R. Kumar, and A. Tomkins. Learning a mixture of two multinomial logits. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 961–969. PMLR, 10–15 Jul 2018.

P. K. Chintagunta, D. C. Jain, and N. J. Vilcassim. Investigating heterogeneity in brand preferences in logit models for panel data. *Journal of Marketing Research*, 28(4):417–428, 1991. ISSN 00222437. URL http://www.jstor.org/stable/3172782.

J. S. Chipman. The foundations of utility. *Econometrica*, 28(2):193–224, 1960. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1907717.

J. de Dios Ortuzar. Nested logit models for mixed-mode travel in urban corridors. *Transportation Research Part A: General*, 17(4):283–299, 1983. ISSN 0191-2607. doi: https://doi.org/10.1016/0191-2607(83)90092-4. URL https://www.sciencedirect.com/science/article/pii/0191260783900924.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

W. H. Greene and D. A. Hensher. A latent class model for discrete choice analysis: contrasts with mixed logit. *Transportation Research Part B: Methodological*, 37(8):681–698, 2003. ISSN 0191-2615. doi: https://doi.org/10.1016/S0191-2615(02)00046-2. URL https://www.sciencedirect.com/science/article/pii/S0191261502000462.

P. M. Guadagni and J. D. C. Little. A logit model of brand choice calibrated on scanner data. *Marketing Science*, 27(1):29–48, 2008. doi: 10.1287/mksc.1070.0331.

Z. Harchaoui, A. Juditsky, and A. Nemirovski. Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming*, 152 (1-2):75–112, 2015.

S. Jagabathula, L. Subramanian, and A. Venkataraman. A conditional gradient approach for nonparametric estimation of mixing distributions. *Management Science*, 2020.

M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 427–435, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/jaggi13.html.

M. Jaggi and M. Sulovskỳ. A simple algorithm for nuclear norm regularized problems. In *ICML*, 2010.

A. T. Kalai, A. Moitra, and G. Valiant. Efficiently learning mixtures of two gaussians. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, STOC '10, pages 553–562, 2010.

W. A. Kamakura and G. J. Russell. A probabilistic choice model for market segmentation and elasticity structure. *Journal of Marketing Research*, 26(4):379–390, 1989. ISSN 00222437. URL `http://www.jstor.org/stable/3172759`.

S. Lacoste-Julien and M. Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *Advances in neural information processing systems*, pages 496–504, 2015.

R. D. Luce. *Individual Choice Behavior.* John Wiley, 1959.

C. F. Manski. The structure of random utility models. *Theory and Decision*, 8(3): 229–254, 1977. doi: 10.1007/BF00133443.

D. McFadden and K. Train. Mixed mnl models for discrete response. *Journal of Applied Econometrics*, 15(5):447–470, 2000.

A. Moitra and G. Valiant. Settling the polynomial learnability of mixtures of gaussians. *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 93–102, 2010.

K. Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110, 1894.

R. L. Plackett. Random permutations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 30(3):517–534, 1968.

R. L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(2):193–202, 1975.

S. Ragain and J. Ugander. Pairwise choice markov chains. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

D. Revelt and K. E. Train. Mixed logit with repeated choices: Households' choices of appliance efficiency level. *Review of Economics and Statistics*, 80:647–657, 1998.

P. R. Rider. The Method of Moments Applied to a Mixture of Two Exponential Distributions. *The Annals of Mathematical Statistics*, 32(1):143 – 147, 1961.

A. Seshadri, S. Ragain, and J. Ugander. Learning rich rankings. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9435–9446, 2020.

W. Tang. Learning an arbitrary mixture of two multinomial logits, 2020.

K. E. Train. Em algorithms for nonparametric estimation of mixing distributions. *Journal of Choice Modelling*, 1(1):40–69, 2008.

K. E. Train. *Discrete choice methods with simulation.* Cambridge university press, 2009.

K. E. Train, D. McFadden, and M. E. Ben-Akiva. The demand for local telephone service: a fully discrete model of residential calling patterns and service choices. *The RAND Journal of Economics*, 18:109–123, 1987.

J. G. Wendel. A problem in geometric probability. *Mathematica Scandinavica*, 11(1): 109–111, 1962.

Z. Yang, Z. Dai, R. Salakhutdinov, and W. W. Cohen. Breaking the softmax bottleneck: A high-rank rnn language model. In *ICLR*, 2018.

Z. Yang, T. Luong, R. Salakhutdinov, and Q. V. Le. Mixtape: Breaking the softmax bottleneck efficiently. In *NeurIPS*, 2019.

Z. Zhao and L. Xia. Learning mixtures of plackett-luce models from structured partial orders. *Advances in Neural Information Processing Systems*, 32, 2019.

Z. Zhao, P. Piech, and L. Xia. Learning mixtures of plackett-luce models. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2906–2914. PMLR, 20–22 Jun 2016.