# Modeling Heterogeneous User Behavior in Interactive Systems by Graphical Model and Collaborative Learning Framework

Jingshuo Feng

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2021

Reading Committee:

Shuai Huang, Chair

Prashanth Rajivan

Chaoyue Zhao

Program Authorized to Offer Degree:
Industrial & Systems Engineering

University of Washington

## Abstract

Modeling Heterogeneous User Behavior in Interactive Systems
by Graphical Model and Collaborative Learning Framework

Jingshuo Feng

Chair of the Supervisory Committee:
Professor Shuai Huang
Industrial & Systems Engineering

In recent years, the rapid technological innovations of smart personal technologies have given rise to the growth of smart apps that can interact with users and implement personalized incentives to coordinate and change user behaviors in various realms such as e-commerce, patient-centered health system, and individual level transportation demand management (TDM) systems. Understanding user behaviors is crucial for further intervention strategy development and user experience optimization, hence the key to the success of the emerging applications.

However, the existing statistical models encounter challenges when facing the unique characteristics of the systems, e.g., the user-system interactions make the apps more than data collection tools, but they also interfere with the user and change the user's behavior; the users are heterogeneous in their preferences but data of a single user is limited and fragmented; the massive user base and its complicated structure will affect personalized learning and recommending. This dissertation develops novel models to address the aforementioned challenges based on collaborative learning framework, graphical models, and deep matrix factorization.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGMENTS

I would first of all like to express sincere appreciation to my advisor, Professor Shuai Huang, for the guidance and support he gave me throughout my way to the Ph.D. degree. Your expertise and patience truly lead me to have learned a lot in this journey. From general abilities like writing, communicating, to the specific technical challenges in conducting experiments, the help from you is always so supportive. It is an honor to work with and learn from you. I would also like to express my gratitude to my committee members. Thank you, Professor Cynthia Chen, for your extensive knowledge and insights in the fields of transportation and behavior realms, which allow our work more down-to-earth. Thank you, Professor Rajivan Prashanth and Professor Chaoyue Zhao, for your valuable comments and professional suggestions in the methodologies, which helped me improve my works.

Great thanks also go to my collaborators and colleagues. I acknowledge Dr. Xi Zhu for her contribution to the Time-varying model and algorithm. Thank you as well for the days we have discussed our works together. Thanks to Feilong Wang, for your ideas and comments. I also want to thank Dr. Aven Samareh for giving me advice on the way forward. I am also grateful to the ISE department and those who helped me in my completion of Ph.D. studies. Thanks to Ameer Hamza Shakur, Yi Yang, and Wengeng Pan, and all my fellow friends in the B14. I feel very privileged to have you as a companion.

There are many others who have given me warm hands during my pursuit. Great thanks to Professor Xiangyu Chang, Taiyun Wei, and Dr. Lei Yuan, Dr. Ji Liu, Dr. Weicong Ding. Without you, I could not achieve what I have now.

Last but never least, to my dearest parents, thank you all for your unconditional love and supports for all these years, even in the toughest days.

# Chapter 1
# INTRODUCTION

## *1.1 Motivation*

In recent years, the rapid technological innovations of smart personal technologies have given rise to many smart apps that could interact with users and implement personalized incentives to change and coordinate user behaviors in various realms such as online retailers with targeted advertising [129, 70], patient-centered health system that aim to provide personalized services [63, 130, 93], and individual level transportation demand management (TDM) systems [6, 4, 132]. Such apps show unprecedented potentials. For example, traditionally, TDM strategies will work generically on a population or a large group of people, by providing incentives or issuing costs to certain travel behavior in order to modify people's travel behavior, examples including offering low-cost public transportation, increasing parking costs in peak hours, and monetary rewards, etc [13, 14, 100, 124]. On the contrary, the app-based TDM systems developed recently are able to achieve real-time and personalized management and optimization [6, 132]. Therefore, unlike traditional generic TDM systems which gained limited success [110, 88], the personalized systems have shown the capability in system-wide energy-saving and congestion mitigation in previous research [6, 132].

To fully unleash the potential of such personalized interactive reward systems, it is crucial to understand user decision-making behavior and estimate user preferences. It is an enabling factor to further intervention strategy development and optimization for user experience. However, as new technologies and new communication tools, the data collected and generated by these apps have several unique characteristics and will lead to corresponding statistical challenges in modeling.

First, as the systems provide personalized services and real-time services, it is important to model the intra- and inter-individual heterogeneity. To be specific, for each user, his/her preferences may evolve over time and will lead to influences on future behaviors. Such intra-individual heterogeneity, also known as preference shift or preference reversal phenomena, often requires dynamic models to explain [50, 29, 9]. Inter-individual heterogeneity refers to the different preferences among different users. Much research has studied this issue by employing users' diverse demographic or socio-economic characteristics [49, 31, 66], or latent psychological factors such as attitudes, perception and attention [11, 10, 115]. Further, in these emerging apps, building individualized models of heterogeneous populations is even more challenging. For one reason, the personal behavior data is usually limited and fragmented. This involves a long-standing problem in statistics and machine learning, considering the huge heterogeneity of the population and the prohibiting cost to collect high quality and sufficient data for each individual [35, 27]. On the other hand, for a mass population with a large number of individuals, the heterogeneity in preferences or characteristics is complicated, such as online shopping preferences and patient documents [18, 105, 39, 3].

Another aspect of the modeling challenge stems from the interactive nature of the system. When users use the app, new data is generated as the system collects data, and the new data is tied to or based on the interaction history. In other words, the data collection procedure is not irrelevant to user behaviors anymore, but will interfere with user behaviors. In online shopping, for example, a particular user will always browse for specific item categories, and the system will need to remember such features so that it can show the user more relevant products in the future [62, 107]. In the personalized TDM system, user preferences will affect the designation of alternative plans and the price of those as well [6, 132]. All these interactions will create problems such as data endogeneity [47, 46] or multicollinearity [58, 103, 32] between the data collection and the user behaviors. It is a unique challenge aroused by interactive systems, and very few existing works can be applied directly to this situation. It may lead to inconsistent estimations and cause difficulties in understanding and explaining behaviors, and special treatment or correction in analyzing is needed.

This dissertation develops novel models to address the aforementioned challenges based on collaborative learning framework [77, 78], graphical models [69], and deep matrix factorization [128, 117]. The proposed models are applied on a real-world dataset collected from a personalized TDM system [132]. They are capable of learning distinct individual-level behavior models, discovering typical or special behavior patterns, and revealing the complicated user heterogeneity structure.

## *1.2 State-of-the-Art*

Corresponding to the different aspects of statistical challenges, several existing research areas are related to the topic and shed light on this work.

### *1.2.1 Discrete Choice Models*

Discrete choice models have been extensively used in modeling user behaviors over the decades, such as the theory of Random Utility Maximization (RUM) [82, 12] and Random Regret Minimization (RRM) [24, 57]. Recently, Hybrid Choice Model (HCM) framework has gained increasing attention to account for latent effects within the discrete choices, such as latent psychological variables like attitudes and perception or latent classes [11, 115], to approximate complicated choice behaviors in reality. There are also works focusing on data endogeneity where the error term is correlated with the choice, using latent variable methods [47, 48]. However, discrete choice models are traditionally developed on a population base and estimate the average effects of groups of people. Individual-level data is often limited and fragmented in personalized apps, which may cause difficulties for discrete choice models. Besides, few works in this realm are specifically designed to characterize the user interaction with the app-based reward systems.

### *1.2.2 Collaborative Learning*

The Mixed Effect Model (MEM) has been a long-standing method to handle heterogeneity of individual models [113, 120, 114, 38, 108]. Although the primary motivation of MEM is

not for personalized modeling, it provides an approach to estimate different models for each individual as it incorporates a level-two distribution model to characterize the variations of the level-one individuals. However, it encapsulates the heterogeneity into random effects.

The collaborative learning framework overcomes such issue and is the state-of-art personalized modeling idea [77, 78]. It portrays the heterogeneous population by learning a shared set of canonical models among all users and a unique membership vector for each user. One canonical model can be considered one behavior pattern or decision mechanism, and membership vectors represent the different degrees of the resemblance of the individual models to the canonical models. By combining the canonical models using membership vector as the weights, the individual models can provide an adequate characterization of the individuals.

In Chapter 2, we integrate the collaborative learning framework with random utility maximization (RUM) and propose Personalized RUM models (Logistic Collaborative Model, LogCM), to learn personal preferences. We formulate the model as an optimization problem, and propose extensions of the LogCM framework in Chapter 3, the Pairwise-fusion LogCM (LogPCM) and the LogCM with Time-varying preferences (LogCM-T), for more complex data structures. LogPCM provides a data-driven technique to discover the canonical model structure in the heterogeneous population, and LogCM-T can cope with the intra-individual heterogeneity of preference shifts.

### *1.2.3 Deep Matrix Factorization*

Matrix factorization is useful when there is a low-rank structure in data. Multi-layer matrix factorization is popular lately for its capability of discovering different levels of hidden attribute representations and the corresponding hierarchical structure [127, 128, 118, 117]. As collaborative models also have a delicate low-rank structure similar to matrix factorization, for a large population with a complex heterogeneity structure, it would also be beneficial to extend the one-layer canonical structure into multi-layers.

In Chapter 4, Hierarchical Collaborative Model (HCM) is proposed. As an extension of collaborative learning, the HCM model preserves the advantage of the ordinary collabora-

tive model that it can efficiently learn individual models when individual data is limited, and discover commonality in the heterogeneous population. Furthermore, the hierarchical structure in HCM makes it suitable for large complex populations, and enables it to understand user behavior under different levels of detail and granularity. We also introduce Contextual HCM (CHCM) to employ useful characteristic information of the individuals in the large heterogeneous population.

#### *1.2.4 Graphical Models*

Graphical model refers to a family of multivariate statistical models that specifically model the interactions among variables and derive their data-generating process [69]. Using the graph, the model can represent the conditional dependencies among variables, enabling the graphical model to reveal the relationship between variables. It overcomes the shortcomings of discrete choice models which lack useful tools to characterize the interactions.

In Chapter 5, we propose an innovative graphical model to model the user-app interaction mechanism. The model is called the Latent Decision Threshold (LDT) model that shows promising results in understanding and discovering user behaviors with better interpretability compared with discrete choice models. In addition, we integrated the LDT model with max-margin learning, so that we resorted to a very computationally efficient algorithm for parameter estimation, unlike many costly algorithms for graphical models [73, 123].

### *1.3 Organization of the Dissertation*

This dissertation is organized according to the following structure. Chapter 2 proposes the Logistic Collaborative Model (LogCM) framework, based on the random utility maximization theory (RUM) and collaborative learning framework. LogCM framework addresses the challenge of learning distinct individual models with limited data size. In Chapter 3, we introduce two extensions of the basic LogCM to fit better specific real-world applications, the Pairwise-fusion LogCM (LogPCM) and the LogCM with Time-varying preferences (LogCM-T). Next, we develop a multi-layer collaborative model, named Hierarchical Collaborative

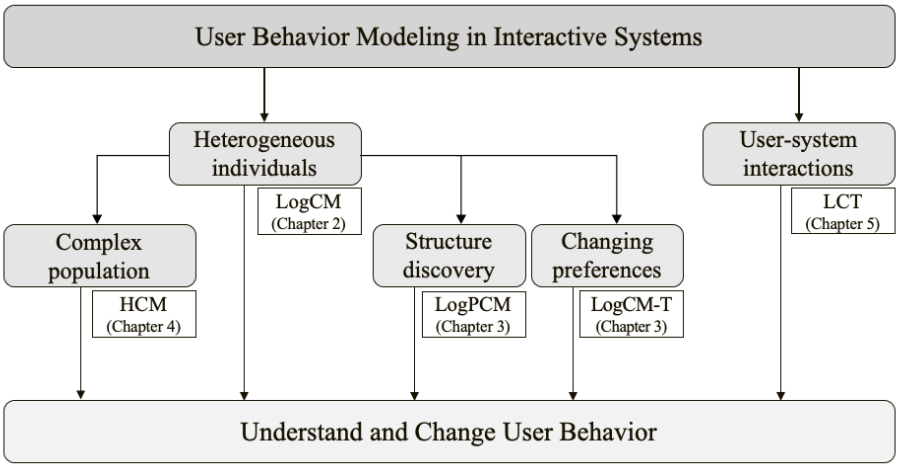


Figure 1.1: The organization of the dissertation.

Model (HCM) in Chapter 4. It has multiple layers of the canonical structure, thus allows for the interpretation of heterogeneous populations at different levels. We also provide Contextual HCM (CHCM) in Chapter 4 which could match the model structure and the population composition structure so that is suitable for a large population with complex composition. The challenge of system interfering with user behaviors will be studied in Chapter 5. The proposed Latent Decision Threshold (LDT) model combines the idea of graphical model and the max-margin learning, and provides a better characterization of the data collection, user-system interactions in these emerging systems. The conclusion and possible future works are briefly discussed in Chapter 6. The relationships among these works and chapters are depicted in Figure 1.1

# Chapter 2

# LOGCM: LOGISTIC COLLABORATIVE MODEL FOR PERSONALIZED RANDOM UTILITY MAXIMIZATION (RUM) MODELING OF USER BEHAVIOR

In this chapter, we address the challenge of understanding user behavior at the individual level and propose the Logistic Collaborative Model (LogCM) [34]. It builds on the concepts such as canonical structure and membership vectors invented in recent works on collaborative learning [76, 77, 78] and is suitable for modeling a heterogeneous population with insufficient data from each individual. A computationally competent algorithm is developed to solve the corresponding optimization formulation. Extensive simulation studies and a real-world application in a smart Transportation Demand Management (TDM) system [132] show the effectiveness of our proposed methods.

## *2.1 Introduction*

The heterogeneous population has been observed in many applications such as modeling in many engineering systems and healthcare problems [77]. In the emerging applications that provide personalized services for target users, it is crucial to take into account the heterogeneity of the crowd, and understand user behavior at the individual level, i.e., learn a distinct behavior model for each user and understand his/her own preferences. For example, recently there has been much research attention on smart Transportation Demand Management (TDM) due to the rising high demand in driving which is closely related to a number of urban issues, such as congestion, air pollution, and public health [65]. TDM strategies are designed to modify travel behaviors by providing travelers incentives (or costs) to certain travel behaviors, for instance, promoting transit use by offering low-cost passes. Personalized

incentive holds great promise to solve many challenges in TDM [41, 110, 88], leveraging on the rapid proliferation of smart personal technologies that make it possible to offer incentives individually [132, 22] rather than an average generic incentive.

The key to the success of any personalized service-providing system is a statistically accurate and efficient personalized model that can estimate individuals' preferences from their behavior data. It has been found that the random utility maximization (RUM) model is an effective tool to learn preferences from data [83, 12, 121, 91, 48, 25, 72, 55]. The challenge is that the personal behavior data is usually limited and fragmented. This actually involves a long-standing problem in statistics and machine learning, considering the vast heterogeneity of the population and the prohibiting cost to collect high-quality and sufficient data from each individual. Most prediction models are learned by pooling the individuals' data together and creating a population model, thus ignoring individuals' variations by only characterizing the average effect. To address this problem, our framework builds on the concepts such as canonical structure and membership vectors invented in recent works on collaborative learning [76, 77, 78] and is suitable for modeling heterogeneous population with insufficient data from each individual.

The personal behavior data itself can be challenging to model. For example, as described in [132], the personal behavior data is collected from each individual who is asked to choose between a promoted sustainable plan and his/her original travel plan. The RUM-based models assume that the probability of choosing among multiple alternatives depends only on the differences in their respective utilities, and an individual will select the alternative that provides the maximum utility. Here, the utility is a concept quantifying the attractiveness of an alternative in a choice scenario, and it is assumed to be indirectly related to the various characteristics of the alternative, the individual and the surrounding environment [12, 53]. In reality, the utilities of alternatives are usually unobservable. To see that, for example, given two alternatives $A$ and $B$, the only information we can observe is the final choice which only indicates that the probability $Pr(U_A \geq U_B) \geq 0.5$ when the individual chooses A ($U_A$ is the utility of alternative A), otherwise, $Pr(U_A \geq U_B) \leq 0.5$. Neither the true probability

nor utilities can be directly observed. Thus, learning preferences for each individual from sparse and fragmented behavior data adds another level of complexity.

To learn user behavior at the individual level, we utilize the theory of RUM and propose a novel logistic collaborative model (LogCM) to address the aforementioned issues. Collaborative learning framework [76, 77, 78] is one of the state-of-the-art personalized modeling methods which can learn distinct personalized models for each individual, even when each individual's data is limited. Random Utility Maximization (RUM theory) has a solid behavior basis and could provide the proposed models with good interpretability to understand user behavior. A set of canonical models will be learned to represent the heterogeneity of the population. Each canonical model can be considered as a representation of one behavior pattern or decision mechanism. It is usually unknown that which mechanism an individual may follow, and some individuals may exhibit a mix of those patterns. Thus, mathematically these canonical models span the modeling space for the individuals and provide a basis to characterize the individuals' variations. We then learn a membership vector for each individual, which represents the degrees of resemblance between the model of an individual and the canonical models. With the knowledge of the canonical models and membership vectors, common patterns are found, and individual models can be derived. The collaborative learning framework is easy to explain and suitable for heterogeneous populations with insufficient data from each individual since it considers both the commonalities among individuals and the characteristics of each individual by learning canonical models and membership vectors, respectively. This novel collaborative learning model leads to a non-linear and non-convex constrained optimization problem, which could be solved by our proposed two-step iterative algorithm.

The proposed work is different from some ongoing works in the literature that aim to provide remedy for RUM for various types of complications in real applications. Most of these models are not designed for learning behavior models at the individual level. For example, Azari et al. [5] aimed to learn utilities of a set of alternatives with rank data and did not estimate personal preferences for each individual. Guevara and Ben-Akiva [47]

dealt with the endogeneity caused by model misspecification (i.e., omission of the attributes). Ben-Akiva et al. [10] incorporated contexts like social network as the decision being made may also be affected by family, friends, and other choices being given. Hancock et al. [50] used a dynamic model from decision field theory to characterize the changing preferences with sequential choices and decisions. However, few literature has systematically tackled the problem of personalized modeling in the framework of RUM theory. One exception is the mixed logit model (MLM) [90, 9] that can deal with heterogeneous population where parameters are assumed to vary across individuals. It is also known as mixed effects logistic models (logistic MEM) where the distinct part of parameters (i.e., random effects) across different individuals are sampled from a distribution. An extensive comparison between MLM with our proposed model will be found in the numerical studies.

The work of LogCM is organized as follows. Section 2.2 presents the details of the proposed logistic collaborative model (LogCM) and an intuitive extension of it, the similarity-regularized LogCM (LogSCM). Related methods like RUM and mixed effect models (MEM), and the relationships between them and the proposed model are discussed. Section 2.3 provides a two-step iterative algorithm for learning the parameters in the proposed models. Implementation guidelines in practice are also discussed around issues such as initializing and hyperparameter tuning. Section 2.4 evaluates the proposed methods on comprehensive simulation studies. A real-world case study is shown in Section 2.5, followed with a detailed discussion of the meanings of canonical models and the collaborative structure. To better fit different application scenarios, we can further extend the LogCM framework with additional structural designs, and we provide two innovative models in detail in the next chapter, Chapter 3. Some supplementary materials are provided in Appendix A.

## *2.2 The Logistic Collaborative Model (LogCM)*

In this section, we present the logistic collaborative model (LogCM) for personalized modeling in a heterogeneous population. Here, we develop the LogCM with the assumption that the decisions made by users are binary, since this is the most common decision scenario in

practice and multiple choice outcomes could always be converted into binary outcomes, e.g., if multiple products are presented to an user, we may use binary outcome variables to indicate the "buy" or "not buy" for each product by the user. We first show that the RUM-based logit model, used in [132] can be reformulated as a logistic regression (LR), which links the characteristics of the alternative and the outcome using the logistic function (also known as sigmoid function in deep learning). We then develop the mathematical formulation of the LogCM. We also derive its connection with the mixed effects logistic model (logistic MEM) [113, 52], i.e., MLM. [90] .

### *2.2.1 Relationship between Logistic Regression and RUM*

As mentioned in Section 2.1, RUM assumes that the probability of selecting among alternatives depends only on the differences in their utilities. In other words, an individual will assign the highest probability to select the alternative which provides the maximum utility [12, 53, 55], where the utility is defined to be indirectly related to the various characteristics of the alternative. For instance, for a scenario with two alternatives, A and B, the decision-making problem is like a binary classification where the concept of utility for any alternative is a function of some variables that characterize the alternative. Specifically, based on the theory of RUM, for two alternatives, the probability of choosing alternative B can be written as $Pr(U_B \geq U_A)$, where $U_A = V_A + \epsilon_A,\ U_B = V_B + \epsilon_B$ are utilities associated with alternative $A$ and $B$, respectively, with $V$ representing the "systematic utility" and $\epsilon$ representing the "random utility" [12]. Then, we can analytically use the utility ratio [132] to represent the probability that an individual will choose alternative B rather than alternative A as $R_B = (e^{U_B}/(e^{U_B} + e^{U_A})$. If $R_B \geq 0.5$, the individual has a higher probability to choose alternative B, otherwise, $R_A \geq 0.5$, the individual will be more likely to choose alternative A.

To characterize the difference between two alternatives, assume that there are $p$ variables and we can define the difference of the two alternatives on these $p$ variables as $x_1$, $x_2$, ..., $x_p$. As only the difference matters to decide which alternative is more favorable, we could

arbitrary appoint one alternative as baseline, i.e., we could set $V_A = 0$ if alternative A is the baseline. Then, we can define the utility of alternative B as $V_B = \sum_p \beta_p x_p$. Since it is assumed that the systematic utilities represent the predictable part in decision-making, which are characterized by some variables that define the alternatives, and the random utilities are not observable, the utility ratio can be simplified by eliminating the random utilities as:

$$R_B = \frac{e^{V_B}}{e^{V_A} + e^{V_B}} = \frac{\exp(\boldsymbol{x}^\top \boldsymbol{\beta})}{1 + \exp(\boldsymbol{x}^\top \boldsymbol{\beta})}. \tag{2.1}$$

Eq.(2.1) is mathematically the same as the logistic function in logistic regressions (LR). Thus, using the definition of utility ratio under binary decision-making cases, the logit model based on RUM is identical to the LR. As here we focus on binary decision-making outcomes, RUM and LR models could be used interchangeably.

### *2.2.2 The Framework of Collaborative Learning*

As personalized modeling often encounters the problem of lack of data, now we present how the collaborative learning model is created for learning personalized models in our target application. Usually, a multivariate statistical model requires a considerable amount of samples for reliable estimation. For instance, it is a commonly held belief that the proportion of sample size over number of parameters should be at least 30 in linear regression models with continuous outcomes [92]. It is therefore expected to be more demanding on the sample size if the outcomes are binary [59]. The high demand in sample size results in a challenge for robust estimation for personalized modeling given limited data size.

To overcome the problem of limited data, we adopt the collaborative learning framework that has been shown as an effective model in a range of engineering and healthcare applications [77, 76, 78]. The general idea of collaborative learning is to exploit the canonical structure that is embedded beneath the heterogeneity of a given population. An exemplary illustration is shown in Figure 2.1. Denote the canonical models as $f_k(x), k = 1, \ldots, K$, which can represent some common patterns or typical types from the heterogeneous $N$ individuals.

The number of canonical models, which could be determined by data-driven approaches as we will show later, is usually much smaller than the number of individuals (i.e., $K \ll N$), granting the advantage of collaborative learning to reduce the burden of estimating a large amount of free parameters.

With the knowledge of the canonical models, each individual model could be characterized as an integration of the canonical models. Here, we assign a membership vector $\boldsymbol{c}_i = [c_{i1}, \ldots, c_{iK}]^\top, i = 1, \ldots, N$ to each individual $i$ to represent the degrees of resemblance of the individual model to the canonical models. In other words, we assume that the model of each individual is a combination of the canonical models, and the weights are the elements of the corresponding membership vector. Since each canonical model describes one kind of mechanism patterns in the population, by integrating this set of canonical models, the individual models, denoted as $g_i(x) = \sum_k c_{ik} f_k(x), i = 1, \ldots, N$, can provide an adequate characterization of the individuals. Specifically, in the models with individual parameters $\boldsymbol{\beta}_i = [\beta_{i1}, ..., \beta_{ip}]^\top, i = 1, \ldots, N$, note that each canonical model is a model of the same form with a parameter vector, i.e., $\boldsymbol{q}_k = [q_{k1}, ..., q_{kp}]^\top, k = 1, ..., K$. Under the collaborative learning framework, we can assume that $\boldsymbol{\beta}_i$ of the model of individual $i$ is a linear combination of the canonical parameters, i.e., $\boldsymbol{\beta}_i = \sum_k c_{ik} \boldsymbol{q}_k$.

For example, in our work, the forms of canonical models and the personalized models are both logistic models, i.e., for $k$-th canonical model, $f_k(x) = \log(\Pr(y = 1))/(1 - \Pr(y = 1)) = \boldsymbol{x}^\top \boldsymbol{q}_k$, where $\boldsymbol{q}_k$ is the parameter vector of this canonical model. Thus, under the collaborative learning framework, the model of individual $i$ is $g_i(x) = \sum_k c_{ik} \boldsymbol{x}^\top \boldsymbol{q}_k = \boldsymbol{x}^\top \sum_k c_{ik} \boldsymbol{q}_k$, and $\boldsymbol{\beta}_i = \sum_k c_{ik} \boldsymbol{q}_k$ is the personalized parameter vector for the individual.

Next, we will present the formulation of our proposed LogCM in detail and an intuitive extension, the similarity-regularized logistic collaborative model (LogSCM).

### *2.2.3 Model Formulation of LogCM and LogSCM*

To derive the analytical formulation of LogCM, we first tidy up the parameters of the $K$ canonical models as a matrix: $\mathbf{Q} := [\boldsymbol{q}_1, ..., \boldsymbol{q}_K] \in \mathbb{R}^{p \times K}$. Then, we can rewrite $\boldsymbol{\beta}_i$ as $\boldsymbol{\beta}_i = \mathbf{Q} \boldsymbol{c}_i$.

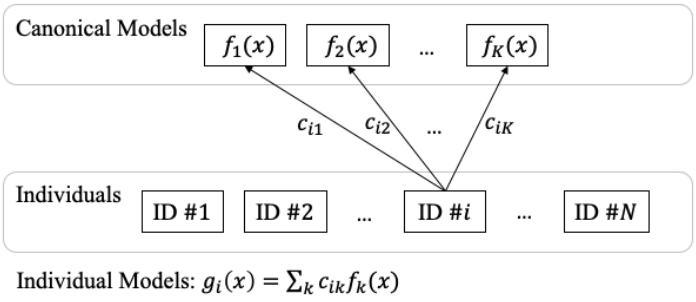


Figure 2.1: Schematic of the collaborative learning framework.

LR is a widely used statistical model for binary-outcome problems. It assumes that the probability of being in a certain category depends on a set of variables ($x$'s), with the link of logit function. Under the collaborative learning framework, the logistic function can be expressed as:

$$\pi_i(\boldsymbol{x}_{ij}) = \Pr(y_{ij} = 1) = \frac{\exp(\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}_i)}{1 + \exp(\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}_i)} = \frac{\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}\boldsymbol{c}_i)}{1 + \exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}\boldsymbol{c}_i)}. \tag{2.2}$$

$\pi_i$ is the logistic regression model for individual $i$, where $y_{ij}$ is the $j$-th binary observation of this individual and $\boldsymbol{x}_{ij}$ is the $p$-length characteristic variables vector. LR is always learned by maximizing the log-likelihood, which can be written as:

$$l = \log(1 + \exp(\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}_i)) - y_{ij}(\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}_i). \tag{2.3}$$

Note that it is also the logistic loss function in machine learning. We can see from Eq.(2.3) that we can learn the parameters without knowing the latent variable given by $\boldsymbol{x}^\top \boldsymbol{\beta}$ (the systematic utility in the context of RUM). It is straightforward to write up the log-likelihood function under collaborative learning framework for parameter estimation:

$$\begin{aligned} \min_{\mathbf{C},\mathbf{Q}} \quad & \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \left\{ \log(1 + \exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}\boldsymbol{c}_i)) - y_{ij}(\boldsymbol{x}_{ij}^\top \mathbf{Q}\boldsymbol{c}_i) \right\}, \\ \text{s.t.} \quad & \boldsymbol{c}_i \geq 0, \ \boldsymbol{c}_i^\top \mathbf{1} = 1 \quad i = 1, ..., N. \end{aligned} \tag{2.4}$$

Here, in our Logistic Collaborative Model (LogCM), note that the objective function is a weighted sum of the logistic loss of all individual models to gauge the goodness-of-fit of the models. $n_i$ is the number of observations of individual $i$. The goal of applying the weight $1/n_i$ is to account for the different sample sizes of different individuals. The two constraints, $\boldsymbol{c}_{ik} \geq 0$ and $\boldsymbol{c}_i^\top \mathbf{1} = 1$, are imposed on $\boldsymbol{c}_i$ due to its definition as a membership vector. By solving this optimization problem, the parameter matrix of the canonical models $\mathbf{Q}$ and the membership vectors $\boldsymbol{c}_i, i = 1, \ldots, N$ can be estimated. Then the individual models can be obtained by $\boldsymbol{\beta}_i = \mathbf{Q}\boldsymbol{c}_i$.

An obvious advantage of formulating the parameter estimation problem as an integrated optimization framework is that it is flexible to incorporate other kinds of data, prior knowledge or any structural constraints that we may want to impose on the models. For instance, in many other applications the similarity information among individuals could be very helpful to learn individual models by allowing similar individuals to have similar models [77]. Denote $w_{lm}$ as the similarity between individuals $l$ and $m$, i.e., the larger $w_{lm}$ is, the more similar is the pair. To incorporate the similarity knowledge in the model formulation of LogCM, we could add a regularization term, $\sum_{l,m} \|\boldsymbol{c}_l - \boldsymbol{c}_m\|^2 w_{lm}$, into the objective function of Eq.(2.4) and extend it to the similarity-regularized logistic collaborative model (LogSCM). Similar as in [17], we can reformulate this regularization term as a trace term which can facilitate the development of our optimization solution in Section 2.3:

$$\begin{aligned} \frac{1}{2}\sum_{l,m} \|\boldsymbol{c}_l - \boldsymbol{c}_m\|^2 w_{lm} &= \sum_{l=1}^{N} \boldsymbol{c}_l^\top \boldsymbol{c}_l d_{ll} - \sum_{l,m} \boldsymbol{c}_l^\top \boldsymbol{c}_m w_{lm} \\ &= \mathrm{Tr}(\mathbf{C}\mathbf{L}\mathbf{C}^\top). \end{aligned} \tag{2.5}$$

Here, $\mathbf{C}$ is the matrix containing all membership vectors $\mathbf{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N] \in \mathbb{R}^{K \times N}$. $\mathbf{L}$ is defined as $\mathbf{D} - \mathbf{W}$, where $\mathbf{W} = (w_{lm}) \in \mathbb{R}^{N \times N}$ is the similarity matrix and $\mathbf{D}$ is a diagonal

matrix with entries $d_{ll} = \sum_m w_{lm}$. Thus, it leads to the following formulation of LogSCM:

$$\begin{aligned} \min_{\mathbf{C},\mathbf{Q}} \quad & \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \left\{ \log(1 + \exp(\boldsymbol{x}_{ij}^\top \mathbf{Q} \boldsymbol{c}_i)) - y_{ij} (\boldsymbol{x}_{ij}^\top \mathbf{Q} \boldsymbol{c}_i) \right\} \\ & + \lambda \operatorname{Tr}(\mathbf{C}\mathbf{L}\mathbf{C}^\top), \\ \text{s.t.} \quad & \boldsymbol{c}_i \geq 0, \ \boldsymbol{c}_i^\top \mathbf{1} = 1 \quad i = 1, ..., N, \end{aligned} \tag{2.6}$$

where $\lambda \geq 0$ is a hyperparameter that can be tuned to control the effect of the regularization term on parameter estimation. The larger $\lambda$ is, the greater influence will be imposed on the estimation by the regularization term.

It is not hard to see that when $\lambda = 0$, the LogSCM formulation in Eq.(2.6) will degenerate to the LogCM formulation in Eq.(2.4). By solving the optimization problem in Eq.(2.6), we can estimate the canonical parameters matrix $\mathbf{Q}$ and the membership matrix $\mathbf{C}$. As the formulation is not jointly convex on both parameter matrices, we will propose an iterative two-step approach to solve them alternatively in Section 2.3.

### *2.2.4 Relationship between LogSCM and Mixed Effects Logistic Model*

The mixed effect model (MEM) has been a long-standing method to handle heterogeneity of individual models [114, 38, 108]. MEM provides an approach for personalized modeling as it incorporates a level-two distribution model to characterize the interrelations of the level-one individuals. Specifically, for logistic regression, mixed effects logistic regression model (logistic MEM) [113, 120] assumes that the random parts of the individual parameters $\boldsymbol{\beta}_i$'s are independently identically distributed, sampled from a multivariate normal distribution, i.e., $\boldsymbol{\beta}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ denotes a covariance matrix. Logistic MEM is mathematically equivalent to the Mixed Logit Model (MLM) under RUM theory, where the individual-level parameters are random samples drew from a multivariate normal distribution [116, 54, 85]. Thus, it is of our interest to study the relationship between our proposed LogSCM and the logistic MEM. We can prove that the objective function of LogSCM is equivalent to the logistic MEM under certain conditions where $w_{lm} = 1/\lambda N$ for all pairs of individuals and

$\boldsymbol{\Sigma} = \mathbf{Q}\mathbf{Q}^\top$.

**Theorem 2.2.** *The objective function of the LogSCM is equivalent to the objective function of logistic MEM when* $\mathbf{W}$ *is a matrix with all entries being* $1/\lambda N$ *and* $\boldsymbol{\Sigma} = \mathbf{Q}\mathbf{Q}^\top$.

The proof of the theorem is provided in Appendix A.1. Theorem 1 shows a useful insight of our proposed collaborative learning approach's unique capability of studying heterogeneous models compared to logistic MEM. Firstly, LogSCM provides greater flexibility of incorporating information sources as the similarity matrix could be freely formed. For MEM, it essentially assumes that $\mathbf{W}$ is a matrix with all entries being $1/\lambda N$. It is not a surprise because the fundamental assumption of mixed effects logistic model is that $\boldsymbol{\beta}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma})$, which treats all individuals equally as independent samples from the same distribution. Further, our model explicitly shows the commonalities and differences in population by providing explicit forms of the canonical models and the membership vectors, while the logistic MEM encapsulates the heterogeneity into random effects. On the other hand, although Theorem 1 reveals the hidden relationship between LogSCM and logistic MEM, it does not indicate that logtistic MEM is simply a special case of the LogSCM. The logistic MEM can apply different forms of covariance matrix, which will lead to a different model from LogSCM. As such, LogSCM can be considered as a knowledge-driven logistic MEM with an extra capability to incorporate the canonical structure and flexible similarity information.

### *2.3 Parameter Estimation Algorithm*

The formulation of LogSCM shown in Eq.(2.6) has a structure that could be utilized to develop a computational algorithm. Specifically, we notice that if we iteratively optimize for $\mathbf{Q}$ and $\mathbf{C}$ in alternation, the optimization problem could be decomposed into two easier subproblems. This strategy has been exploited in [77, 76, 78] for linear regression models and has shown promising performances.

### *2.3.1 Estimation Step for Canonical Models (**Q** Step)*

In this step, we focus on solving $\mathbf{Q}$ with a given $\mathbf{C}^*$, i.e., $\mathbf{C}^*$ could be the latest estimation of $\mathbf{C}$. Given $\mathbf{C}^*$, the regularization term $\mathrm{Tr}(\mathbf{CLC}^\top)$ in Eq.(2.6) is a constant. Therefore, the original problem degenerates to the subproblem:

$$\min_{\mathbf{Q}} \quad \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \left\{ \log(1 + \exp(\boldsymbol{x}_{ij}^\top \mathbf{Q} \boldsymbol{c}_i^*)) - y_{ij} (\boldsymbol{x}_{ij}^\top \mathbf{Q} \boldsymbol{c}_i^*) \right\}. \tag{2.7}$$

To further reveal its structure for the benefit of showing how to solve this optimization problem, we can define

$$\widetilde{\boldsymbol{x}}_{ij} = \widetilde{\mathbf{X}}_{ij}^\top \boldsymbol{c}_i^* \in \mathbb{R}^{pK \times 1},$$

where

$$\widetilde{\mathbf{X}}_{ij} = \begin{bmatrix} \boldsymbol{x}_{ij}^\top & 0 & \cdots & 0 \\ 0 & \boldsymbol{x}_{ij}^\top & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{x}_{ij}^\top \end{bmatrix}_{K \times pK}.$$

Furthermore, denote $\boldsymbol{q} \in \mathbb{R}^{pK \times 1}$ as the vectorized $\mathbf{Q}$, and it is not hard to see that: $\boldsymbol{x}_{ij}^\top \mathbf{Q} \boldsymbol{c}_i^* = \widetilde{\boldsymbol{x}}_{ij}^\top \boldsymbol{q}$. Thus, Eq.(2.7) can be simplified to:

$$\min_{\mathbf{Q}} \quad \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \left\{ \log(1 + \exp(\widetilde{\boldsymbol{x}}_{ij}^\top \boldsymbol{q})) - y_{ij} (\widetilde{\boldsymbol{x}}_{ij}^\top \boldsymbol{q}) \right\}. \tag{2.8}$$

This is a weighted sum of logistic loss, and the logistic loss has been proved convex in literature [86]. Therefore, Eq.(2.8) could be solved by many off-the-shelf algorithms. Here we use an R package called `CVXR` for specifying and solving convex programs [37] to solve the problem in Eq.(2.8).

### *2.3.2 Estimation Step for Membership Vectors (**C** Step)*

In this step, we focus on solving $\mathbf{C}$ with a given $\mathbf{Q}^*$. We briefly show how to solve $\mathbf{C}$ using a closed-form updating rule here and the detailed derivation can be found in Appendix A.2.

Given $\mathbf{Q}^*$, the Lagrangian function of the original formulation as shown in Eq.(2.6) could be derived as:

$$L = \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \left\{ \log(1 + \exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)) - y_{ij} (\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i) \right\} + \lambda \operatorname{Tr}(\mathbf{C}\mathbf{L}\mathbf{C}^\top) + \sum_{i=1}^{N} \eta_i (\boldsymbol{c}_i^\top \mathbf{1} - 1),$$

by introducing the Lagrangian multiplier $\eta_i$ for constraint $\boldsymbol{c}_i^\top \mathbf{1} = 1$. Optimal $\mathbf{C}$ must follow the complementary condition, i.e., $(\partial L / \partial c_{ik}) c_{ik} = 0$:

$$\frac{1}{n_i} \sum_{j=1}^{n_i} \left[ \frac{\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)}{1 + \exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)} - y_{ij} \right] \left( \mathbf{Q}^{*\top} \boldsymbol{x}_{ij} \right)_k c_{ik} + 2\lambda \left( (\mathbf{C}\mathbf{L})_i \right)_k c_{ik} + \eta_i c_{ik} = 0. \tag{2.9}$$

Then, with $\mathbf{L} = \mathbf{D} - \mathbf{W}$, and the constraint that $\boldsymbol{c}_i^\top \mathbf{1} = 1$, the closed-form of $\eta_i$ is:

$$\eta_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \left\{ y_{ij} (\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i) - \frac{\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)}{1 + \exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)} (\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i) \right\} - 2\lambda (\mathbf{C}\mathbf{D})_i^\top \boldsymbol{c}_i + 2\lambda (\mathbf{C}\mathbf{W})_i^\top \boldsymbol{c}_i. \tag{2.10}$$

Plug in the above expression of multiplier $\eta_i$ into the complementary condition in Eq.(2.9), we can generate the following updating rule similar as in [77, 76, 78]:

$$\begin{aligned}
c_{ik}^{(m+1)} = c_{ik}^{(m)} \times \Bigg\{ & \frac{1}{n_i} \sum_{j=1}^{n_i} \left[ -\delta_- \left( \frac{\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i^{(m)})}{1 + \exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i^{(m)})} \left( \mathbf{Q}^{*\top} \boldsymbol{x}_{ij} \right)_k \right) + \delta_+ \left( y_{ij} \left( \mathbf{Q}^{*\top} \boldsymbol{x}_{ij} \right)_k \right) \right] \\
& + \frac{1}{n_i} \sum_{j=1}^{n_i} \left[ -\delta_- \left( y_{ij} (\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i^{(m)}) \right) + \delta_+ \left( \frac{\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i^{(m)})}{1 + \exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i^{(m)})} (\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i^{(m)}) \right) \right] \\
& + 2\lambda \left( (\mathbf{C}^{(m)} \mathbf{W})_i \right)_k + 2\lambda (\mathbf{C}^{(m)} \mathbf{D})_i^\top \boldsymbol{c}_i^{(m)} \Bigg\} \Bigg/ \\
& \Bigg\{ \frac{1}{n_i} \sum_{j=1}^{n_i} \left[ \delta_+ \left( \frac{\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i^{(m)})}{1 + \exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i^{(m)})} \left( \mathbf{Q}^{*\top} \boldsymbol{x}_{ij} \right)_k \right) - \delta_- \left( y_{ij} \left( \mathbf{Q}^{*\top} \boldsymbol{x}_{ij} \right)_k \right) \right] \\
& + \frac{1}{n_i} \sum_{j=1}^{n_i} \left[ \delta_+ \left( y_{ij} (\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i^{(m)}) \right) - \delta_- \left( \frac{\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i^{(m)})}{1 + \exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i^{(m)})} (\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i^{(m)}) \right) \right] \\
& + 2\lambda \left( (\mathbf{C}^{(m)} \mathbf{D})_i \right)_k + 2\lambda (\mathbf{C}^{(m)} \mathbf{W})_i^\top \boldsymbol{c}_i^{(m)} \Bigg\}.
\end{aligned} \tag{2.11}$$

Here, $\delta_+(\cdot)$ is a function defined as $\delta_+(x) := \max(x, 0)$ and $\delta_-(\cdot)$ is defined as $\delta_-(x) := \min(x, 0)$. Eq.(2.11) is derived from Eq.(2.9) and Eq.(2.10) which are the complementary

condition and the original constraint for the normalization of the membership vector. Therefore, Eq.(2.11) is a necessary condition to solve Eq.(2.6) and it is a stationary point. In addition, by introducing the $\delta$-functions, we ensure that the numerator and the denominator are both non-negative. Therefore, given any positive initial $\mathbf{C}^{(0)}$, the non-negativity of $\mathbf{C}^{(m)}$ is guaranteed.

Thus, we derive an algorithm that iteratively optimizes for $\mathbf{C}$ and $\mathbf{Q}$. The procedure of the proposed algorithm for learning LogCM and LogSCM is shown in Algorithm 1.

**Algorithm 1** The Learning Algorithm for LogCM and LogSCM

**Input:**

Data $\mathbf{X}_i$ and $\boldsymbol{y}_i$ for all $i = 1, \ldots, N$;

Initial values $\mathbf{Q}^{(0)}$ and $\mathbf{C}^{(0)}$;

Maximum iteration number $MaxIter$;

Similarity matrix $\mathbf{W}$; Tuning parameter $\lambda$;

**Output:**

$\mathbf{Q}^{(MaxIter+1)}$, $\mathbf{C}^{(MaxIter+1)}$.

1: **for** each $m \in [0, MaxIter]$ **do**
2: convert $\boldsymbol{x}_{ij}$ to $\widetilde{\boldsymbol{x}}_{ij}$ with current $\mathbf{C}^{(m)}$;
3: solve Eq.(2.8) and get $\boldsymbol{q}^{(m+1)}$;
4: transform $\boldsymbol{q}^{(m+1)}$ to $\mathbf{Q}^{(m+1)}$ by partitioning the $pK \times 1$ vector to the $p \times K$ matrix;
5: calculate $\mathbf{C}^{(m+1)}$ by applying Eq.(2.11).
6: **end for**

### *2.3.3 Empirical Guidelines for Implementing the Algorithm*

In concluding this section, we introduce a few important empirical guidelines for implementing the algorithm.

*Initialization of the parameters*

One important issue is the initialization of our two-step computational algorithm shown in Algorithm 1, i.e., to determine the initial values of canonical matrix $\mathbf{Q}$ and membership matrix $\mathbf{C}$. Suppose there are sufficient data from each individual to obtain a reliable estimation of the regression coefficients of the individuals. In that case, we can first construct individual models independently, with his/her own data. Then, with the estimated regression coefficients for every individual, we can employ clustering techniques such as the $k$-means method on the estimated regression coefficient vectors. The clustering algorithm will identify $K$ vectors as centers of the $K$ clusters, which will be our initial values of $\mathbf{Q}$. On the other hand, if data is limited, which is more likely in practice, we recommend using the mixed effects logistic regression model to obtain estimations of the regression coefficients of the individuals.

Next, the resemblances between the regression coefficient vector of an individual $i$ and the center vectors of the clusters can be calculated and further normalized to obtain the initial values of $\boldsymbol{c}_i$. In our work, inspired by the assumption of canonical structure, i.e., individual models are combinations of the canonical models, we will gain the initial membership matrix $\mathbf{C}$ by solving an optimization problem as following:

$$\begin{aligned} \min_{\boldsymbol{c}_i} \quad & \sum_i \|\boldsymbol{\beta}_i - \mathbf{Q}^{(0)}\boldsymbol{c}_i\|^2, \\ \text{s.t.} \quad & \boldsymbol{c}_i \geq 0, \quad \boldsymbol{c}_i^\top \mathbf{1} = 1, \quad i = 1, ..., N, \end{aligned} \tag{2.12}$$

where $\mathbf{Q}^{(0)}$ is the initial value of $\mathbf{Q}$, and $\boldsymbol{\beta}_i$ is the regression coefficient vector of individual $i$.

*Determination of K*

In some applications [77], we have prior knowledge to help us determine the number of canonical models, i.e., $K$. Without reliable prior knowledge, we could obtain the optimal $K$ using model selection methods such as AIC/BIC criteria along with the cross-validation technique. For instance, in our numerical studies which will be shown later in Section 2.4

and Section 2.5, we use 5-fold cross-validation to evaluate a range of $K$ (for instance, from 2 to 10) in terms of prediction accuracy using metrics such as error rate and the area under the Receiver Operating Characteristic (ROC) curve, i.e., the $AUC$ value. Specifically, the training set will be randomly divided into 5 sets. At each fold, 1 of them will serve as the validation set. We learn models based on the other 4 sets and report the prediction accuracy on the validation set. By comparing the average accuracy for each candidate $K$, we can determine which fits the training data best, in terms of prediction accuracy, i.e., lowest error rate or largest $AUC$.

*Acquisition of the similarity matrix* $\mathbf{W}$

In some applications, the similarity matrix $\mathbf{W}$ is already known through expert opinion or previous studies. We could also quantify the similarities between individuals using some personal characteristics such as demographic and social-economic factors, and other factors depending on the application contexts etc [122, 111]. To define similarity, existing approaches including 0-1 weighting, heat kernel weighting and dot-product weighting could be used. For instance, denote the personal characteristics of individual $i$ as $\boldsymbol{z}_i$, the heat kernel is defined as $w_{lm} = \exp(-\|\boldsymbol{z}_l - \boldsymbol{z}_m\|^2/\sigma^2)$ and the dot-product is $w_{lm} = \boldsymbol{z}_l^\top \boldsymbol{z}_m$. The 0-1 weighting is calculated in a way that for each individual, we treat its $k$ nearest neighbor as equally similar and assign the similarities as 1 for these nearest neighbors; for others, the similarities are 0. We recommend to use the heat kernel weighting in practice when $\mathbf{W}$ is not available because it creates a continuous similarity metric and has a tuning parameter $\sigma^2$ which can adapt to the data. On the other hand, even if personal characteristics are not available, we can obtain the similarities using a data-driven approach developed in [77]. The idea is to treat the estimated regression coefficients of the individuals as personal characteristics and calculate the similarities accordingly. In numerical studies, we will use mixed effects logistic model to obtain initial estimations of the regression coefficients of the individuals, and use heat kernel weighting to calculate the similarity matrix $\mathbf{W}$.

### *2.4 Simulation Studies*

To evaluate the model performance, we conduct extensive simulation studies in this section, and will show the performance in a real-world case of learning personal travel preferences, in the next section. We compare our proposed LogCM and LogSCM models with several benchmark methods including: 1) the one-size-fits-all logistic regression model (LR) that treats all individuals homogeneously, pooling all the individuals' data together to learn one population model; 2) the mixed effects logistic regression model (logistic MEM) which considers that the coefficients of individuals are sampled from a certain distribution; and 3) the independent logistic regression model (ILM) that learns the regression coefficients of each individual solely based on his/her own data.

In the simulation where the true parameters are known, we can evaluate the model performance by looking at the differences between the learned coefficients and the real ones. We can use the average Absolute Error defined as $\frac{1}{N}\sum_{i=1}^{N}|\boldsymbol{\beta}_i - \hat{\boldsymbol{\beta}}_i|$, and also the average correlations as $\frac{1}{N}\sum_{i=1}^{N}\rho(\boldsymbol{\beta}_i, \hat{\boldsymbol{\beta}}_i)$. A better model will lead to a smaller average absolute error value and a higher average correlation which reflect smaller gaps between the estimated and real values.

Besides, there are also several prediction accuracy metrics which can also reflect the model performance. As the outcomes are binary, we can evaluate the models using the Error Rate and the Receiver Operating Characteristics (ROC) curve. The ROC curve is a probability curve plotted with true positive rate ($TPR$, also known as sensitivity) against the false positive rate ($FPR$, equals to 1-specificity). We can further extract the area under the curve ($AUC$) from the ROC curve for each model. The larger the $AUC$ value is, the better the model is in terms of prediction accuracy. Since the evaluations for prediction accuracy (error rate and $AUC$ value) do not require knowing the true parameters, they can also be used in evaluating model performances in real-world cases. The experiments are conducted on R (version 3.4.2) on an Intel Core i5, 8 GB 2.40GHz PC, and the running times are reported in the results as well.

### *2.4.1 Design of the Simulation Experiments*

We conduct a comprehensive set of experiments to evaluate the performance of our proposed methods with benchmark methods in a variety of scenarios. We design the following guidelines to generate data. A detailed note for data generation is given in Appendix A.3.

With any given number of canonical models, for example $K = 3$, we manually set the parameters of the canonical models encoded in $\mathbf{Q}$ to make sure that they are different enough, as the canonical models are assumed to represent different preferences, behaviors or mechanism patterns. Then, for generating $\mathbf{C}$, the Dirichlet distribution is utilized to meet the constraints for membership vectors that $\boldsymbol{c}_i^\top \mathbf{1} = 1$ and $\boldsymbol{c}_i \geq 0$. To ensure the heterogeneity among individuals, we design three distinct Dirichlet distributions as: $F_1(\boldsymbol{c}) \sim Dir(\nu, 1, 1)$, $F_2(\boldsymbol{c}) \sim Dir(1, \nu, 1)$, and $F_3(\boldsymbol{c}) \sim Dir(1, 1, \nu)$ for $K = 3$, with a large tuning parameter $\nu = 20$. Each individual is first assigned randomly to one of the designed Dirichlet distributions, then the parameter vector can be obtained by $\boldsymbol{\beta}_i = \mathbf{Q}\boldsymbol{c}_i$. With this generation procedure, it is guaranteed that we can see the canonical structure in individual models. $\boldsymbol{x}_{ij}$'s are generated from multivariate normal distributions and $y_{ij}$'s are calculated accordingly with a small normal distributed noise. We consider one more layer of complexity, which is the balance of the two classes in the binary outcomes. For balanced data, the two labels are of similar sizes (50% for each), and for imbalanced data, the percentage of one label is designed to be around 80%.

We generate 40 data points for each individual and randomly pick 10 for testing. For the remaining 30 data points, two realistic scenarios are considered, i.e., dense sampling where data size $M \sim Unif(21, 30)$, and sparse sampling where $M \sim Unif(6, 12)$. Sparse sampling scenario is designed to refer to the application contexts in which there are only a few data points for each individual.

Table 2.1: Model performance comparison on testing set of the simulated balanced data

| | | | | Known Similarity | | Unknown Similarity | |
|---|---|---|---|---|---|---|---|
| | LR | MEM | ILM | LogCM | LogSCM | uLogCM | uLogSCM |
| **Dense Sampling** | | | | | | | |
| Time(sec) | **0.026** | 72.29 | 0.238 | 52.38 | 55.73 | 49.05 | 60.37 |
| Absolute Error | 4.769 | 5.578 | 4.311 | 1.684 | **1.589** | **1.684** | 1.707 |
| Correlation | 0.413 | 0.465 | 0.580 | 0.921 | **0.925** | 0.921 | **0.925** |
| Error Rate | 0.253 | 0.453 | 0.200 | **0.137** | **0.137** | **0.137** | **0.137** |
| AUC | 0.832 | 0.555 | 0.855 | 0.955 | **0.958** | **0.955** | 0.951 |
| **Sparse Sampling** | | | | | | | |
| Time (sec) | **0.025** | 32.12 | 0.213 | 82.57 | 43.72 | 79.42 | 37.39 |
| Absolute Error | 4.745 | 4.257 | 5.398 | 3.973 | **2.606** | 3.973 | **2.906** |
| Correlation | 0.411 | 0.506 | 0.434 | 0.557 | **0.758** | 0.557 | **0.721** |
| Error Rate | 0.240 | 0.257 | 0.213 | 0.130 | **0.110** | 0.130 | **0.117** |
| AUC | 0.749 | 0.624 | 0.865 | 0.947 | **0.949** | 0.947 | **0.954** |

### *2.4.2 Simulation Results*

In applying our methods on the simulated data, we test our methods under two scenarios: one scenario assumes that we have known the number of canonical models $K$ and the similarity matrix $\mathbf{W}$ by prior knowledge, and the other scenario assumes that we have no such knowledge so we have to use a data-driven approach to obtain both. As discussed in previous section, we use cross-validation to determine the $K$ for LogCM and LogSCM models, choosing the one has the highest average accuracy on validation sets, and derive the similarities between individuals based on the estimated coefficients by logistic MEM using heat kernel function. The cross-validation technique is also used in determining the tuning hyperparameter $\lambda$ for LogSCM.

Table 2.2: Model performance comparison on testing set of the imbalanced data

| | | | | Known Similarity | | Unknown Similarity | |
|---|---|---|---|---|---|---|---|
| | LR | MEM | ILM | LogCM | LogSCM | uLogCM | uLogSCM |
| **Dense Sampling** | | | | | | | |
| Time (sec) | **0.026** | 53.72 | 0.272 | 46.93 | 40.98 | 45.34 | 134.9 |
| Absolute Error | 6.835 | 4.004 | 4.120 | 1.969 | **1.626** | **1.969** | 2.853 |
| Correlation | 0.354 | 0.649 | 0.628 | 0.876 | **0.934** | **0.876** | 0.772 |
| Error Rate | 0.227 | 0.210 | 0.150 | **0.100** | 0.113 | 0.100 | **0.093** |
| AUC | 0.723 | 0.839 | 0.859 | 0.930 | **0.939** | **0.930** | 0.924 |
| **Sparse Sampling** | | | | | | | |
| Time (sec) | **0.022** | 24.28 | 0.208 | 4.208 | 44.75 | 4.032 | 20.55 |
| Absolute Error | 6.860 | 5.676 | 6.280 | 4.614 | **2.834** | 4.614 | **4.415** |
| Correlation | 0.355 | 0.414 | 0.455 | 0.656 | **0.764** | 0.656 | **0.679** |
| Error Rate | 0.233 | 0.210 | 0.243 | 0.143 | **0.113** | 0.143 | **0.140** |
| AUC | 0.763 | 0.784 | 0.780 | 0.866 | **0.914** | 0.866 | **0.900** |

Table 2.1 summarizes the results for balanced data with $K = 3$ and $p = 5$. We also conducted simulation experiments for other values such as $K = 5, 10$ and $p = 10, 50, 100$ and similar results could be observed. In Table 2.1, our LogCM and LogSCM model learned with estimated similarities are labeled as uLogCM and uLogSCM, respectively. While not shown in the table, by applying cross-validation technique based on average $AUC$, we can successfully identify $K = 3$ which is the ground truth number of canonical models. The results for imbalanced data of the same setting shown in Table 2.2. $K = 3$ can also be automatically learned.

We can observe from Table 2.1 and Table 2.2 as follows.

1) Overall, the LogSCM model outperforms the others, since it can exploit the canonical

structure of the individual models and the similarity information between individuals to enhance model estimation.

2) When data is more imbalanced and sparser, the advantage of LogSCM generally becomes larger, showing that a knowledge-driven model can overcome the lack of observations.

3) When the canonical structure is significant, i.e., $\nu$ is large, the proposed LogCM and LogSCM are better than other benchmark models in terms of both prediction accuracy and parameter learning, showing their efficacy in exploiting the canonical structure for better modeling.

4) The learned similarity matrix (in uLogSCM) can also help enhance the model estimation to some degrees, especially in sparse cases although may not as good as the real known information.

Furthermore, while it is not shown in the tables, it comes to us as a frequent observation that in some sparse sampling or imbalanced scenarios, the ILM and logistic MEM may not even be applied as the lack of data points result in highly ill-conditioned matrix operations that lead to immature breakdown of their computational algorithms.

## *2.5 Real-World Case Study*

In this section, we apply our proposed methods on a real-world data set collected from an innovative personalized transportation demand management (TDM) system introduced in [132]. This smart TDM system is a good example of the emerging smart apps that can interact with users and implement personalized incentives, showing promising capacity in coordinating and changing user behaviors.

### *2.5.1 Personalized TDM System*

TDM strategies are designed and widely deployed in most metropolitan regions to modify travel behavior patterns by providing incentives or costs to certain travel behaviors [4, 132, 84, 81, 65]. Examples include increasing parking costs in peak periods, offering low-cost public transportation, and monetary rewards, etc [14, 13, 89, 100, 99, 102, 7]. Traditionally, these TDM strategies are developed based on population level, and literature shows that those generic strategies have limited success [41, 110, 88]. Individuals will respond differently to these TDM strategies [49], given their diverse demographic and socio-economic characteristics [31]. Such diversity makes generic incentive ineffective, and thus, personalized incentive strategies are potentially more promising.

In recent years, the rapid proliferation of smart personal technologies makes it possible to interact with commuters and offer incentives individually [4, 132, 22]. For example, [132] proposed a new personalized TDM system which could offer personalized promotions to each commuter to change his/her behaviors. Ideally, the promotions with tailored rewards would be designed based on statistically accurate modeling of the user behavior, by learning from the interaction history between the commuter and the system. It has been reported in [132] that the personalized TDM system is quite effective in changing users' behaviors, i.e., the acceptance rate of the promoted suggestions reaches 68%, which leads to a significant travel time saving and congestion mitigation on the transportation system.

The smart TDM system works as follows. When a commuter is about to depart, he/she can request a trip in the app. Then a promoted alternative travel plan will be generated based on the app's knowledge of this commuter, i.e., based on the decision choice model that can be learned from previous interactions between the user and the app system. The alternative may differ from the user's original travel plan in some attributes, such as departure time and total travel time. To encourage the user to accept the promotion, a certain amount of reward points is assigned and the commuter will be awarded if he/she accepts the promotion and changes the travel plan accordingly.

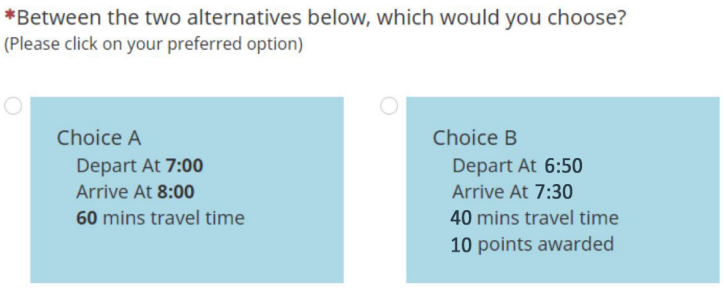


Figure 2.2: An example of the alternatives and the choice question.

Figure 2.2 shows an example of user scenario of this smart TDM system. Before the trip, the user will be asked to choose between Choice A (which is the original travel plan) and Choice B (a promotion). In this case, the attributes that characterize the choices include Schedule Delay Early ($SDE$), Schedule Delay Late ($SDL$), and Travel Time Saving ($TTS$). Besides, a certain amount of Reward Points ($Reward$) is given as well to encourage the commuter to change plan. The use of $SDE$ and $SDL$ as two variables is needed because it has been found that people have different preferences on departing earlier or later if they are asked to change their travel plan [13, 14].

The study investigated 1956 individuals about their preferences in daily commuting. Each respondent is interviewed for up to 13 rounds of the choice scenario like in Figure 2.2, where two alternatives were compared each time. Table 2.3 shows several rows of the data. The primary goal of this study is to learn personalized models of individuals' decisions in selecting among alternatives.

### *2.5.2 Results*

After a proper data cleaning, 828 individual's data [132] will be investigated in our study, and all of them answered all 13 questions. We assign the first ten rounds as training data and leave the last 3 as testing.

As the number of canonical models is unknown, we apply the same cross-validation

Table 2.3: Examples of the smart TDM system data

| Schedule Delay Early (min) | Schedule Delay Late (min) | Travel Time Saving (min) | Reward Points | Choice | Respondent ID | Question Number |
|---|---|---|---|---|---|---|
| 30 | 0 | 5 | 40 | B | 1 | 1 |
| 0 | 30 | 5 | 40 | A | 1 | 2 |
| 10 | 0 | 5 | 40 | B | 1 | 3 |
| ... | ... | ... | ... | ... | ... | ... |

procedure used in simulation studies to determine $K$ for LogCM and LogSCM. The left panel of Figure 2.3 shows the results of 5-fold cross-validation for determining $K$ for LogCM, ranging from 2 to 12. The black dot indicates the average $AUC$ on validation sets across all 5 folds and the error bar indicates the maximum and minimum. It can be observed from the left panel that when $K = 8$, the average $AUC$ on validation sets reaches the maximum and the variation across 5 folds is also smaller than the others. The similarities between individuals are derived based on the estimated coefficients given by logistic MEM as recommended. Then we also use the same cross-validation procedure to determine the tuning hyperparameter $\lambda$. The right panel of Figure 2.3 shows the results of LogSCM with $\lambda$ ranging from 0.018 ($e^{-4}$) to 7.389 ($e^{2}$). We can observe that when $\lambda = e^{-0.21} = 0.811$ the average $AUC$ reaches the maximum, although the differences are not significant compared with other values of $\lambda$.

The algorithms for both LogCM and LogSCM converge quickly. Figure 2.4 shows the evolutions of the values of the objective functions of LogCM (Eq.( 2.4)) and LogSCM (Eq.( 2.6)) over the iterations, on training set of the travel behavior preferences data. The left panel shows the convergence performance of LogCM, and the right panel shows it of LogSCM. The algorithms for both models converge quickly, in only 5 rounds (10 steps).

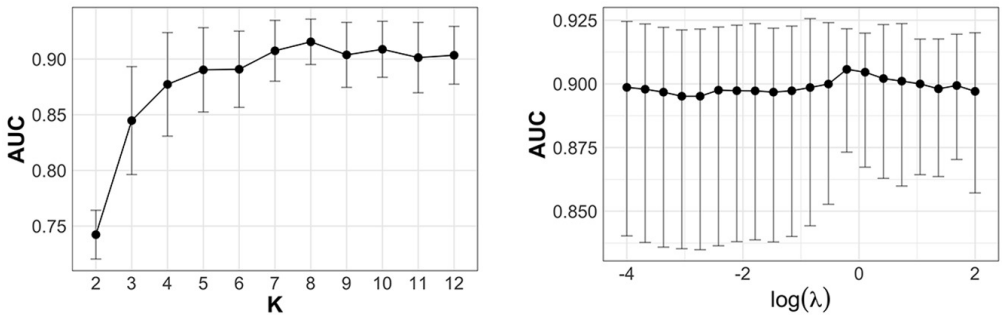


Figure 2.3: Determining the value for $K$ and $\lambda$ based on average $AUC$ on validation sets using 5-fold cross-validation technique, on the smart TDM system data.

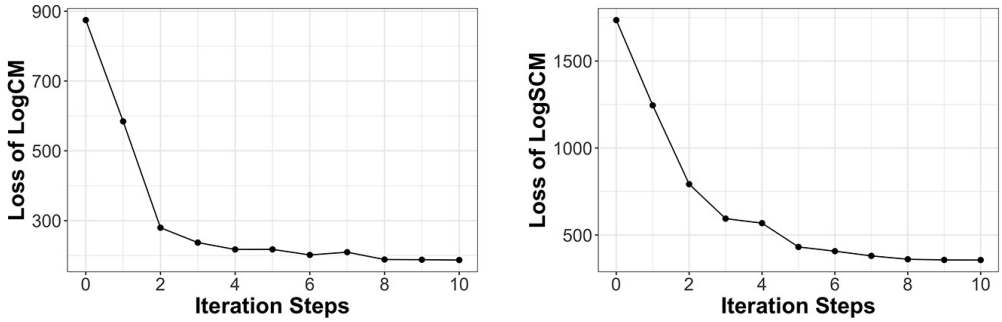


Figure 2.4: Convergence performances of the computational algorithms for LogCM and LogSCM, on the smart TDM system data.

We compare our proposed models with the benchmark models mentioned before: one-size-fits-all logistic regression (LR), mixed effects logistic regression (logistic MEM), and independent logistic regression (ILR). Unlike in simulation studies where we know the true values of the parameters of the individual models, here, as a real-world application, we do not have such knowledge. Thus, we focus only on prediction accuracy on the testing data. The error rate and $AUC$ value are adopted, as well as the Mean Squared Error ($MSE$). $MSE$ is quite popular in judging the goodness-of-fit of a model. Lower $MSE$ will indicate higher prediction accuracy of the model. When applied in logistic regression, it is calculated

Table 2.4: Model performance comparison on testing set of the smart TDM system data

| | LR | MEM | ILM | LogCM | LogSCM |
|---|---|---|---|---|---|
| Time (sec) | **0.034** | 900.5 | 1.658 | 75.67 | 762.7 |
| MSE | 0.195 | 0.269 | 0.151 | 0.126 | **0.125** |
| Error Rate | 0.289 | 0.268 | 0.157 | 0.154 | **0.150** |
| AUC | 0.672 | 0.742 | 0.836 | **0.852** | 0.851 |

based on the probabilities given by the learned model, i.e.,

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i}(y_{ij} - \pi_i(\boldsymbol{x}_{ij}))^2.$$

Table 2.4 summarizes the results. We can observe that the one-size-fits-all model (LR), has the worst prediction performance. It can also be observed that LogCM and LogSCM outperform others in terms of all three metrics. However, no significant difference between LogCM and LogSCM is observed here. It may come from the fact that the similarity information is learned from the data and the data for each individual is not large enough (only 10 questions). It is also consistent with what we observed in cross-validation procedure for $\lambda$, that the similarity information here may not improve the model performance a lot since different $\lambda$'s show quite similar performances on validation sets.

We further conduct another experiment to evaluate the effectiveness of the models, using different sample sizes. Although in this complete training set only 10 questions were collected from each individual, it is likely that in real-world implementation of the travel behavior intervention system, the number of observations of an individual may be even more insufficient, and sometimes may be imbalanced. To account for these complexities, we further randomly eliminate 30% (**dense sampling**, 70% remained) and 50% (**sparse sampling**, 50% remained) of the data from the training set. Less than 5 questions for each individual will make it impossible to run individual logistic regression with 4 variables and to run 5-fold cross-validation as well. We repeat the whole learning procedure (with cross-validation

Table 2.5: Model performance comparison on testing set of the sampled smart TDM system data

| | LR | MEM | ILM | LogCM | LogSCM |
|---|---|---|---|---|---|
| **Dense Sampling** | | | | | |
| MSE | 0.195 | 0.306 | 0.187 | 0.135 | **0.134** |
| Error Rate | 0.290 | 0.306 | 0.192 | 0.159 | **0.158** |
| AUC | 0.671 | 0.706 | 0.780 | **0.844** | 0.840 |
| **Sparse Sampling** | | | | | |
| MSE | 0.196 | 0.220 | 0.224 | 0.164 | **0.159** |
| Error Rate | 0.291 | 0.317 | 0.228 | 0.191 | **0.182** |
| AUC | 0.670 | 0.681 | 0.733 | 0.822 | **0.823** |

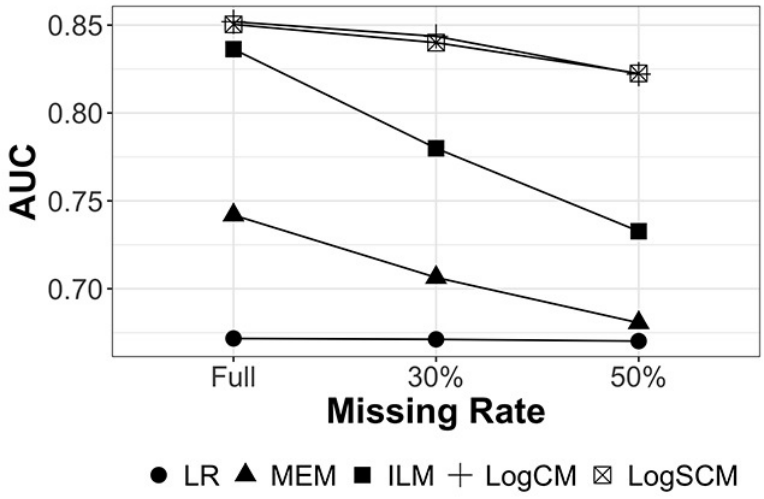


Figure 2.5: The change of $AUC$ of the models on testing set of the smart TDM system data, at different levels of missing rate of training set.

technique for determining $K$ and $\lambda$) on those new sparse data. Table 2.5 summarizes the results of these models, and the tendency can be seen in Fig 2.5.

It can be observed that:

1) The prediction performances of all models become worse when the training data becomes more sparse.

2) Our proposed models outperform the others in all three cases.

3) Figure 2.5 shows that the accuracy of ILM and MEM drops very fast, while the accuracy of LR remains stable but low. It is because LR uses all individuals' data so that it does not have the problem of limited data size, but it cannot capture heterogeneity. Meanwhile, our models show a high and stable capacity of learning under insufficient data.

4) LogSCM overall is slightly better than LogCM since it incorporates additional information in the data. This advantage is more noticeable when the data is very scarce, consistent with what we have observed in simulation studies.

As the similarity matrix is also learned from the data, we can expect an even better performance of LogSCM if valuable prior knowledge allows us to derive the similarity information. In all, the study shows that our proposed models are more powerful in personalized modeling and prediction. The advantage is even more obvious when learning with limited data.

### *2.5.3 Learning Behavior Patterns*

As the collaborative learning has a particular structure where all users share the same set of canonical models, the LogCM and LogSCM can provide explicit insight into some common behavioral preferences in the population by the explicit modeling of the canonical models.

For example, Figure 2.6 shows 3 examples of the canonical models (out of 8) learned by LogCM from the smart TDM dataset. The first canonical model ($\boldsymbol{q}_1 = [-0.246, 0.045, 0.320, 0.026]^\top$) shows one kind of pattern that the user does not want to depart home earlier while departing later is acceptable, and saving time on-road is very appealing. However, the second canonical model ($\boldsymbol{q}_2 = [-0.195, -0.499, 0.226, 0.005]^\top$) represents the pattern where the user

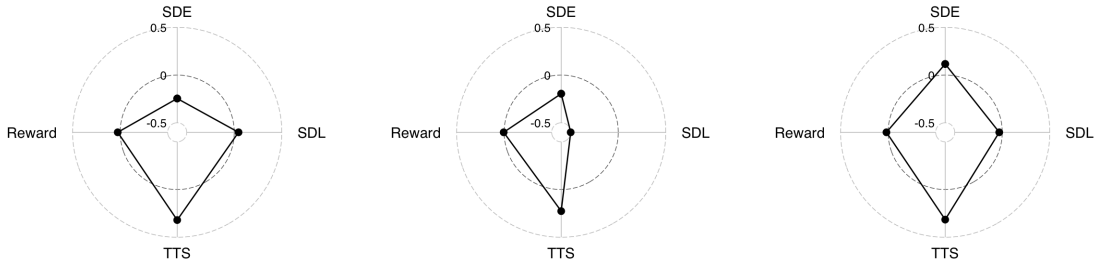


Figure 2.6: Three examples of the canonical models learned by LogCM, on the smart TDM system data.

does not want to change the departure time either way, but still wants to save time on-road. The third example ($\boldsymbol{q}_3 = [0.116, -0.003, 0.315, 0.018]^\top$) refers to another kind of user who may be willing to depart earlier.

These canonical models can refer to the different typical patterns in the heterogeneous population, or more precisely, the most representative or extreme ones among the behavior and preferences patterns, as each individual's model is a convex combination of the canonicals, making it a mixture of different typical patterns.

We can investigate the membership vectors to further understand the mixture of types for every user, and to grasp the canonical structure in the population. For instance, for User ID 596, the membership vector $\boldsymbol{c} = [0.981, 0.004, 0.003, 0.002, 0.002, 0.002, 0.002, 0.004]^\top$. The element for $\boldsymbol{q}_1$ being 0.981 indicates that this individual's behavior basically follows the first example shown in Figure 2.6, representing this typical pattern. Similarly, we can also find users whose behavior can be mostly explained by other canonical models as well, such as User ID 2364, whose element for $\boldsymbol{q}_2$ in membership vector is 0.964. However, these users are not the majority of the population, as the canonical models are the most "representative" or "extreme" patterns of user behaviors and user preferences. There are only 10 users out of the 828 investigated users have more than 0.95 for $\boldsymbol{q}_1$ in their membership vectors, meaning they have minimal influence from the other patterns.

Most individuals' preferences are more like in between of several canonicals, mixing mul-

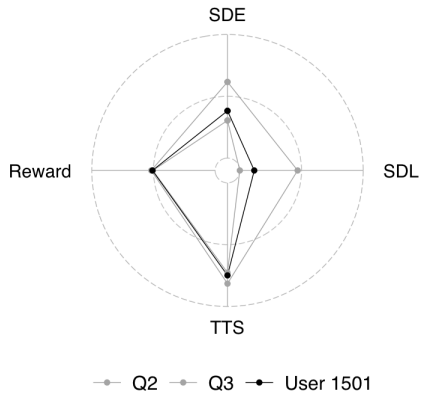


Figure 2.7: An example of the mixture of canonical models, User ID 1501.

tiple typical patterns. For example, User ID 1501, whose membership vector has 0.747 for $\boldsymbol{q}_2$, 0.245 for $\boldsymbol{q}_3$, and only 0.08 for other canonical models. His/Her preferences for departure earlier, later, time saving, and reward points are shown in Figure 2.7, where the black line represents this user's preferences $\boldsymbol{\beta} = 0.747\boldsymbol{q}_2 + 0.245\boldsymbol{q}_3 + \ldots$. We can tell that User 1501 is not as resistant to postponing departure as canonical model 2 ($\boldsymbol{q}_2$), but also does not like to departure earlier like canonical model 3 ($\boldsymbol{q}_3$). This user's preferences are somewhere in between, and closer to $\boldsymbol{q}_2$, based on the membership vector.

With this understanding of canonical models and membership vectors, we can further get insights into the heterogeneous structure of the population. We can define a $\boldsymbol{q}_1$-dominated group, whose behavioral patterns can be mostly explained by $\boldsymbol{q}_1$, and can also be influenced by the other canonical models, by filtering out the membership vectors with the first element being larger than 0.5. For example, in smart TDM data, we can find 130 users out of 828 investigated individuals as $\boldsymbol{q}_1$-dominated.

Such partition of the individuals can be further analyzed with other information of the users to help us understand the different patterns. For example, in the $\boldsymbol{q}_1$-dominated group, most individuals have flexible (33.85%) or very flexible (43.08%) arrival time requirements in the company, and only 10.77% users in this group have strict arrival time requirements.

In the whole population, those percentages for users with flexible and very flexible arrival times are 31.59% and 35.07%, while about 33.33% are having strict requirements. This may partially contribute to the reason that in typical pattern $\boldsymbol{q}_1$, departing later is acceptable.

In conclusion, we can see that the LogCM framework not only can provide better prediction accuracy for personalized modeling with limited data, but also has the potential for knowledge discovery and behavioral learning, in explaining personalized models, understanding typical patterns, and revealing the heterogeneous structure of the population.

## *2.6 Conclusion*

This chapter proposed a collaborative learning framework in learning user behavior (with binary outcomes) at the individual level, called LogCM. The LogCM framework addresses the challenge of the lack of observations for personalized modeling. It allows us to learn individual models from both individual's own data and the relationships with other individuals' data. Besides, LogCM does not assume a central tendency of the population, which is usually required in many other competing methods such as the mixed effects model. Thus, it has greater flexibility and can capture individual preferences better when considerable heterogeneity exists. We also provide a competent method for the LogCM framework, which achieves less computation time than mixed effects model. Extensive simulation studies and application on a new smart TDM system [132] show that the new method works well in learning individual-level models, even when data is very limited.

With its optimization formulation, LogCM can be easily extended with more information provided, and an intuitive example, LogSCM is presented. More delicate extensions for various circumstances will be provided in later chapters.

# Chapter 3

# EXTENSIONS OF LOGCM FOR SPECIAL CASES: UNEVEN CANONICAL STRUCTURE AND TIME-VARYING PREFERENCES

The Logistic Collaborative Model (LogCM) provides a framework for learning distinct individual classification models in a heterogeneous population. On top of its basic optimization formulation, additional information or structural design can be incorporated so that it can better fit in different, more complicated circumstances. For example, LogSCM in Chapter 2 utilizes the similarity information between individuals, and can further improve the prediction accuracy, especially when data is quite scarce.

In this chapter, we will propose two new models with the LogCM framework. The Pairwise-fusion LogCM (LogPCM) provides a comprehensive tool to discover uneven canonical structures. The LogCM with Time-varying preferences (LogCM-T) provides a time-relevant formulation to handle the challenges of preference shifting [50, 29, 9].

## *3.1 LogPCM: A Pairwise-Fusion Technique for Uneven Canonical Structure*

The LogCM and LogSCM models use cross-validation to select the number of canonical models, i.e., $K$. This implicitly assumes that the canonical structure is balanced and even. In other words, the subpopulations that correspond to the canonical models are of similar sizes. If this is not the case, there would be minority canonical models which only represent patterns in minor subpopulations, and they could be hard to be identified by cross-validation technique, since the increased model complexity (i.e., more canonical models) would not be justified by the minor gain in model performance.

To address such problem, our strategy is to reveal the whole structure of the population

heterogeneity, rather than identifying one best choice of $K$. This is inspired by the idea of path solution trajectory that has been used in sparse learning literature. We begin with a large $K$ and incorporate pairwise-fusion regularization [94, 80] to penalize the pairwise differences of candidate canonical models. Similar to LASSO or group LASSO [106] methods, with a proper penalty, small differences can be shrunk to zero, i.e., the canonical models with small differences will be fused into one. Thus, we name this method the Pairwise-fusion Logistic Collaborative Model (LogPCM). A visual tool will be introduced to show how LogPCM can help researchers get insights into the canonical structures.

### *3.1.1 Formulation of LogPCM*

Eq.(3.1) shows the formulation of LogPCM.

$$\begin{aligned} \min_{\mathbf{C},\mathbf{Q}} \quad & \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \left\{ \log(1+\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q} \boldsymbol{c}_i)) - y_{ij}(\boldsymbol{x}_{ij}^\top \mathbf{Q} \boldsymbol{c}_i) \right\} \\ & + \frac{\mu}{2} \sum_{k_1,k_2}^{K} \|\boldsymbol{q}_{k_1} - \boldsymbol{q}_{k_2}\|_2, \\ \text{s.t.} \quad & \boldsymbol{c}_i \geq 0,\ \boldsymbol{c}_i^\top \mathbf{1} = 1 \quad i = 1, ..., N. \end{aligned} \tag{3.1}$$

Here, $\boldsymbol{q}_{k_1}, \boldsymbol{q}_{k_2}$ are two candidate canonical models, i.e., two columns in matrix $\mathbf{Q}$, and $\mu \geq 0$ is the tuning parameter controlling the trade-off between prediction loss and the pairwise differences of the canonical models. The $L_2$-norm penalty exploits the non-differentiability at $\boldsymbol{q}_{k_1} - \boldsymbol{q}_{k_2} = 0$, setting the difference to be exactly 0 when it is small enough.

To implement LogPCM, we start at a large $K$ and a small $\mu$. Then gradually increase $\mu$ to see how the canonical models evolve, i.e., with the shrinkage effect of the $L_2$-norm, similar candidate canonical models will be fused into one. Eventually when $\mu$ is large enough, all canonical models will merge into one. Note the definition of $K$ here is slightly different from the $K$ in the original LogCM, that it is not the number of final canonical models, but rather an estimated upper bound.

### *3.1.2 Parameter Estimation Algorithm for LogPCM*

As Chapter 2 suggests, the $\mathbf{Q}$ step in original LogCM vectorizes the matrix $\mathbf{Q}$ and therefore simplifies the problem into a weighted logistic regression formulation which is convex optimization. Clearly, the additional $L_2$-norm is also convex with regard to canonical models $\boldsymbol{q}_k$ $(k = 1, \ldots, K)$. Thus, it is still convex with the vectorized $\boldsymbol{q}$. To see that, we define auxiliary matrices as:

$$\mathbf{A}_{(ij)} = [\mathbf{0}_p, \ldots, \mathbf{0}_p, \mathbf{I}_p, \mathbf{0}_p, \ldots, \mathbf{0}_p, -\mathbf{I}_p, \mathbf{0}_p, \ldots, \mathbf{0}_p]_{p \times Kp}, \tag{3.2}$$

where each $\mathbf{0}_p$ is a $p \times p$ square matrix with all entries being 0, and $\mathbf{I}_p$ is a $p \times p$ unit matrix. In total there are $K$ square matrices that make up an auxiliary matrix $\mathbf{A}_{(ij)}$, where the $i$-th square matrix is a unit matrix $\mathbf{I}_p$, the $j$-th is a negative unit matrix $-\mathbf{I}_p$, and all the others are zero matrices $\mathbf{0}_p$. We can see that for vectorized $\boldsymbol{q} = [\boldsymbol{q}_1^\top, \ldots, \boldsymbol{q}_K^\top]^\top$:

$$\mathbf{A}_{(ij)}\boldsymbol{q} = \boldsymbol{q}_i - \boldsymbol{q}_j. \tag{3.3}$$

With Eq.(2.8) for non-regularized $\boldsymbol{q}$ estimation, the modified $\mathbf{Q}$ step for LogPCM is also simplified to a non-constrained convex optimization as follows, and can also be solved by `CVXR` [37] package as well.

$$\begin{aligned} \min_{\boldsymbol{q}} \quad & \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \left\{ \log(1 + \exp(\widetilde{\boldsymbol{x}}_{ij}^\top \boldsymbol{q})) - y_{ij}(\widetilde{\boldsymbol{x}}_{ij}^\top \boldsymbol{q}) \right\} \\ & + \mu \sum_{k_1, k_2}^{K} \|\mathbf{A}_{(k_1 k_2)}\boldsymbol{q}\|_2. \end{aligned} \tag{3.4}$$

The $\mathbf{C}$ step for LogPCM remains the same, with the updating rule shown in Eq.(2.11), as there are no additional changes are made to the membership vectors.

### *3.1.3 Simulation Studies*

We run a simulation study similar as in Section 2.4, also based on $K = 3$ and $p = 5$. However, unlike the simulation studies in Section 2.4 where individuals are randomly assigned to 3

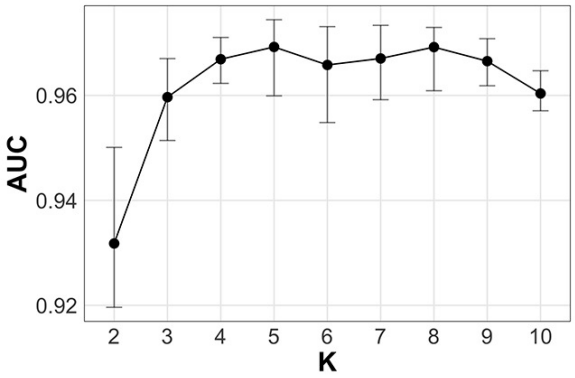


Figure 3.1: Determining the value for $K$ based on average $AUC$ on validation sets using 5-fold cross-validation technique, on the simulated data with one minority canonical model.

groups (each group mainly represents one of the canonical models), here, only about 10% of the generated individuals correspond to the canonical model #3, i.e., $P(F_3(\boldsymbol{c})) = 0.1$, which is called a minority canonical model.

First, we investigate the performance of the cross-validation technique for determining $K$ in this uneven canonical structure. Figure 3.1 shows that $K = 5$ is chosen based on the average $AUC$, which is not the ground truth. As a data-driven method, the cross-validation technique is designed to choose parameters which lead to highest prediction accuracy, but it may not reveal the real canonical structure.

If the primary goal is prediction, cross-valuation technique can lead to a reasonably good result, as indicated in Chapter 2. But it is less advantageous in knowledge discovery. Figure 3.2 shows the evolving plot of the candidate canonical models in LogPCM, when we tune the hyperparameter $\mu$ from small to large. It shows the relationships between the candidate canonical models during the fusing progress, such as which ones are fused together. Starting with $K_0 = 15$ (the initial $K$), when $\mu$ grows lager, more candidate canonical models will be fused together, and when $\mu$ is large enough, they will all merge together and become an one-size-fit-all model, i.e., $K = 1$. Here, we can see that $K = 3$ represents a distinct geometrical pattern that indicates the possibility that there are 3 unique canonical models.

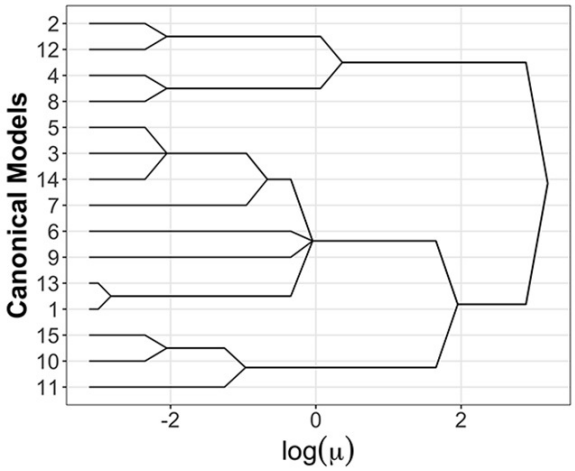


Figure 3.2: The fusion progress the canonical models with increasing $\mu$, on the simulated data, $K_0 = 15$.

Table 3.1: Parameter estimation comparison of the simulated data with one minority canonical model

| | LR | MEM | ILM | LogCM-5 | LogCM-3 |
|---|---|---|---|---|---|
| Absolute Error | 4.080 | 7.243 | 188.8 | 3.164 | **2.715** |
| Correlation | 0.849 | 0.747 | 0.612 | 0.928 | **0.960** |

Using $K = 3$ would lead to better model estimation, as shown in Table 3.1. Thus, the pairwise-fusion regularization could give us more insight into the dataset beyond what is provided by cross-validation technique. It does not mean that $K = 3$ is the only best answer. Actually, it is rather a flexible suggestion to determine $K$ and can be combined with domain knowledge. This revelation of the structure is important knowledge the LogPCM could provide for practitioners to make informed decisions.

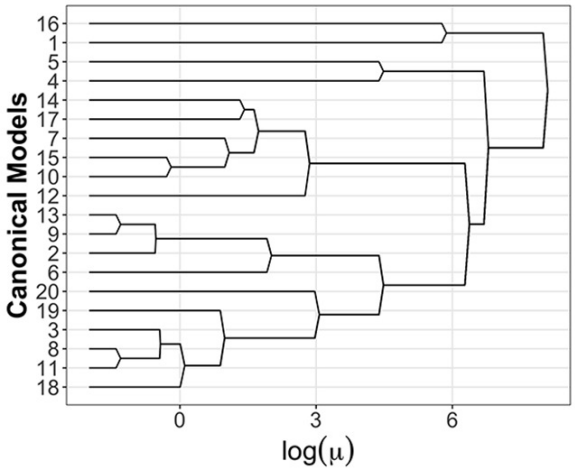


Figure 3.3: The fusion progress of the canonical models with increasing $\mu$, on the smart TDM system data, $K_0 = 20$.

Table 3.2: Model performance comparison on testing set of the smart TDM system data using LogCM with different $K$

| | $K = 8$ | 5 | 7 | 9 |
|---|---|---|---|---|
| MSE | 0.126 | **0.118** | 0.123 | 0.121 |
| Error Rate | 0.154 | 0.149 | 0.156 | **0.147** |
| AUC | 0.852 | **0.871** | 0.856 | 0.858 |

### *3.1.4 Real-World Case Study*

We also implement the LogPCM on the real-world travel behavior preferences dataset from the smart TDM system [132] that we used in Section 2.5. Remember that in Section 2.5, the cross-validation technique identified $K = 8$. Here, the evolving plot shown in Figure 3.3 reveals that it is quite likely that $K = 5(4.5 < \log(\mu) < 6)$. If we seek further refinement of the canonical structure, $K = 7$ $(3 < \log(\mu) < 4.5)$, $K = 9$ $(2 < \log(\mu) < 3)$ are also reasonable choices. As in this real-world case study, we don't know the ground truth of the dataset, we test all these choices of $K$ on testing set. Results are shown in Table 3.2.

From Table 3.2 we can observe that $K = 5$ outperforms $K = 8$ in terms of $AUC$ and $MSE$. $K = 9$ reaches the best result in error rate. The results indicate that the new LogPCM method provides a powerful tool in selecting $K$, which is flexible in practice because it shows the whole view of the canonical structure. Unlike the cross-validation technique which is entirely data-driven and determines one value of $K$ based on a given validation criteria (which is its strength, while is also its limitation), LogPCM reveals more information about the canonical structure and gives practitioners a good reference and the flexibility to make informed decisions.

### *3.1.5 Conclusion*

As an extension of the LogCM framework, the pairwise-fusion LogCM (LogPCM) holds the ability to learn distinct individual models when data is limited. Furthermore, it is a specialized approach to address the challenge of uneven canonical structure, where some minor subpopulations may not be adequately represented in the data and thereby could not be effectively discovered by automatic methods such as cross-validation. Numerical studies demonstrated the effectiveness of this model.

In addition, LogPCM provides an innovative knowledge discovery tool for better understanding or revealing the canonical structure, by incorporating pairwise difference regularization on the candidate canonical models. Inspired by the path solution trajectory which is commonly used in sparse learning, a graphical tool (canonical model evolving plot) is derived to show the full view of the structure, and grants LogPCM the flexibility for researchers to look for a proper $K$ meeting their needs.

## *3.2 LogCM-T: An Online Updating Method for Time-Varying Preference Learning*

Despite inter-individual heterogeneity, i.e., preferences between different individuals are different, intra-individual heterogeneity is sometimes noteworthy as well, for example, time-varying preferences. Many works have found that an individual's preferences may not be

constants but change along over time or evolve in the choice-making process [15, 67, 56, 71]. This will make the lack of personal data even more challenging.

LogCM handles well the challenge of limited data. However, in basic LogCM framework, each individual's model is a combination of the canonical models, which are stable logistic models, i.e., irrelevant to time. Therefore, unlike LogSCM or LogPCM where additional information is added directly into the objective function, we will need to extend the form of canonical models to incorporate time information. In this section, we will briefly introduce a co-authored work presented in [131], the Logistic Collaborative Model with Time-varying preferences (LogCM-T).

### *3.2.1 Formulation of LogCM-T*

Recall that in LogCM, the individual model can be expressed as:

$$\pi_i(\boldsymbol{x}_{ij}) = \Pr(y_{ij} = 1) = \frac{\exp(\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}_i)}{1 + \exp(\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}_i)} = \frac{\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}\boldsymbol{c}_i)}{1 + \exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}\boldsymbol{c}_i)}. \tag{3.5}$$

$\pi_i$ is the logistic regression model for individual $i$, where $y_{ij}$ is the $j$-th binary observation of this individual and $\boldsymbol{x}_{ij}$ is the $p$-length characteristic variables vector. Here, we assume that the parameter vector $\boldsymbol{\beta}_i$ and canonical model parameter matrix $\mathbf{Q}$ are both static over time. In LogCM-T, we will relax the assumption and have $\boldsymbol{\beta}_i = \boldsymbol{\beta}_i(t)$ as a function of time $t$ to express time-dependent variations, for every individual $i = 1, \ldots, N$. Note that here, the observations $\boldsymbol{x}_{ij}, y_{ij}$ will correspond to time, i.e. $\boldsymbol{x}_{ij}(t), y_{ij}(t)$, due to the time-dependency of $\boldsymbol{\beta}_i(t)$, so in this work, we will use subscript $t$ instead of the subscript $j$ to denote a specific observation of individual $i$, i.e., $\boldsymbol{x}_{it}, y_{it}$, in sake of notation simplicity. To be concrete, the individual models in LogCM-T can be expressed as:

$$\pi_i(\boldsymbol{x}_{it}) = \Pr(y_{it} = 1) = \frac{\exp(\boldsymbol{x}_{it}^\top \boldsymbol{\beta}_i(t))}{1 + \exp(\boldsymbol{x}_{it}^\top \boldsymbol{\beta}_i(t))}. \tag{3.6}$$

There are various parametric approaches to describe the time-varying preferences, for example, we use cubic polynomial functions with regard to time, to approximate every

element of $\boldsymbol{\beta}_i$, denote as $\beta_{ip}, p = 1, \ldots, P$. In other words, we consider the case that the preferences of every attributes can change over time. To be specific, for $p$-th attribute,

$$\beta_{ip}(t) = b_{ip,0} + b_{ip,1}t + b_{ip,2}t^2 + b_{ip,3}t^3 = \boldsymbol{b}_{ip}^\top \boldsymbol{v}_t,$$

where $\boldsymbol{v}_t = [1, t, t^2, t^3]^\top$ for cubic polynomial function, and $\boldsymbol{b}_{ip} = [b_{ip,0}, b_{ip,1}, b_{ip,2}, b_{ip,3}]^\top$ is the parameter vector to be estimated to approximate the change of preference $\beta_{ip}$ over time.

We then can apply collaborative modeling framework to the changing patterns of the preferences, i.e, $\boldsymbol{\beta}_i(t)$. For $p$-th attribute's preferences, we suppose there are $K_p$ canonical models, i.e., $K_p$ common changing patterns, whose parameter vectors can form a canonical matrix $\mathbf{Q}_p = [\boldsymbol{q}_{p,1}, \ldots, \boldsymbol{q}_{p,K_p}] \in \mathbb{R}^{4\times K_p}$ for the aforementioned cubic function. And for this attribute, each individual will have a membership vector $\boldsymbol{c}_{ip} \in \mathbb{R}^{K_p}$ which follows the same membership constraints as in LogCM (Eq.(2.4)), and we have $\boldsymbol{b}_{ip} = \sum_{k=1}^{Kp} c_{ip,k}\boldsymbol{q}_{p,k} = \mathbf{Q}_p\boldsymbol{c}_{ip}$. Hence the canonical structure for time-varying preferences can be expressed as $\beta_{ip}(t) = (\mathbf{Q}_p\boldsymbol{c}_{ip})^\top \boldsymbol{v}_t$, and the LogCM-T model is shown as follows,

$$\begin{aligned}
\min_{\boldsymbol{c}_{ip}, \mathbf{Q}_p, \forall i,p} \quad & \sum_{i=1}^{N} \frac{1}{n_i} \sum_{t=1}^{n_i} \left\{ \log\left(1 + \exp\left(\sum_{p=1}^{P} (x_{it,p}(\mathbf{Q}_p\boldsymbol{c}_{ip})^\top \boldsymbol{v}_t)\right)\right) - y_{it}\left(\sum_{p=1}^{P} (x_{it,p}(\mathbf{Q}_p\boldsymbol{c}_{ip})^\top \boldsymbol{v}_t)\right)\right\}, \\
\text{s.t.} \quad & \boldsymbol{c}_{ip} \geq 0,\ \boldsymbol{c}_{ip}^\top \mathbf{1} = 1 \quad i = 1, \ldots, N; p = 1, \ldots, P.
\end{aligned} \tag{3.7}$$

Further, if we assume that all the attributes share the same set of changing patterns, in other words, the time-varying preferences $\beta_{ip}$ of all attributes ($p = 1, \ldots, P$) share the same set of canonical models ($\mathbf{Q}_1 = \ldots = \mathbf{Q}_P = \mathbf{Q}$), we can simplify Eq.(3.7) into a similar matrix format as in LogCM (Eq.(2.4)) as follows,

$$\begin{aligned}
\min_{\mathbf{Q}, \mathbf{C}_i, \forall i} \quad & \sum_{i=1}^{N} \frac{1}{n_i} \sum_{t=1}^{n_i} \left\{ \log\left(1 + \exp(\boldsymbol{x}_{it}^\top ((\mathbf{Q}\mathbf{C}_i)^\top \boldsymbol{v}_t))\right) - y_{it}(\boldsymbol{x}_{it}^\top ((\mathbf{Q}\mathbf{C}_i)^\top \boldsymbol{v}_t))\right\}, \\
\text{s.t.} \quad & \boldsymbol{c}_{ip} \geq 0,\ \boldsymbol{c}_{ip}^\top \mathbf{1} = 1 \quad i = 1, \ldots, N; p = 1, \ldots, P.
\end{aligned} \tag{3.8}$$

where $\mathbf{Q}$ is the shared canonical model set, and $\mathbf{C}_i = [\boldsymbol{c}_{i1}, \ldots, \boldsymbol{c}_{iP}]$ is the membership matrix for individual $i$ whose columns are membership vectors for every attribute. Note that other

forms of time-dependency can also be used as well in Eqs.(3.7) and (3.8) by defining different $\boldsymbol{v}_t$'s. For example, in linear case, we can let $\boldsymbol{v}_t = [1, t]^\top$ and $\mathbf{Q}_p \in \mathbb{R}^{2\times K_p}$.

### *3.2.2 Parameter Estimation Algorithm for LogCM-T*

With the optimization problem shown in Eqs.(3.7) and (3.8), we can estimate the canonical parameters matrix $\mathbf{Q}$/matrices $\mathbf{Q}_p$ and the membership matrices $\mathbf{C}_i$ in alternation in the same algorithm framework as in Section 2.3. However, the computation process may take time. Thus, this wholesome update of all parameters of all individuals based on only a few observations from one individual is not efficient, and is not suitable for a time-sensitive model like LogCM-T. To update an individual's preferences when a new observation is available, we separate the process of canonical model updating and membership vector updating of the LogCM-T into two stages – "offline updating" which can be applied periodically to update the wholesome model and "online updating" which can perform real-time calculations to update the membership vector of an individual with new observations. We will propose the Online LogCM-T (OLCM) in this section, which focuses on the online stage of updating the membership vector only.

In the online updating stage, an individual's membership vectors will be updated given his/her own data, while the canonical models are not updated. The online updating can also be utilized to learn the membership vector when a new user comes to the system. Assume that for individual $i$, there are $n_i$ observations $(\boldsymbol{x}_{it}, y_{it}), t = 1, \ldots, n_i$ available now with new observations. In online updating process, we fix the canonical models and the optimization problem in Eq.(3.7) will become,

$$\begin{aligned} \min_{\boldsymbol{c}_{ip},\forall p} \quad & \sum_{t=1}^{n_i} \left\{ \log\left(1 + \exp\left(\sum_{p=1}^{P} (x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^\top \boldsymbol{v}_t)\right)\right) - y_{it}\left(\sum_{p=1}^{P} (x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^\top \boldsymbol{v}_t)\right)\right\}, \\ \text{s.t.} \quad & \boldsymbol{c}_{ip} \geq 0,\ \boldsymbol{c}_{ip}^\top \mathbf{1} = 1 \quad p = 1, \ldots, P. \end{aligned} \tag{3.9}$$

Here the decision variables are individual $i$'s $P$ membership vectors towards each dimension of the pareferences, i.e., each attributes. Similar as in Section 2.3, an updating rule for

every element can be derived on the basis of Karush-Kuhn-Tucker (KKT) conditions The detailed derivation process can be found in Appendix B.1. The final updating rule is shown as follows,

$$
\begin{aligned}
c_{ip,k}^{(m+1)} = c_{ip,k}^{(m)} \times \Bigg\{ \sum_{t=1}^{n_i} \Bigg[ & \delta_+ \left( \frac{\exp(\sum_{p=1}^{P} x_{it,p} (\mathbf{Q}_p \boldsymbol{c}_{ip}^{(m)})^\top \boldsymbol{v}_t)}{1 + \exp(\sum_{p=1}^{P} x_{it,p} (\mathbf{Q}_p \boldsymbol{c}_{ip}^{(m)})^\top \boldsymbol{v}_t)} \times x_{it,p} (\mathbf{Q}_p^\top \boldsymbol{v}_t)^\top \boldsymbol{c}_{ip}^{(m)} \right) \\
& - \delta_- \left( \frac{\exp(\sum_{p=1}^{P} x_{it,p} (\mathbf{Q}_p \boldsymbol{c}_{ip}^{(m)})^\top \boldsymbol{v}_t)}{1 + \exp(\sum_{p=1}^{P} x_{it,p} (\mathbf{Q}_p \boldsymbol{c}_{ip}^{(m)})^\top \boldsymbol{v}_t)} \times x_{it,p} (\mathbf{Q}_p^\top \boldsymbol{v}_t)_k \right) \\
& + \delta_+ \left( y_{it} \times x_{it,p} (\mathbf{Q}_p^\top \boldsymbol{v}_t)_k \right) - \delta_- \left( y_{it} \times x_{it,p} (\mathbf{Q}_p^\top \boldsymbol{v}_t)^\top \boldsymbol{c}_{ip}^{(m)} \right) \Bigg] \Bigg\} \Bigg/ \\
\Bigg\{ \sum_{t=1}^{n_i} \Bigg[ & \delta_+ \left( \frac{\exp(\sum_{p=1}^{P} x_{it,p} (\mathbf{Q}_p \boldsymbol{c}_{ip}^{(m)})^\top \boldsymbol{v}_t)}{1 + \exp(\sum_{p=1}^{P} x_{it,p} (\mathbf{Q}_p \boldsymbol{c}_{ip}^{(m)})^\top \boldsymbol{v}_t)} \times x_{it,p} (\mathbf{Q}_p^\top \boldsymbol{v}_t)_k \right) \\
& - \delta_- \left( \frac{\exp(\sum_{p=1}^{P} x_{it,p} (\mathbf{Q}_p \boldsymbol{c}_{ip}^{(m)})^\top \boldsymbol{v}_t)}{1 + \exp(\sum_{p=1}^{P} x_{it,p} (\mathbf{Q}_p \boldsymbol{c}_{ip}^{(m)})^\top \boldsymbol{v}_t)} \times x_{it,p} (\mathbf{Q}_p^\top \boldsymbol{v}_t)^\top \boldsymbol{c}_{ip}^{(m)} \right) \\
& + \delta_+ \left( y_{it} \times x_{it,p} (\mathbf{Q}_p^\top \boldsymbol{v}_t)^\top \boldsymbol{c}_{ip}^{(m)} \right) - \delta_- \left( y_{it} \times x_{it,p} (\mathbf{Q}_p^\top \boldsymbol{v}_t)_k \right) \Bigg] \Bigg\}
\end{aligned}
\tag{3.10}
$$

where the superscript $m$ refers to the order of iteration.

In summary, the membership vectors could be learned and updated iteratively with the following steps:

1. Input: $\mathbf{Q}_p$ for all $p = 1, \ldots, P$, $\boldsymbol{v}_t = [1, t, t^2, t^3]^\top$, $(\boldsymbol{x}_{it}, y_{it})$ for all $t = 1, \ldots, n_i$, initial value $\boldsymbol{c}_{ip}^{(0)}$, for all $p$;

2. For $m = 1, 2, ...$, iteratively update each dimension of each membership vector $c_{ip,k}^{(m+1)}$ with Eq.(3.10), given $\boldsymbol{c}_{ip}^{(m)}$ calculated in the previous iteration step;

3. Give a pre-determined criteria $\epsilon$. When $\gamma = \sum_{p=1}^{P} \| \boldsymbol{c}_{ip}^{(m+1)} - \boldsymbol{c}_{ip}^{(p)} \|_2^2 \leq \epsilon$, stop the iteration.

We could apply the proposed online updating method to the new data obtained by an individual and only update his/her membership vectors with much shorter computational

time. Since the canonical models are vital in the preferences learning process but they are not updated in the online updating stage, it is beneficial to apply the offline updating stage after once after a time. With such combination of the online and offline updating, we can achieve a balance between the requirements of real-time computing and accuracy of changing patterns. In the following sections, we will only consider the online updating stage, i.e., OLCM-T, as offline updating has been studies extensively in previous works (Chapter 2,[76, 77, 78]).

### *3.2.3 Simulation Studies*

We then conduct simulations to test the performance of the proposed model. The benchmark methods includes LR, ILM, MEM, same as in Section 2.4, and the original LogCM, where individual preferences are constant values $\boldsymbol{\beta}$ rather than time-varying variables $\boldsymbol{\beta}(t)$. Note that in LogCM, the canonical models represent different utility models in a format of $V = \boldsymbol{x}^\top \boldsymbol{\beta}$, while in LogCM-T, the canonical models represent the changing patterns of preferences $\boldsymbol{\beta}(t)$.

To mimic the online updating process over time, we assume that at each time step, one data point could be obtained from each individual. For a given time step, the data that could be used to learn the preferences of the individual is the set of data points obtained from all previous time steps, and the new data point collected at the current time step. In addition, we generate another 10 data points for the testing purpose only.

We start from a basic setting where (1) the total number of individual $N = 120$, (2) the number of attributes (i.e., preference dimension, preference parameters in utility function) $P = 4$, and (3) the number of canonical models for each preference dimension $K = 2$. With this basic setting, each individual has 4 membership vectors towards the 4 dimensions of his/her preferences, and each dimension could be described by the two canonical models of that preference dimension and his/her membership vector towards that dimension. We then generate simulated data sets in a similar way as described in Section 2.4.

The results of the prediction accuracy of each model at each time step are presented in Figure 3.4. Each curve in the figure is the average of 100 independent runs so that we can

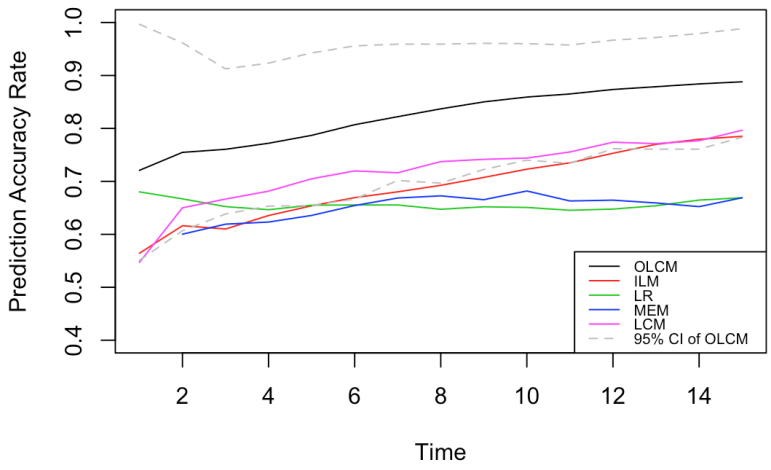


Figure 3.4: The prediction accuracy of each model over time in online updating process, on the simulated data.

report the 95% confidence interval (CI) as well. At time $t = 1$, each individual has 1 data point available for estimation. As time goes from $t = 1$ to $t = 15$, the data points collected from each individual increase from 1 to 15. In general, the prediction accuracy of all five models increases over time when more data are available. Among five models, OLCM-T considers time-varying preferences and has better performances at every time step.

Other settings of the simulation are also tested, such as different dimensions of preferences and different number of canonical models. Similar results can be observed as shown in Figure 3.5. The left panel shows the prediction accuracy of OLCM-T with different numbers of canonical models for each preference dimension, and the right panel presents the accuracy with different numbers of dimensions. Each curve is the average of 20 independent runs for more stable results.

### *3.2.4 Real-World Case Study*

Similarly, we implement the LogCM-T method and the online updating algorithm OLCM on the real-world travel behavior preferences dataset from the smart TDM system [132] that we used in Section 2.5. As we are testing the online updating, we separate the 828 respondents

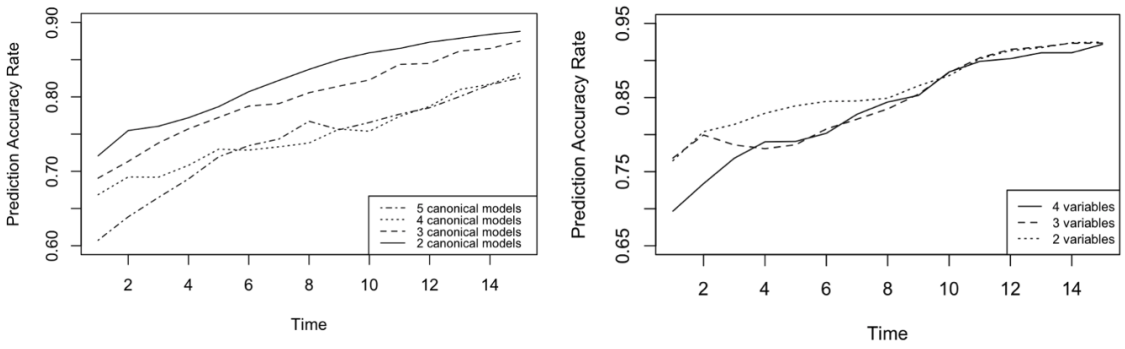


Figure 3.5: The prediction accuracy of OLCM-T on different settings of the simulated data.

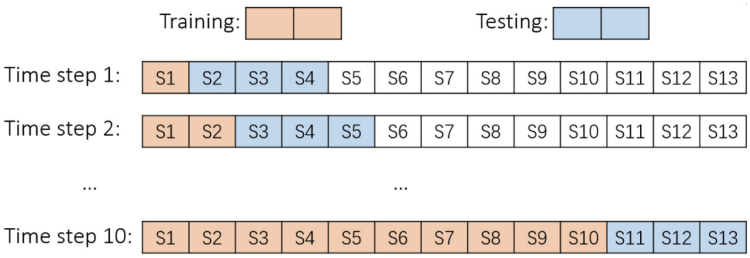


Figure 3.6: Illustration of the data points used for model training and testing with real-world dataset at each time step.

into two subsets. One subset contains randomly selected 80% of respondent, as the data set to learn the canonical models for the whole population, assuming that the canonicals are also applicable for the rest 20% respondents, which will be used to apply online updating algorithm.

Figure 3.6 illustrates the usage of data points in analysis. Similar as in simulation studies, at each time step, the data points obtained from all previous time steps and the new data point collected at this time step are used as training data. The following three scenarios and choices are used as the test data to evaluate the prediction accuracy, given the preferences learned at the current time step.

For OLCM-T, the membership vectors are updated at each time step with new data

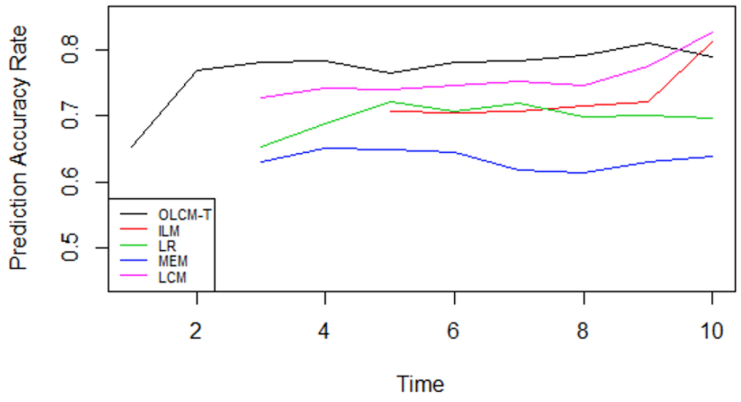


Figure 3.7: The prediction accuracy of different models, on the smart TDM system data.

points. The canonical models will not be updated, which are pre-trained based on the 80% subset of respondents. For other baseline models (LR,ILM,MEM,LogCM), the whole model will be updated with new data points at each time step.

From Figure 3.7, we could see that the proposed OLCM-T has higher prediction accuracy than other models from the first time step until the 10th time step, at which the prediction accuracy of LogCM and ILM exceeds that of OLCM-T. This might because while the canonical models of OLCM-T are fixed in the whole process, the original LogCM updates both canonical models and membership vectors at each time step, which may impact the performance of the model in later time steps. Furthermore, as time goes, more data are available so that LogCM and ILM may performance better than earlier time steps.

### *3.2.5 Conclusion*

We propose a time-varying extension of LogCM framework (LogCM-T) in the section, and investigate performance of the online updating strategy of the LogCM-T. We use polynomial functions to model the changing preferences, and in both simulation and real-case studies, the proposed method shows promising outcomes in individual preference learning and behavior prediction.

This work again shows that the LogCM framework can be easily extended to fit various real-world circumstances. Such flexibility not only lies in the ability to add regularization terms to the objective, it also lies in the fact that we can specify a more complex structure in base functions on top of logistic regression.

# Chapter 4

# CHCM: CONTEXTUAL HIERARCHICAL COLLABORATIVE MODELING FRAMEWORK FOR HETEROGENEOUS POPULATION WITH COMPLEX COMPOSITION

We propose Hierarchical Collaborative Model (HCM) in this chapter. It extends the collaborative structure into multiple layers and can learn different levels of canonical models, so that the commonalities under various degrees of details can be learned simultaneously. HCM has a flexible hierarchical structure, making it a better fit for the complex nature of large heterogeneity populations.

## *4.1 Introduction*

Collaborative models (CM) [75, 77] has been found very useful for individualized modeling where each individual has a distinct regression model. They can capture both the commonalities and inter-individual differences, even when individual data are limited due to its innovative low-rank representation of the individual models. As stated in previous chapters, collaborative learning framework postulates that in a heterogeneous population, the models of individuals are not independent, instead, they share one common set of canonical models which can represent some common patterns or typical types, and for each individual there is a unique membership vector representing the different degrees of resemblance of the individual model to the canonical models. In other words, an individual model can be represented as a combination of canonical models, and the weights are elements of the corresponding membership vector. Hence, population knowledge can help learn individual models and the common mechanism pattern in the population can be portrayed by the shared set of canonical models.

For a population with a large number of individuals and maybe a complex composition, however, the population heterogeneity is much more complicated, such as online shopping preferences and patient documents [18, 105, 39, 3]. The LogPCM in Chapter 3 also observed that in a real world case, there are multiple choices of canonical structure which lead to reasonable prediction accuracy and therefore can be seen as various depth of the canonical structure. Therefore, typical collaborative models are not ideal for such circumstances. For the first, it could be difficult to determine the number of canonical models. Too few canonical models will not be able to represent different kinds of common patterns, and results in information loss. On the other hand, too many canonical models will make each canonical model is not very common, and it will also increase the number of parameters-to-be-estimate and make the model less stable. Moreover, CMs' two-layer structure (one level of canonical models) could be an over-simplification of the population with complex heterogeneity. It may not be sufficient in depicting such complicated heterogeneity. In many real-world studies, population characteristics could be uneven, multi-level, and may also under different granularity for various circumstances. Thus, the single-level canonical models will lack such flexibility.

Like the extension of deep matrix factorization which can benefit from the deep structure to learn hierarchical attributes representations of a given dataset [127, 128, 118, 117], in this work, we develop the Hierarchical Collaborative Models (HCM) to effectively estimate distinct models for a large complex heterogeneous population. With HCM, the differences and the commonness of individuals may be captured by a hierarchical structure, and it can give us a potential to echo the complex composition in population and discover intrinsic heterogeneity structure.

The rest of the chapter is organized as follows. Section 4.2 formulates the proposed Hierarchical Collaborative Model (HCM) and explains the different constraint settings. Contextual HCM (CHCM) is also presented in this section to illustrate how population structure and model structure can be linked. Section 4.3 provides an alternating iterative algorithm to estimate the parameters. Numerical studies is given in Section 4.4 to demonstrate the

effectiveness of the new model. Conclusion and discussion are presented in Section 4.5.

## *4.2 The Contextual Hierarchical Collaborative Model (CHCM)*

We propose the innovative Hierarchical Collaborative Model (HCM) framework with the form of linear regression. To be concrete, for each individual $i, i = 1, \ldots, N$, we let the regression model of this individual $i$ be $g_i(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\beta}_i$, where $\boldsymbol{x}$ represents $P$ predictor variables, and $\boldsymbol{\beta}_i$ represents the corresponding regression parameters distinctly for this individual $i$. Later, we will also illustrate how individual characteristic variables can be utilized to construct the hierarchical canonical structure and present the Contextual HCM (CHCM).

### *4.2.1 Formulation of Hierarchical Collaborative Model*

Typical collaborative models assume that there is one set of canonical models in a heterogeneous population, denoted as $f_k(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{q}_k, k = 1, \ldots, K$, where $\boldsymbol{q}_k$ is the corresponding regression parameter vector. The number of canonical models is usually much smaller than the number of individuals (i.e., $K \ll N$), granting the advantage of collaborative learning to reduce the burden of estimating a large number of free parameters so that it can learn distinct individual models when data is limited.

With the the canonical models, each individual's model could be characterized as an integration of them, and we assign a membership vector $\boldsymbol{c}_i$ to each individual $i,\ i = 1, \ldots, N$, to represent the degrees of resemblance of the individual model to the canonical models. Since each canonical model describes one kind of mechanism pattern in the population, by integrating this set of canonical models, the individual models $g_i(\boldsymbol{x}) = \sum_k c_{ik} f_k(\boldsymbol{x}), i = 1, \ldots, N$, can provide an adequate characterization of the individuals. Specifically, in the models with individual parameters $\boldsymbol{\beta}_i, i = 1, \ldots, N$, the collaborative structure can be expressed as $\boldsymbol{\beta}_i = \sum_k c_{ik} \boldsymbol{q}_k = \mathbf{Q}\boldsymbol{c}_i$, or in matrix format:

$$\mathbf{B} = \mathbf{Q}\mathbf{C}, \tag{4.1}$$

where $\mathbf{B} = [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N] \in \mathbb{R}^{P \times N}$ contains the parameter vectors of all individual models, $\mathbf{Q} =$

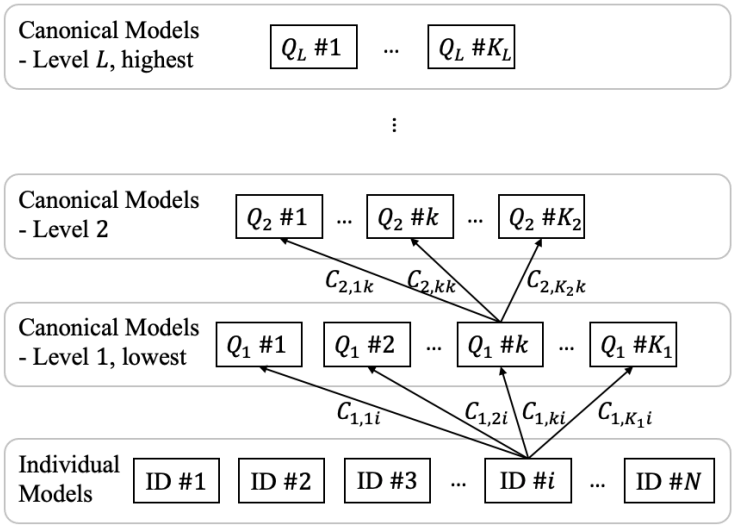


Figure 4.1: Schematic of the hierarchical collaborative model.

$[\boldsymbol{q}_1, \ldots, \boldsymbol{q}_K] \in \mathbb{R}^{P\times K}$ contains all the canonical model parameters, and $\mathbf{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N] \in \mathbb{R}^{K\times N}$ is the matrix with all membership vectors. An exemplary illustration of typical collaborative models can be found in Figure 2.1 in Chapter 2.

We notice in CM formulation, one set of canonical models will be found in a data-driven manner, and every canonical model is of a similar status level. Therefore, for a large population with complex composition, we may want to extend the framework of CM to a multi-layer structure so that canonical models of different granularity can be learned. For example, in the smart TDM system data described in Section 2.5, users with different commute times may have distinct common patterns, which can be modeled as one level of canonical models. Furthermore, even within those with similar commute times, user preferences will vary because of their different departure time or company requirements. It may lead to another set of canonical models, which are more detailed than the previous set. Figure 4.1 illustrates the structure of the Hierarchical Collaborative Model (HCM).

In HCM, we assume there exists multiple layers of the collaborative structure, i.e., there are in total $L$ levels of canonical models instead of one. Between every two consecutive layers,

we assign "membership vectors" for lower-level canonical models (or individual models) so that they are combinations of the canonical models of the higher level. To be concrete, the canonical-membership structure similar to Eq.(4.1) is embedded in every consecutive layers in HCM, i.e.,

$$\mathbf{Q}_l = \mathbf{Q}_{l+1}\mathbf{C}_{l+1}, \tag{4.2}$$

where $\mathbf{Q}_l$ and $\mathbf{Q}_{l+1}$ are two different levels of canonical models, and $\mathbf{Q}_{l+1}$ is one level higher than $\mathbf{Q}_l$. As defined, Eq.(4.2) has the exact same structure as in Eq.(4.1). It means that between these two layers, $\mathbf{Q}_{l+1}$ are the canonical models for $\mathbf{Q}_l$, and $\mathbf{C}_{l+1}$ are the corresponding membership vectors. Higher level canonical models represent higher-level common patterns. They are more "rough" expressions of the the most common patterns, e.g., the common patterns for respondents with different commute times. While lower level canonical models are more "detailed" expressions of the subdivision of common patterns for individuals, and can be considered as the details of higher level canonical models, such as further breakdown the commute time canonical models based on departure time.

With Eq.(4.2) for all $l = 1, \ldots, L-1$, we can develop the chain rule for factorizing the individual model parameter matrix as,

$$\mathbf{B} = \mathbf{Q}_L\mathbf{C}_L \cdots \mathbf{C}_2\mathbf{C}_1, \tag{4.3}$$

and derive the optimization formulation for HCM with $L$ levels of canonical models as following:

$$\begin{aligned}
\min_{\mathbf{Q}_L, \mathbf{C}_l, \forall l} \quad & \sum_{i=1}^{N} \|\boldsymbol{y}_i - \mathbf{X}_i\mathbf{Q}_L\mathbf{C}_L \cdots \mathbf{C}_2\mathbf{C}_{1,\cdot i}\|^2 \\
\text{s.t.} \quad & (A) \quad \mathbf{C}_{1,\cdot i} \geq 0,\ \sum_{k=1}^{K_1} \mathbf{C}_{1,ki} = 1,\ \forall i, \\
& (B) \quad \mathbf{C}_{l,\cdot i} \geq 0,\ \sum_{k=1}^{K_l} \mathbf{C}_{l,ki} = 1,\ \forall i, l, \\
& (C) \quad \mathbf{C}_l \cdots \mathbf{C}_2\mathbf{C}_{1,\cdot i} \geq 0, \\
& \qquad \sum_{k=1}^{K_l} (\mathbf{C}_l \cdots \mathbf{C}_2\mathbf{C}_{1,\cdot i})_k = 1,\ \forall i, l.
\end{aligned} \tag{4.4}$$

Table 4.1: Notation system in Hierarchical Collaborative Model (HCM)

| Category | Notation | Meaning | Dimension |
|---|---|---|---|
| Data | $\boldsymbol{y}_i$ | outcome of individual $i$ | $n_i \times 1$ |
| | $\mathbf{X}_i$ | predictor variables of individual $i$ | $n_i \times P$ |
| Parameter | $\mathbf{Q}_L$ | top level (level $L$) canonical matrix, each column is one cononical | $P \times K_L$ |
| | $\mathbf{C}_L$ | link matrix between level $L$ and $L-1$ canonicals | $K_L \times K_{L-1}$ |
| | $\vdots$ | $\vdots$ | $\vdots$ |
| | $\mathbf{C}_l$ | link matrix between level $l$ and $l-1$ canonicals | $K_l \times K_{l-1}$ |
| | $\vdots$ | $\vdots$ | $\vdots$ |
| | $\mathbf{C}_1$ | membership matrix between $1st$ level canonicals and individual models | $K_1 \times N$ |
| Hyper-parameter | $N$ | number of individuals, $i = 1, \ldots, N$ | scale |
| | $n_i$ | number of data points of individual $i$ | scale |
| | $P$ | number of predictor variables, $p = 1, \ldots, P$ | scale |
| | $L$ | number of levels, $l = 1, \ldots, L$ | scale |
| | $\boldsymbol{K}$ | number of canonical models of every level, $\boldsymbol{K} = [K_1, \cdots, K_L]^\top$, $N > K_1 > \ldots > K_L$ | $L \times 1$ |
| Derived natation | $\widetilde{\mathbf{Q}}_l$ | $= \mathbf{Q}_l = \mathbf{Q}_L\mathbf{C}_L \cdots \mathbf{C}_{l+1} = \mathbf{Q}_{l+1}\mathbf{C}_{l+1}$, level $l$ canonical matrix | $P \times K_l$ |
| | $\widetilde{\mathbf{C}}_l$ | $= \mathbf{C}_l\mathbf{C}_{l-1} \cdots \mathbf{C}_1$, the membership matrix from level $l$ canonicals to individual models; note $\widetilde{\mathbf{C}}_1 = \mathbf{C}_1$ | $K_l \times N$ |
| | $\mathbf{B}$ | $= \mathbf{Q}_L\mathbf{C}_L \cdots \mathbf{C}_1 = \widetilde{\mathbf{Q}}_l\widetilde{\mathbf{C}}_l$ for any $l = 1, \ldots, L$, individual model matrix, each column is $\boldsymbol{\beta}_i$ | $P \times N$ |
| | $\widetilde{\mathbf{X}}_{l,i}$ | $= \mathbf{X}_i\widetilde{\mathbf{Q}}_l$, can be considered as the predictions using level $l$ canonical models for data of individual $i$ | $n_i \times K_l$ |
| | $\widetilde{\boldsymbol{c}}_{l,i}$ | $= \widetilde{\mathbf{C}}_{l,\cdot i}$, $i$-th column of $\widetilde{\mathbf{C}}_l$, the membership vector at level $l$ for individual $i$ | $K_l \times 1$ |

For future clarity, we summarize the notation system for HCM in Table 4.1. A detailed explanation of the HCM formulation as shown in Eq.(4.4) is in the next subsection.

### *4.2.2 Explanation of HCM Formulation*

The hierarchical collaborative model has a very elegant structure which makes it easy to interpret and also very versatile to respond to different needs. The flexibility of HCM is

entitled by its chain structure in Eq.(4.3), and the different constraints.

For HCM with $L$ levels, the objective of the Eq.(4.4) is the total sum of least square loss of all the individual-level regression models, where:

$$\boldsymbol{\beta}_i = \mathbf{Q}_L \mathbf{C}_L \cdots \mathbf{C}_2 \mathbf{C}_{1,\cdot i}. \tag{4.5}$$

We know that level $l$ canonical models can be calculated as $\widetilde{\mathbf{Q}}_l = \mathbf{Q}_L \mathbf{C}_L \ldots \mathbf{C}_{l+1}$ by definition, and we can denote $\widetilde{\mathbf{C}}_l = \mathbf{C}_l \mathbf{C}_{l-1} \cdots \mathbf{C}_1$ and $\widetilde{\boldsymbol{c}}_{l,i} = \mathbf{C}_l \mathbf{C}_{l-1} \cdots \mathbf{C}_2 \mathbf{C}_{1,\cdot i}$ as the $i$-th column of $\widetilde{\mathbf{C}}_l$. Therefore, individual model $\boldsymbol{\beta}_i$ can be expressed by canonical models at any level $l$ as:

$$\boldsymbol{\beta}_i = \widetilde{\mathbf{Q}}_l \widetilde{\boldsymbol{c}}_{l,i}, \tag{4.6}$$

where $\widetilde{\mathbf{Q}}_l$ is the canonical model matrix at level $l$. This form will be the same as canonical structure, if $\widetilde{\boldsymbol{c}}_{l,i}$ follows certain constraints.

Similar to collaborative models, Constraint $(A)$ in Eq.(4.4) is called the "membership constraints" which comes from the definition of membership vectors. It works on $\mathbf{C}_1$ in HCM, which connects the lowest level canonical models and the individualized models. Specifically, we have $\boldsymbol{\beta}_i = \widetilde{\mathbf{Q}}_1 \widetilde{\boldsymbol{c}}_{1,i}$, where $\widetilde{\boldsymbol{c}}_{1,i}$ is non-negative and the sum of the elements is 1. It indicates that at level 1 (the lowest level), HCM can be interpreted as a single-level CM, where each individual is a combination of the canonical models. This constraint is required for all collaborative modeling framework and can assure the model identifiability.

Besides the membership constraints, unlike CMs, HCM has more than one level of canonical models; hence it has more than one level of membership vectors as well. Constraint $(B)$ and constraint $(C)$ offer two different strategies in extending membership constraints. Constraint $(B)$ has the same requirements as constraint $(A)$ except that it works on all other $\mathbf{C}$ matrices $(\mathbf{C}_2, \ldots, \mathbf{C}_L)$. It focuses on the collaborative structure between consecutive layers. It requires every level to strictly follow membership constraints, meaning that for two consecutive levels, the higher level canonical models are canonical models for the lower level. As these $\mathbf{C}$ matrices ($l \geq 2$) links different levels of the canonical models, we will name them "link matrices" to distinguish from membership matrix $\mathbf{C}_1$. Therefore, constraint $(B)$ can

also be called "link constraint", which is exactly the same as membership constraint, but applied on link matrices.

Constraint $(C)$ works on derived membership matrices $\widetilde{\mathbf{C}}_l$. It emphasizes more on the connections between different levels of canonical models ($\widetilde{\mathbf{Q}}_l$) and individual models. In other words, it ensures that any $\widetilde{\boldsymbol{c}}_{l,i}$ in Eq (4.6) will follow the membership constraints, i.e., $\boldsymbol{\beta}_i = \widetilde{\mathbf{Q}}_l \widetilde{\boldsymbol{c}}_{l,i}$ now can represent the collaborative structure on every level $l$. We name it as "general membership constraint". Hence, with constraints $(A)$ and $(C)$, HCM can be considered as a single-level CM for any given level $l$, so that it has the flexibility to interpret the individual models and common patterns under different granularity.

Moreover, the following Lemma 4.2 reveals the relation between link constraints (constraint $(B)$) and general membership constraints (constraint $(C)$), and will add another aspect of versatility of HCM.

**Lemma 4.2:** *For two non-negative matrices $\mathbf{A}_1 \in \mathbb{R}^{p\times q}$ and $\mathbf{A}_2 \in \mathbb{R}^{q\times r}$, whose column sums are all 1, the product $\mathbf{A}_1\mathbf{A}_2$ is also a non-negative matrix with all column sums being 1.*

The proof is provided in Appendix C.1. Lemma 4.2 indicates that constraint $(B)$ is stricter than constraint $(C)$, i.e., when link constraints and membership constraints are satisfied, the general membership constraints are always satisfied. In addition, when constraint $(B)$ is applied in HCM, Lemma 4.2 will assure that there is collaborative structure between any two levels of canonical models. That is, for any two levels of canonical models, $\mathbf{Q}_{l_1}, \mathbf{Q}_{l_2}$, $(l_1 > l_2)$, $\mathbf{Q}_{l_1}$ can be formulated as canonical models for $\mathbf{Q}_{l_2}$, because $\mathbf{Q}_{l_2} = \mathbf{Q}_{l_1} \prod_{l1 \geq l > l_2} \mathbf{C}_l$, and $\prod_{l1 \geq l > l_2} \mathbf{C}_l$ meets the requirements of link constraint, i.e., membership constraint. Equivalently, by multiplying any number of consecutive link matrices, the product matrix will also satisfy the membership constraint, and can be served as a new link matrix with hierarchical structure preserved in the collapsed new model. This add another aspect of flexibility to HCM models.

Finally, we briefly discuss the relationship between HCM and semi-nonnegative matrix factorization algorithms (Semi-NMF). On the surface, the multi-layer structure in HCM looks

very alike matrix factorization works, especially Semi-NMF [117, 2, 79, 30], as we constrain the membership matrix as a non-negative matrix. However, there are several key differences.

Most importantly, the MF algorithms focus on the decomposition of a known multivariate data matrix. The optimization objective is usually the differences between the actual matrix and the estimated low-rank representation, using matrix norms such as the Frobenius norm. However, in our model, although the collaborative structure of the individual model matrix $\mathbf{B}$ is a multi-layer matrix factorization, we do not have the knowledge of $\mathbf{B}$. On the contrary, the individual models are what the HCM seeks to estimate. Therefore, the MF algorithms cannot be readily applied to HCM since they all utilize the information in the matrix $\mathbf{B}$. The loss function of HCM is also different from the MF algorithms due to the same reason. It applies the least square error of the predictions, which is more similar to the regression models.

Other than that, the deep semi-NMF [117] and HCM are slightly different in the constraints of the factor matrices. Deep semi-NMF factorizes the data matrix into a series of mixed-signed matrices and a soft membership right matrix which is non-negative, while in our model, the non-negative constraints are often applied to all the link matrices as well as the membership matrix. It will lead to slightly different understanding of the model, that in our model, the different level of canonical models are also linked by membership vectors which enable the model to explain the preferences heterogeneity in population with various depths of details.

### *4.2.3 Characteristic Information and Contextual HCM*

In the HCM framework, we construct a hierarchical structure to learn various levels of canonical models, i.e., common patterns under different granularities in heterogeneous populations, and ultimately obtain distinct individual models for each individual. Such a hierarchical structure can echo the complex composition of the population, so that the model learning may be further enhanced with some characteristic information of individuals, such as demographics like gender and age, or the different commuting routines in the example of the

smart TDM system [132]. Note that characteristic variables are different from the predictor variables $\boldsymbol{x}$ we use in HCM.

To reflect the composition of population, we introduce Contextual Hierarchical Collaborative Model (CHCM), where we specify the hierarchical structure for HCM based on characteristic variables in the data. For instance, we consider a dataset with 3 characteristic variables, and the characteristic variables have 4,3,2 labels, respectively. In other words, the population can be divided into 4 groups based on the first characteristic, into 3 groups based on the second characteristic, and into 2 groups based on the third. Therefore, we can specify a hierarchical collaborative model with 3 levels of canonical models, where in each level, we specify the number of canonical models as $4, 4 \times 3, 4 \times 3 \times 2$, respectively. We can use $\boldsymbol{K}$ to represent the structure of HCM, e.g. here $\boldsymbol{K} = [24, 12, 4]^\top$.

The Contextual HCM (CHCM) provides an intuitive approach to specify the model structure of HCM. Besides, as the model structure given by CHCM matches the population composition (in terms of those characteristics), CHCM has a good interpretation, where each canonical model could correspond to a common pattern in a specific subgroup, defined by characteristic variables. For example, in the previous example, the first characteristic variable can divide the population into four different groups, and the top level of CHCM will also have four canonical models. Each canonical model at the top level can represent the most typical pattern in one group. Each individual model is a combination of such different typical patterns, while the weights would possibly be related to his/her characteristic labels.

In summary, HCM and CHCM benefit heterogeneous population learning in multiple ways:

- HCM preserves the advantages of CMs, comparing with population-level model (one-size-fits-all), independent individual-level models (independent regression) or mixed-effect models, as HCM covers the basic structure of CM, as shown in Eq.(4.6). It can learn distinct individual models by exploiting both individual's own data and the relationships with other individuals' data, just as single-level CMs.

- The hierarchical structure can reveal the heterogeneity in population with different levels of details by deploying multiple level of canonical models. Especially in CHCM, where the hierarchical structure echos the population composition, the canonical models have a good interpretation which matches the subpopulations with different characteristics.

- The utilization of characteristic information in CHCM grants the possibility to have a good guess for the individual model for a brand new individual, as we can always find the canonical model of the smallest group he/she belongs to by his/her characteristic variables.

- There are collaborative structures between two levels of canonical models as well in HCM. Thus, the collaborative structure not only can embed the heterogeneity of the population, but also can reveal a hidden structure between subpopulations or in the entire population.

### *4.3 Parameter Estimation Algorithm*

The formulation of HCM shown in Eq.(4.4) is not easy to solve directly. In this section, we present an alternating algorithm with "back propagation" [20, 51] fashion to solve this model with several easier sub-problems. Similar strategy has been exploited in collaborative learning framework [76, 77, 78] where $\mathbf{Q}$ and $\mathbf{C}$ are learnt alternatively and iteratively until model converges.

At each round of iteration for solving HCM with $L$ levels, we will update the estimations starting from level 1 membership matrix $\mathbf{C}_1$, through all the link matrices (from $\mathbf{C}_2$ to $\mathbf{C}_L$) and to the level $L$ canonical matrix (highest level $\mathbf{Q}_L$). In total, there are $L+1$ steps in each round.

### *4.3.1 Estimation Step for Membership Matrix* $\mathbf{C}_1$ *(*$\mathbf{C}$ *Step)*

In this step, we focus on solving $\mathbf{C}_1$ with a given $\widetilde{\mathbf{Q}}_1 = \mathbf{Q}_L\mathbf{C}_L \cdots \mathbf{C}_2$, i.e., $\widetilde{\mathbf{Q}}_1$ could be the latest estimation of all other parameters. The main optimization in Eq.(4.4) can be reduced to the following sub-problem,

$$\begin{aligned} \min_{\mathbf{C}} \quad & \sum_{i=1}^{N} \|\boldsymbol{y}_i - \mathbf{X}_i\widetilde{\mathbf{Q}}_1\mathbf{C}_{1,\cdot i}\|^2 \\ \text{s.t.} \quad & (A) \quad \mathbf{C}_{1,\cdot i} \geq 0,\ \sum_{k=1}^{K_1} \mathbf{C}_{1,ki} = 1,\ \forall i. \end{aligned} \tag{4.7}$$

The constraints for link matrices are eliminated and will be investigated in later steps, due to the back propaganda nature. Here we can see that the membership vector for each individual are mutually independent in the optimization problem. Thus they can be optimized separately, i.e., for individual $i$:

$$\begin{aligned} \min_{\boldsymbol{c}_{1,i}} \quad & \|\boldsymbol{y}_i - \mathbf{X}_i\widetilde{\mathbf{Q}}_1\boldsymbol{c}_{1,i}\|^2 \\ \text{s.t.} \quad & (A) \quad \boldsymbol{c}_{1,i} \geq 0,\ \mathbf{1}^\top\boldsymbol{c}_{1,i} = 1. \end{aligned} \tag{4.8}$$

This is a constrained regression, and can be efficiently solved as a convex optimization problem. In [77], a closed form updating rule is derived from KKT conditions. We will use the following formulation as our $\mathbf{C}_1$ *Step*.

$$\mathbf{C}_{1,ki}^{(new)} = \mathbf{C}_{1,ki} \times \frac{\left[\widetilde{\mathbf{X}}_{1,i}^\top\boldsymbol{y}_i\right]_k + \left[\widetilde{\mathbf{X}}_{1,i}^\top\widetilde{\mathbf{X}}_{1,i}\mathbf{C}_{1,\cdot i}\right]^\top \mathbf{C}_{1,\cdot i}}{\left[\widetilde{\mathbf{X}}_{1,i}^\top\widetilde{\mathbf{X}}_{1,i}\mathbf{C}_{1,\cdot i}\right]_k + \left[\widetilde{\mathbf{X}}_{1,i}^\top\boldsymbol{y}_i\right]^\top \mathbf{C}_{1,\cdot i}} \tag{4.9}$$

In this closed-form updating rule, the $\mathbf{Q}, \mathbf{C}$ matrices used in the right-hand side are current estimations, and $\widetilde{\mathbf{X}}_{1,i}$ is defined as $\mathbf{X}_i\widetilde{\mathbf{Q}}_1 = \mathbf{X}_1\mathbf{Q}_L\mathbf{C}_L \ldots \mathbf{C}_2$ as shown in Table 4.1 in sake of the simplicity in showing the formulation. Further, we can prove that the updating rule showing in Eq.(4.9) has the stationary property.

### *4.3.2 Estimation Steps for Link Matrices (Link Steps)*

This part contains $L-1$ steps from $\mathbf{C}_2$ to $\mathbf{C}_L$. All these steps will have a same form of sub-problem as shown in follows (e.g., at $\mathbf{C}_l$ step),

$$\begin{aligned} \min_{\mathbf{C}_l} \quad & \sum_{i=1}^{N} \|\boldsymbol{y}_i - \mathbf{X}_i \widetilde{\mathbf{Q}}_l \mathbf{C}_l \widetilde{\boldsymbol{c}}_{l-1,i}\|^2 = \sum_{i=1}^{N} \|\boldsymbol{y}_i - \widetilde{\mathbf{X}}_{l,i} \mathbf{C}_l \widetilde{\boldsymbol{c}}_{l-1,i}\|^2 \\ \text{s.t.} \quad & (B) \quad \mathbf{C}_{l,\cdot i} \geq 0,\ \sum_{k=1}^{K_l} \mathbf{C}_{l,ki} = 1,\ \forall i, \\ & (C) \quad \mathbf{C}_l \widetilde{\boldsymbol{c}}_{l-1,i} \geq 0, \\ & \qquad \sum_{k=1}^{K_l} (\mathbf{C}_l \widetilde{\boldsymbol{c}}_{l-1,i})_k = 1,\ \forall i. \end{aligned} \tag{4.10}$$

We will take a gradient step at this stage,

$$\begin{aligned} l(\mathbf{C}_l) &= \sum_{i=1}^{N} \|\boldsymbol{y}_i - \widetilde{\mathbf{X}}_{l,i} \mathbf{C}_l \widetilde{\boldsymbol{c}}_{l-1,i}\|^2 \\ \frac{\mathrm{d}l}{\mathrm{d}\mathbf{C}_l} &= 2 \sum_{i=1}^{N} \widetilde{\mathbf{X}}_{l,i}^{\top} (\widetilde{\mathbf{X}}_{l,i} \mathbf{C}_l \widetilde{\boldsymbol{c}}_{l-1,i} - \boldsymbol{y}_i) \widetilde{\boldsymbol{c}}_{l-1,i}^{\top} \end{aligned} \tag{4.11}$$

Here, we only consider the objective function. A widely used approach is to take a small gradient step each time and then project the new estimation to the feasible area given by the constraints. The computational cost of computing gradients is relatively low, so each step is very fast. However empirically, a projection operation is very likely needed, especially when $\mathbf{C}_l$ matrix is large, so that the objective value may not be monotonically decreasing in each iteration.

Alternatively, we also provide an optimization solution for link matrices as well. Note that the objective function in Eq.(4.10) has the similar format as in $\mathbf{Q}$ step in collaborative learning framework [76, 77]. Hence, we will apply the strategy of vectorization. Denote $\boldsymbol{l} = \mathrm{Vec}(\mathbf{C}_l)$ and also define $\mathbf{X}_i^* = (\widetilde{\boldsymbol{c}}_{l-1,i})^{\top} \otimes \widetilde{\mathbf{X}}_{l,i}$, then we will have the reformulated objective:

$$\min_{\boldsymbol{l}} \quad \sum_{i=1}^{N} \|\boldsymbol{y}_i - \mathbf{X}_i^* \boldsymbol{l}\|^2. \tag{4.12}$$

The constraints can also be reformulated using the vectorized $\boldsymbol{l}$. To see that, we define the following Constraint Matrix $\mathbf{D}$ for constraint $(C)$:

$$\mathbf{D} = \begin{bmatrix} c_{11} & c_{11} & \cdots & c_{k1} & \cdots & c_{K_{l-1}1} \\ c_{12} & c_{12} & \cdots & c_{k2} & \cdots & c_{K_{l-1}2} \\ \vdots & \vdots & & \ddots & & \vdots \\ c_{1N} & c_{1N} & \cdots & c_{kN} & \cdots & c_{K_{l-1}N} \end{bmatrix} \in \mathbb{R}^{N \times K_l K_{l-1}}, \tag{4.13}$$

where $c_{ki}$ are the entries in $\widetilde{\mathbf{C}}_{l-1} \in \mathbb{R}^{K_{l-1} \times N}$, and each entry is repeated $K_l$ times. The constraint $(C)$ is now

$$\mathbf{D}\boldsymbol{l} = \boldsymbol{d}, \tag{4.14}$$

where $\boldsymbol{d}$ is a $N$-length vector with all elements being 1. Lemma 4.2 shows that constraint $(B)$ can be considered as a special case of constraint $(C)$, where $\widetilde{\mathbf{C}}_{m-1}$ is replaced by unit matrix $\mathbf{I} \in \mathbb{R}^{K_{l-1} \times K_{l-1}}$, thus the $\mathbf{D}$ matrix and $\boldsymbol{d}$ can be adjusted accordingly, i.e.,

$$\mathbf{D} = \begin{bmatrix} \mathbf{1}^\top & \mathbf{0}^\top & \cdots & \mathbf{0}^\top \\ \mathbf{0}^\top & \mathbf{1}^\top & \cdots & \mathbf{0}^\top \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}^\top & \mathbf{0}^\top & \cdots & \mathbf{1}^\top \end{bmatrix} \in \mathbb{R}^{K_{l-1} \times K_l K_{l-1}}, \qquad \boldsymbol{d} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^{K_{l-1}}. \tag{4.15}$$

In summary, each Link Step can be reformulated as the following sub-problem:

$$\begin{aligned} \min_{\boldsymbol{l}} \quad & \sum_{i=1}^{N} \|\boldsymbol{y}_i - \mathbf{X}_i^* \boldsymbol{l}\|^2 \\ \text{s.t.} \quad & \mathbf{D}\boldsymbol{l} = \boldsymbol{d}, \quad \boldsymbol{l} \geq 0. \end{aligned} \tag{4.16}$$

It is now a simple quadratic programming which can be solved in polynomial time [36, 87], and there are off-the-shelf solvers available [42, 37].

### *4.3.3 Estimation Step for Canonical Models (**Q** Step)*

In this step, we focus on solving the highest level canonical matrix $\mathbf{Q}_L$ with all $\mathbf{C}$ matrices fixed. As the constraints are all with regard to $\mathbf{C}$ matrices, this sub-problem is an uncon-

strained optimization problem.

$$\min_{\mathbf{Q}_L} \quad \sum_{i=1}^{N} \|\boldsymbol{y}_i - \mathbf{X}_i \mathbf{Q}_L \widetilde{\boldsymbol{c}}_{L,i}\|^2 \tag{4.17}$$

Similar as in Link Step, there are two approaches for updating $\mathbf{Q}_L$. First, we can take one step of gradient descent where the gradient is as follows,

$$\frac{\mathrm{d}l}{\mathrm{d}\mathbf{Q}_L} = 2 \sum_{i=1}^{N} \mathbf{X}_i^\top (\mathbf{X}_i \mathbf{Q}_L \widetilde{\boldsymbol{c}}_{L,i} - \boldsymbol{y}_i) \widetilde{\boldsymbol{c}}_{L,i}^\top. \tag{4.18}$$

We can also apply the strategy of vectorization as well. Denote $\boldsymbol{q} = \mathrm{Vec}(\mathbf{Q}_L)$ and define $\mathbf{X}_i^* = (\widetilde{\boldsymbol{c}}_{L,i})^\top \otimes \mathbf{X}_i$, we will have Eq.(4.17) simplified as:

$$\min_{\boldsymbol{q}} \quad \sum_{i=1}^{N} \|\boldsymbol{y}_i - \mathbf{X}_i^* \boldsymbol{q}\|^2 \tag{4.19}$$

Stack the outcome vectors $\boldsymbol{y}^* = \begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_N \end{bmatrix}$ and design matrices $\mathbf{X}^* = \begin{bmatrix} \mathbf{X}_1^* \\ \vdots \\ \mathbf{X}_N^* \end{bmatrix}$. It will be identical to multivariate regression, in other words:

$$\hat{\boldsymbol{q}} = (\mathbf{X}^{*\top} \mathbf{X}^*)^{-1} \mathbf{X}^{*\top} \boldsymbol{y}^* \tag{4.20}$$

Notice that by the definition of HCM, the number of top level canonical models $K_M$ is usually a small number, which won't lead to dimensional curse for the inverse operation of $\mathbf{X}^{*\top}\mathbf{X}^*$). However, as population ($N$) can be very large in the application of HCM, the computational cost for tidying auxiliary matrix $\mathbf{X}^*$ and the additional storage cost may still be high.

In summary, the procedure of the proposed "back propagation" algorithm for HCM is shown in Algorithm 2.

In Algorithm 2, the Link steps and $\mathbf{Q}$ step can also apply gradient method as discussed in this section, which will lead to less computational time in each iteration but take more iteration rounds to converge.

**Algorithm 2** The Learning Algorithm for HCM

**Input:**

Data $\mathbf{X}_i$ and $\boldsymbol{y}_i$ for all $i = 1, \ldots, N$;

Initial values $\mathbf{Q}_L^{(0)}$ and $\mathbf{C}_l^{(0)}$ for all $l = 1, \ldots, L$;

Maximum iteration number $MaxIter$;

**Output:**

$\mathbf{Q}_L^{(MaxIter+1)}$, $\mathbf{C}_l^{(MaxIter+1)}$ for all $l = 1, \ldots, L$.

1: **for** each $m \in [0, MaxIter]$ **do**
2: calculate $\mathbf{C}^{(m+1)}$ by applying Eq.(4.9).
3: **for** each $l \in [2, L]$ **do**
4: solve quadratic optimization of Eq.(4.16) and get $\boldsymbol{l}$;
5: transform $\boldsymbol{l}$ to $\mathbf{C}_l^{(m+1)}$ by partitioning the vector to the $K_l \times K_{l-1}$ matrix;
6: **end for**
7: solve Eq.(4.19) or Eq.(4.20) and get $\boldsymbol{q}$;
8: transform $\boldsymbol{q}$ to $\mathbf{Q}_L^{(m+1)}$ by partitioning the vector to the $P \times K_L$ matrix;
9: **end for**

### *4.3.4 Empirical Guidelines for Initializing CHCM*

In concluding this section, we introduce some empirical guidelines to initialize the CHCM before the Algorithm 2 can be applied.

Contextual HCM utilizes the characteristic information to specify the hierarchical structure of HCM, hence echos the complex composition of the heterogeneous population. The CHCM structure implicitly assumes that the canonical models correspond to the subpopulations divided by those characteristic variables. Therefore, we can initialize the canonical models for every level first, by estimating the average model for each group, defined by those characteristic. To be specific, if characteristic A defines 4 groups, and the top level canonical models can be initialized by modeling these 4 group-level models, each representing the com-

monality within this group. Further, if the next level corresponds to characteristic A and B together, defining 12 groups, then the 12 canonical models in this level can be initialized these 12 group-level models.

After having $\mathbf{Q}_l^{(0)}$ for all $l = 1, \ldots, L$ levels, we can gain the initial link matrices by solving an optimization problem similar to Eq.(2.12). Because in HCM, the two consecutive levels of the canonical models have the same collaborative structure as in single-layer CM, i.e., Eq.(4.2).

Finally, for the last membership matrix $\mathbf{C}_1^{(0)}$, we can assign them based on the characteristics of the individuals. Since $\mathbf{Q}_1^{(0)}$ corresponds to the subpopulation defined by the characteristics, for each individual, we can find the corresponding canonical model based on his/her characteristic, and set it as the dominant, by assigning a large weight, e.g. 0.8. All the other initial canonical models at level 1, which correspond to other groups of individual, can be assigned randomly with the remaining weight.

### *4.4 Simulation Studies*

To evaluate the model performance, we will compare the hierarchical collaborative model (HCM) with single-level collaborative models. As HCM provides multiple levels of canonical models, it can explain the individualized models under various granularity. Meanwhile, given $K$, single-level CM has only one set of canonical models on the same level, so it only represent one certain degree of canonical details. Therefore, we compare single-level CM with different $K$ choices to see whether it can reveal similar heterogeneity structure of the population as the HCM does. To be specific, we preset $K = 4, 20, 100, 500$ respectively for 4 different single-level CMs. In addition, to show the heterogeneity in population, we will also report the one-size-fits-all linear regression (LR) as well.

Several metrics will be used to compare the performances. For each individual, a regression model is fit for all methods. Therefore, prediction error, in terms of total squared loss on testing set is reported, and the overall loss (Mean Squared Loss, $MSE$) will be calculated

by taking average across all individuals, i.e.,

$$MSE = \frac{1}{N}\sum_{i=1}^{N} \|\widehat{\boldsymbol{y}}_i - \boldsymbol{y}_i\|^2$$

We also report the Mean Absolute Error ($MAE$) and their relative version, i.e., Mean Mean Absolute Percentage Error ($MAPE$) and Mean Squared Percentage Error ($MSPE$) to filter out the magnitude effects.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |\widehat{\boldsymbol{y}}_i - \boldsymbol{y}_i|$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N} \frac{|\widehat{\boldsymbol{y}}_i - \boldsymbol{y}_i|}{|\boldsymbol{y}_i|}$$

$$MSPE = \frac{1}{N}\sum_{i=1}^{N} \frac{\|\widehat{\boldsymbol{y}}_i - \boldsymbol{y}_i\|^2}{\|\boldsymbol{y}_i\|^2}$$

R-squared ($R^2$), known as the coefficient of determination, is another common way in evaluating the prediction performance of a regression model, it is defined as how much of sum of the squares is explained by the prediction model. A better model should have a higher $R^2$. We also take the average of $R^2$ across all the individualized models to see how well each method predicts.

In addition, as a simulation study, real parameters for each individual ($\mathbf{B}$) are known. We will also compute the estimation error as well, using the absolute ($MAE, MAPE$) and squared error ($MSE, MSPE$) between real and estimated individual model coefficients. All the metrics are reported using average across all individuals.

Finally, the total loss (objective value) averaged by individuals on training data is also reported, as an indicator of final result of optimization problem.

Table 4.2 shows the result comparing the proposed HCM and single-level CM models with different $K$ choices. First of all, from the realut of LR, we can confirm that there is significant heterogeneity in the population so that the one-size-fits-all model does not work well. HCM and single-level CM's all worked well in learning indivdiual models with limited

Table 4.2: Model performance comparison based on average prediction and estimation errors on the simulated data

| Model | LR | CM-4 | CM-20 | CM-100 | CM-500 | HCM |
|---|---|---|---|---|---|---|
| Average $R^2$ | 0.2885 | 0.8919 | 0.8917 | 0.8691 | 0.8181 | **0.8946** |
| Prediction MSE | 709.41 | 99.907 | 100.67 | 119.56 | 166.78 | **98.693** |
| Prediction MSPE | 0.6242 | 0.0942 | 0.0958 | 0.1148 | 0.1574 | **0.0926** |
| Prediction MAE | 45.806 | 17.818 | 17.878 | 19.514 | 22.909 | **17.718** |
| Prediction MAPE | 0.6836 | 0.2716 | 0.2728 | 0.2981 | 0.3489 | **0.2698** |
| Estimation MSE | 125.86 | 2.2045 | 2.3538 | 6.0015 | 15.753 | **2.0100** |
| Estimation MSPE | 0.3927 | 0.0064 | 0.0071 | 0.0180 | 0.0460 | **0.0061** |
| Estimation MAE | 114.83 | 14.020 | 14.527 | 23.471 | 38.147 | **13.175** |
| Estimation MAPE | 0.5912 | 0.0706 | 0.0737 | 0.1189 | 0.1920 | **0.0669** |
| Average Loss | 2868.4 | 374.93 | 361.00 | 362.75 | 391.91 | **358.79** |

data. Based on both prediction accuracy and estimation accuracy, we can observe that the proposed HCM method outperforms all the single-level collaborative models. Figure 4.2 also compares the $MAPE$ and $MSPE$ of these models. HCM performs slightly better than CM with $K = 4$ and CM with $K = 20$. A trend that the accuracy is getting worse when $K$ is larger can be observed. It is understandable as more canonical models will have more free parameters to estimate, which could be a hazard when data is limited.

Besides the better performance than all single-level CM, HCM shows its strength in other two aspects. First, HCM learns all 4 levels of canonical models at once, which covers all the single-level collaborative models. However, in order to achieve similar information about canonical models, i.e., different degrees of details, single-level collaborative model has to be applied 4 times with different $K$'s each time. Plus, to learn a single-level CM, we always have to start from scratch, as they do not borrow or transfer information from other CM models. Second, due to its hierarchical structure and chain rule, interpreting HCM at different levels is

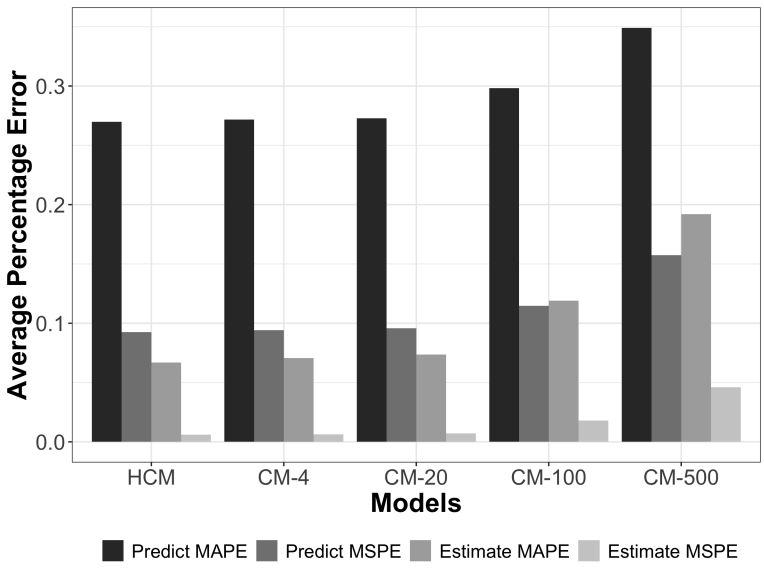


Figure 4.2: Model performance comparison based on average prediction and estimation errors, on the simulated testing data.

equally accurate. However, as single-level collaborative models with different $K$'s are learned independently, the performance with varying degrees of details will be different. And as we observed in Figure 4.2, when $K$ is large, both prediction and estimation performance in CM is worse. It indicates that when we try to investigate more detailed common patterns, CM may not work well.

Furthermore, unlike Contextual HCM, single-level collaborative modeling does not have a hierarchical structure to reflect the composition of population. Therefore, it is usually the case that the number of canonical models ($K$) for collaborative model has to be decided in a data-driven manner such as cross-validation. As we have discussed in Chapters 2 and 3, the ground-truth $K$, if there is one, is not guaranteed to be found by such techniques. In addition, there might not be a "real" $K$ in complex populations such as the case for HCM, where $K$ may be a series of value instead of one. Therefore, we further tested single-level collaborative model with more $K$ choices ($K = 2, 3, 5, 10, 30, 50, 200$ along with $K = 4, 20, 100, 500$), and compare our proposed HCM to them. Figure 4.3 present the result. The horizontal line is the

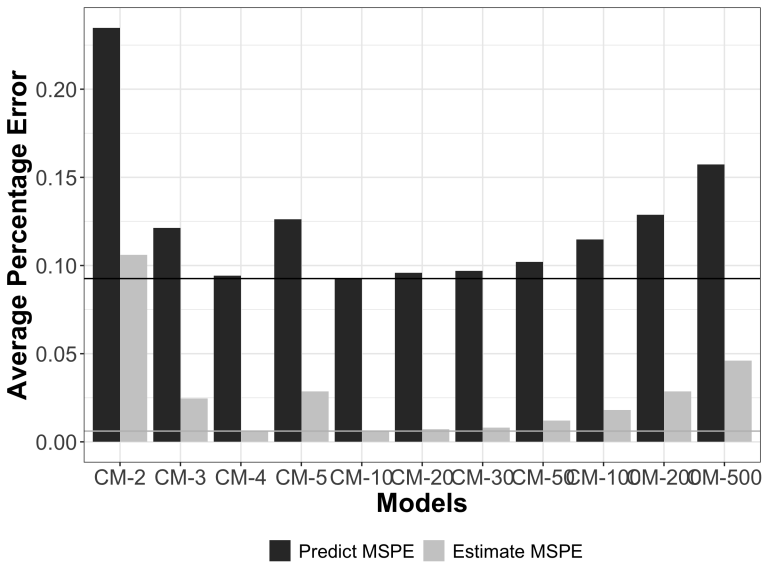


Figure 4.3: Model performance comparison with more $K$ selections for single-level CMs based on average prediction and estimation errors, on the simulated testing data.

performance of HCM. It can be seen that HCM outperforms all the single-level CM models. The trend of worse performance when $K$ is larger can be observed again. In addition, when $K$ is very small, the performance will also drop. This is also understandable, since too few canonical models may cause some information lost for the data.

## *4.5 Conclusion*

This chapter proposes the Hierarchical Collaborative Model, where multiple levels of canonical models will be learned simultaneously. HCM has an elegant hierarchical structure so that the collaborative structure is embedded between different levels, and between various levels of canonical models and individual models. We formulate HCM with linear regression as an optimization problem, and propose Contextual HCM, showing how characteristic information can be connected with the HCM structure. Such connection of the data structure and model structure makes CHCM a suitable model for populations with complex composition, and grants CHCM good interpretability with all those levels of canonical models. In the

future, the HCM framework can be combined with non-linear base models so that it will have more potential to fit in various circumstances.

# Chapter 5

# LDT: LATENT DECISION THRESHOLD MODEL FOR MODELING USER-SYSTEM INTERACTIONS BY GRAPHICAL MODEL APPROACH AND MAX-MARGIN LEARNING

In personalized interactive systems, the personalized recommendations or tailored incentives which give to the users are designed based on previous user behaviors, and are designed to coordinate or change user future behaviors. The new interaction will also becomes new data point of the history, which will then affect future recommendation and incentive designs. Thus, these smart systems not merely are data collection tools but also change users' behaviors. In this chapter, We propose a graphical model to better characterize this unprecedented connection, named Latent Decision Threshold Model (LDT) [33]. In addition, an efficient computational strategy is developed based on a max-margin formulation, to overcome the learning challenge of the non-linearity of graphical models.

## *5.1 Introduction*

Many emerging applications that provide personalized services for target users [129, 70, 63, 130, 93, 6, 4, 132], aim to coordinate and change user behaviors by implementing personalized incentives or rewards. For example, the smart TDM system proposed in [132] could offer personalized promotions with tailored rewards, to each commuter to change his/her travel behaviors. As mentioned before, traditional TDM strategies are usually developed based on the population level, but the new smart TDM system will design the promotions and the tailored incentives based on the individual level, to be specific, modeling from each user's interaction history with the system. It has been reported in [132] that the personalized TDM system is quite effective in changing users' behaviors, i.e., the acceptance rate of

the promoted suggestions reaches 68%, which leads to a significant travel time saving and congestion mitigation on the transportation system.

To fully unleash the potential of such personalized interactive systems, understanding user behavior at the individual level is very important, which is the main focus of previous chapters. Equally important is to understand user behavior through such user-system interactions. It is an enabling factor in further intervention strategy development and optimization of user experience. The application of user models can improve the personalized recommendations [19, 64] and it is essential for assigning proper personalized incentives such as monetary rewards in order to encourage behavioral changes. Discrete choice models based on the theory of Random Utility Maximization (RUM) [25, 82, 12, 55] such as logit model are widely used in learning and understanding behavioral preferences for their good interpretability. However, they commonly assume that the data collection is independent with the human behavior, in other words, the technology for data collection is usually considered as irrelevant and will not interfere with the object to be observed and modeled.

The connection between the data collection and user behavior in the new system needs a more advanced statistical model to characterize. While not exactly concerning the same problem as ours, there have been some studies in the literature on the endogeneity between choices and the decisions, or choices and the travel behaviors. In [47], it deals with the endogeneity caused by model misspecification (i.e., omission of the attributes) using control function and latent variable. It does not relate the data collection procedure with the change of user behavior caused by a reward mechanism typical in nowadays smart TDM systems. In [46], it corrects the endogeneity which comes from the stated preferences and revealed preferences. Ben-Akiva et al. (2012) [10] describes a framework of incorporating contexts like social network as the decision being made may also be affected by family, friends, and other choices being given. Both works do not concern personalized reward systems. In [50], a dynamic model is used from decision field theory to characterize the changing preferences with sequential choices and decisions, which concerns a different problem from ours.

Thus, none of the above approaches are fully applicable to the unique challenge posed

by the personalized interactive system where rewards are influenced by the alternative's attributes and personal preferences. To fill in this gap, we propose a graphical model to depict the user-system interaction and to model the user behavior. Graphical model is a generic term referring to a family of multivariate statistical models that specifically model the interactions among variables and derive their data-generating process [69]. Using the graph, the model can represent the conditional dependencies among the variables, which enables graphical model to reveal the relationship between variables and makes it easy to interpret. Some existing hybrid choice models could also be potentially cast into the framework of graphical model, e.g., similar efforts have been undertaken in the literature for a range of linear models that show in the framework of graphical models many existing models find a unifying framework [101]. A distinct concept that separates our proposed method with the existing works is the concept of "decision threshold", which is the tipping point where a user may change his/her decision between alternatives under a combined influence of the alternative's attributes and personal preferences. To model this new mechanism we develop the graphical model named the Latent Decision Threshold (LDT) model. It can characterize a user's decision behavior considering the attributes of the alternatives, the user's preferences, and the user's decision threshold between the alternatives. As the tipping point encodes a nonlinear relationship among variables, our proposed LDT model is a new type of graphical model that tailors the user-system interaction mechanism and provides a more delicate understanding of how a user makes a decision in a particular situation influenced by rewards.

Many graphical models and algorithms are very computational costly [73, 123], particularly when the interactions among variables are not all linear, as in our proposed LDT model. Therefore, another important contribution of this work is that we further reveal an interesting connection between the parameter estimation problem of the LDT model and max-margin learning. Using this connection, we resort to the computationally efficient solutions of max-margin formulation as a better approach than algorithms such as Expectation-Maximization algorithm (EM) which is usually used when a graphical model has latent variables.

The work of LDT is organized as follows. In Section 5.2, we review the example of smart TDM system [132], and develop a characterization of the user-system interaction process using existing models for user behavior such as logit model and mixed logit model (MLM). It will illustrate the importance of noticing the interactive nature of such personalized system. In Section 5.3, we develop the formulation of the proposed LDT model and construct an efficient parameter estimation algorithm based on the max-margin learning principle in Section 5.4. Comprehensive simulation studies are conducted in Section 5.5. Real case study on the smart TDM system is presented in Section 5.6. Section 5.7 includes a brief conclusion and discussion.

## *5.2 Background and Motivation*

### *5.2.1 Personalized TDM System*

To better introduce the proposed LDT model and demonstrate its performance, we use a specific example of such reward systems, which we have used in Section 2.5. When a commuter is about to depart, one can request a trip in the app and a promoted alternative travel plan will be generated and offered by the system. The promoted plan is designed based on the system's knowledge of this commuter, i.e., based on the decision choice model that can be learned from previous interactions between the user and the app system.

The alternative will differ from the original travel plan in departure time and total travel time, which means the attributes that characterize the choices are Schedule Delay Early ($SDE$), Schedule Delay Late ($SDL$), and Travel Time Saving ($TTS$) [132, 13, 14]. The system will also determine the amount of reward points ($Reward$) needed for the potential acceptance of the promoted alternative [132]. In other words, a personalized incentive should be derived based on the alternatives' attributes and the user's preferences to encourage the user to switch to the new alternative. An example of user scenario is shown in Figure 2.2 where $SDE = 10, SDL = 0, TTS = 20, Reward = 10$, and several rows of the user-system interaction history data is shown in Table 2.3.

### *5.2.2 Limitations of the Logit Models*

The discrete choice models based on Random Utility Maximization (RUM) theory are found effective in learning behaviors or preferences. Here, we concentrate on a class of RUM known as logit models, similar as in previous chapters. The RUM theory assumes that the probability of selecting among alternatives depends only on the differences in their utilities, and a user will select the alternative with the highest utility, where utility is a concept quantifying the attractiveness of an alternative [12, 55, 53] which is defined to be related to attributes, for example, in our example,

$$U_A = V_A + \epsilon_A = \beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}_A + \epsilon_A = \beta_0 + \sum_{p=1}^{P} \beta_p x_{Ap} + \epsilon_A,$$

$$U_B = V_B + \epsilon_B = \beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}_B + \epsilon_B = \beta_0 + \sum_{p=1}^{P} \beta_p x_{Bp} + \epsilon_B,$$

where $V = \beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}$ is the measurable systematic utility and $\epsilon$ is the random utility [12]. The parameter $\boldsymbol{\beta}$ shows the preferences of the user, i.e., a positive $\beta_p$ indicates that this attribute $x_p$ of the alternative is attractive to the user, granting higher utility to the alternative and making it more likely to be chosen. With the multinomial logit (MNL) formulation [82], the logit model can be expressed as the probability of choosing among alternatives. Furthermore, $\boldsymbol{x}$ could be the differences between the attributes of the two alternatives [55, 83] for applications with two choices. Specifically, to apply the logit model on the new personalized TDM system as shown in Figure 2.2, where four attributes are $x_{SDE} = 10$, $x_{SDL} = 0$, $x_{TTS} = 20$, and the monetary reward points $r = 10$, we use $y = -1$ to denote that the user chooses Choice A, and use $y = 1$ to denote that the user chooses Choice B, the logit model will lead to the following expression of the probability of accepting the promotion ($y = 1$):

$$Pr(y = 1) = \frac{\exp(V_B)}{1 + \exp(V_B)}, \text{ where } V_B = \beta_0 + \beta_{SDE} x_{SDE} + \beta_{SDL} x_{SDL} + \beta_{TTS} x_{TTS} + \beta_r r. \quad (5.1)$$

As the RUM theory is used for population-level modeling, a more related model in our context is the mixed logit model (MLM) (also called the random parameters logit model) that allows the parameters of the individuals vary. It can model the heterogeneity of the population [45, 54, 8] by modeling the individual-level parameters $\boldsymbol{\beta}^{(i)}$'s as random samples drew from a multivariate normal distribution. For example, the probability that individual $i$ will accept the alternative can be expressed in the form [116, 54, 85]:

$$Pr(y=1) = \int_{\boldsymbol{\beta}^{(i)}} \frac{\exp(V_B)}{1+\exp(V_B)} f(\boldsymbol{\beta}^{(i)}) d\boldsymbol{\beta}^{(i)}, \tag{5.2}$$

where $f(\boldsymbol{\beta}^{(i)})$ is the density function for all the individual-level parameters $\boldsymbol{\beta}^{(i)}$, and it can be specified as a multivariate normal distribution with unknown mean and covariance. We can treat all the preference parameters as random parameters and the systematic utilities $V_B$ in Eq.(5.2) are based on such personal preferences, i.e.,$V_B = \beta_0^{(i)} + \beta_{SDE}^{(i)} x_{SDE} + \beta_{SDL}^{(i)} x_{SDL} + \beta_{TTS}^{(i)} x_{TTS} + \beta_r^{(i)} r$. MLM will be a baseline in our study to be compared with our proposed LDT model, and results are shown in Sections 5.5 and 5.6.

However, after applying the logit model and MLM to analyze the data collected in the smart TDM system, we observe some counter-intuitive results. Table 5.1 shows the estimated coefficients of the logit model for the population. The fact that $\hat{\beta}_r$ in this logit model is negative literally implies the higher the rewards, the lower the probability the user could accept the promotion. If this were true, it will lead to a consequence that we cannot change user's travel behavior by providing more rewards. On top of this conceptual difficulty to understand the model, we also observe an unusual statistical phenomenon: the estimated coefficients are close to zero. Later in-depth analysis in Section 5.6 will reveal that this may be due to the large heterogeneity of the users. Hence we also apply mixed logit model (MLM) as shown in Eq.(5.2) to learn personalized models. The results of MLM are shown in Table 5.2.

MLM did not alleviate the difficulty in interpretation; rather, more counter-intuitive signs of $\hat{\boldsymbol{\beta}}^{(i)}$ are observed. For instance, a considerable portion of $\hat{\beta}_r^{(i)}$ is negative. This paradox leads us to hypothesize that we could not take it at its face value, i.e., $\hat{\beta}_r^{(i)}$ not only estimates

Table 5.1: Estimated coefficients using population-level logit model on the smart TDM system data

| | $\hat{\beta}_0$ | $\hat{\beta}_{SDE}$ | $\hat{\beta}_{SDL}$ | $\hat{\beta}_{TTS}$ | $\hat{\beta}_r$ |
|---|---|---|---|---|---|
| Estimated Coefficients | 1.9161 | -0.0210 | -0.0290 | 0.0504 | -0.0202 |
| p-value | $0.000^{***}$ | $0.000^{***}$ | $0.000^{***}$ | 0.327 | $0.000^{***}$ |

Table 5.2: Results of the Mixed Logit Model (MLM) on the smart TDM system data

| | $\hat{\beta}_0$ | $\hat{\beta}_{SDE}$ | $\hat{\beta}_{SDL}$ | $\hat{\beta}_{TTS}$ | $\hat{\beta}_r$ |
|---|---|---|---|---|---|
| % of Counter-Intuitive Sign | – | 15.81% | 1.76% | 21.25% | 38.84% |

$\beta_r^{(i)}$ but also something else. This indicates that the data-generating mechanism is different from the one assumed by the logit model and MLM. The same observation could be made on $\hat{\beta}_{TTS}^{(i)}$, as time saving should also be an encouraging factor for users and is supposed to have positive sign [13, 14].

### *5.2.3 Analysis of the Paradox*

The counter-intuitive results shown in Tables 5.1 and 5.2 imply a mismatch of the modeling framework as illustrated in Eq.(5.1) with the underlying mechanism of the users' decision-making process when using such reward systems. To illustrate this, Figure 5.1 shows a conceptual understanding of the data-generating mechanism assumed by the logit models. It takes all the attributes including the reward as a set of variables with additive effects to predict the final decision. This "flattened" treatment of all attributes in the decision-making behavior as a set of variables with interchangeable positions in the same layer, while each variable's effect is additive, is probably an oversimplification of the problem that caused the counter-intuitive results.

To provide a remedy for this problem, first, it is worthy of pointing out that in existing app-based TDM systems it is a common strategy that the rewards assigned to the promo-

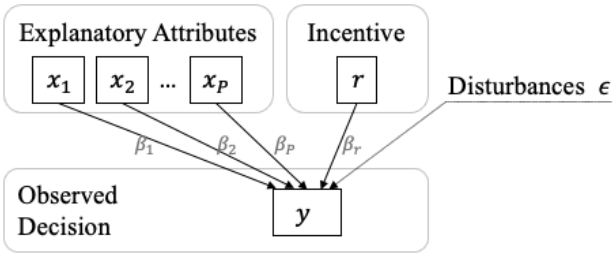


Figure 5.1: An illustration of the logit model based on RUM.

tions are usually related to the attributes, e.g., in [132], when an alternative travel plan is generated, the reward points for this alternative are derived based on the attributes and the estimated user preferences. In other words, $r$ is often correlated with other attributes $\boldsymbol{x}$. This correlation or multicollinearity has been known to be a hazard to models that have linear forms where the logit models are no exception. Lemma 5.2 articulates the danger of applying those models with one-layer flattened structure to model the user behavior in interactive apps and offers an explanation of the unstable and counter-intuitive estimations.

**Lemma 5.2:** *In logit models, when the reward $r$ is linearly related to the attributes $\boldsymbol{x}$, there will be an infinite set of estimated coefficients $\hat{\boldsymbol{\beta}}$ that can lead to same utilities, and therefore, all maximize the log-likelihood of the model. In some of these solutions, $\hat{\beta}_r$ may be negative.*

The proof is provided in the Appendix D.1. Lemma 5.2 indicates that the parameter estimations of the logit model will be unstable and misleading when the attributes are correlated with rewards $r$. It is then of interest to see if this is true in the personalized TDM system data we have used in this study. Table 5.3 shows the population-level correlations between rewards $r$ and the other three attributes, i.e., $x_{SDE}$, $x_{SDL}$ and $x_{TTS}$, by pooling all users' data together. The result shows that $r$ is significantly correlated with other three attributes. We also check the individual-level correlations (i.e., using each user's own data). The average correlations and the average of absolute values of correlations over all users are

Table 5.3: Correlations between rewards $r$ and other attributes $\boldsymbol{x}$ in the smart TDM system data

| | $x_{SDE}$ and $r$ | $x_{SDL}$ and $r$ | $x_{TTS}$ and $r$ |
|---|---|---|---|
| Correlation | 0.0296 | -0.0733 | -0.1890 |
| p-value | 0.006*** | 0.000*** | 0.000*** |
| Average of personal correlations | 0.1036 | -0.0141 | -0.0526 |
| Average of absolute personal correlations | 0.3299 | 0.2993 | 0.2058 |

Table 5.4: Estimated coefficients using linear regression model for $r$ on $\boldsymbol{x}$ on the smart TDM system data

| | $\hat{\gamma}_0$ | $\hat{\gamma}_{SDE}$ | $\hat{\gamma}_{SDL}$ | $\hat{\gamma}_{TTS}$ |
|---|---|---|---|---|
| Estimated Coefficients | 39.558 | 0.0106 | -0.0793 | -0.7485 |
| p-value | 0.000*** | 0.481 | 0.000*** | 0.000*** |

reported in Table 5.3 as well. We can observe that there are users whose individual-level correlations between rewards and other attributes are high. Note that here we report the absolute values because the magnitude reflects the strength of correlation.

To further examine if the condition in Lemma 5.2 exists in this data, we build a population-level linear regression model for $r$ using $\boldsymbol{x}$ as predictors, i.e., $r = \gamma_0 + \boldsymbol{\gamma}^\top \boldsymbol{x} + \epsilon$. Table 5.4 shows the result, clearly suggesting that there is significant correlation between the reward and other attributes. While this analysis is done on the population level, i.e., we pooled all users' data and build one regression model, we also built personalized regression models, i.e., we built a regression model for each individual, using only his/her own data, and a similar observation could be obtained.

Although a common way to deal with multicollinearity among the variables is to apply variable selection techniques before modeling, i.e., to remove the highly correlated variables, it is not suitable in our case because all the variables are important and contextually mean-

ingful although statistically correlated, and the reward $r$, which is the highly correlated variable, is a key component for the success of personalized reward system and should be included in the model. Principal Component Analysis (PCA) [125] is another common practice in dealing with multicollinearity as it projects related variables into a new coordinate system such that the new features are not linearly correlated. However, it will lead to a loss of interpretability as the variables are transformed into new features. Putting all together, we conclude that there is a significant gap between the existing literature with the personalized TDM systems we aim to study. To fill in this gap, we propose a graphical model to characterize the data-generating mechanism in these user-system interactions in the following sections.

## *5.3 The Latent Decision Threshold (LDT) Model*

Motivated by the limitations of the RUM-based models discussed in Section 5.2, we propose Latent Decision Threshold (LDT) model in this section, aiming to provide a fair characterization of the data-generating mechanism underlying users' decision-making, which is shown in Figure 5.2. Squares indicate observed variables and the ellipse indicates the latent variable. Unlike the logit models shown in Figure 5.1 that flatten the multi-layered mechanism, the LDT model uses a middle layer of latent variable to model the decision threshold and its interaction with the reward $r$. Thus, LDT model consists of two parts: the latent variable model, and the decision model.

LDT model postulates that each alternative the user faces has a certain cost or difficulty, which is related to the explanatory attributes. For example, in transportation demand management, the user might risk being late for work if asked to depart 10 minutes later than he/she usually does. It might cost an individual more time or more fuel on the road if he/she has to change the route. And it may be hard for some users to change their schedules [13, 14]. Therefore, TDM strategies either design regulations like toll [89, 100] or increase parking costs [99, 102] to reduce the traffic demand, or offer incentives like monetary rewards [13, 14] or free public transport [7] to compensate such cost and encourage people to switch

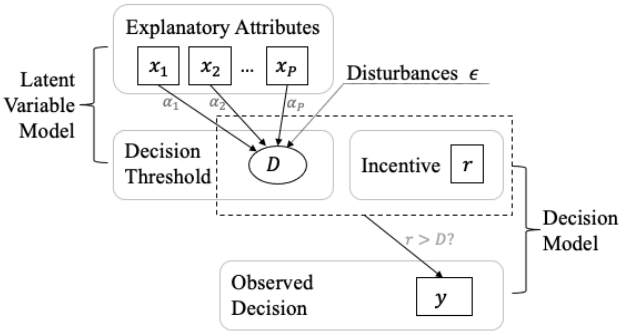


Figure 5.2: An illustration of the Latent Decision Threshold (LDT) model.

to a desirable travel plan.

The second part of the LDT model, the decision model, characterizes the comparison mechanism. For a user to accept a choice, the given incentive should be strong enough to offset the cost or difficulty. The tipping point cost or difficulty which is comparable with the reward, is called the **Decision Threshold** (denote as $D$), and any amount larger than $D$ would change the user's decision. In other words, if the incentive exceeds the threshold, the user will accept the promotion, otherwise, he/she will reject it. Hence, this decision threshold can be considered as the minimum reward the system needs to provide to the user for a given alternative. Eq.(5.3) shows the decision-making part of the LDT model:

$$y = \begin{cases} 1, & r > D; \\ -1, & r \leq D. \end{cases} \tag{5.3}$$

This design will lead to the following consolidated formulation for the decision model:

$$y(r - D) \geq 0, \tag{5.4}$$

which can facilitate the development of our optimization solution in Section 5.4.

The first part of the LDT model, the latent variable model, characterizes user preferences on the attributes and captures the relation between the attributes and the latent variable,

the decision threshold $D$. In this work, we use the linear model as shown in Eq.(5.5).

$$D(\boldsymbol{x}) = \alpha_0 + \alpha_1 x_1 + \ldots + \alpha_M x_M + \epsilon = \alpha_0 + \boldsymbol{\alpha}^\top \boldsymbol{x} + \epsilon, \tag{5.5}$$

where parameter $\boldsymbol{\alpha}$ reflects user preferences on the attributes. Here we use $\boldsymbol{\alpha}$ to represent the user preferences instead of $\boldsymbol{\beta}$ which has been used in logit models, to distinguish the two because of their different meanings. Their signs imply different interpretations. As discussed in Section 5.2, a positive $\beta_p$ indicates that attribute $x_p$ is attractive to the user, granting higher utility to the alternative and makes it more likely to be chosen. However, a positive $\alpha_p$ means that the decision threshold will be increased, and makes it harder for a user to accept the promoted alternative. The magnitude of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ also differs, because $\alpha_p$ indicates the "cost" of change for attribute $x_p$ in terms of the reward, but $\beta_p$ is not comparable with reward unless normalized by $\beta_r$, i.e., $\beta_p/\beta_r$.

### *5.4 Parameter Estimation Algorithm*

As shown in Figure 5.2, there are two types of unknown parameters: the preferences on the attributes, $\boldsymbol{\alpha}$, and the latent variable, the decision threshold $D$. It is important to recognize that if we know $\boldsymbol{\alpha}$, we could readily derive $D$ based on Eq.(5.5). Hence, $D$ is an intermediate parameter.

As LDT is a graphical model, we could use the Expectation-Maximization (EM) algorithm to estimate the latent variable and the other parameters in iterative steps, i.e., the Expectation step estimates the sufficient statistics of the unobserved variable, given the observed data and current estimates of the coefficients, while the Maximization step takes the estimated complete data and estimates the coefficients [28]. However, EM algorithm is not ideal here. First, to use the EM algorithm, a joint likelihood function is needed which requires additional probabilistic assumptions for all the variables including the latent variable. Construction of likelihood function also asks for conditional probability distribution of the observed variables conditional on the latent variable. Also, when the distribution for the latent variable is continuous as here in our case, the computation of integral is usually needed.

However, not like in some other graphical models, here we have no closed-form for the E-step due to the unique mechanism outlined in Eq.(5.3). It will lead to extra computational difficulty since an approximation method is often needed [73, 123].

Thus, we propose a computational method that is based on the max-margin learning, as LDT model bears such an interesting structure that could be utilized. We then no need to estimate the intermediate latent variable $D$. To see that, we can consolidate Eqs.(5.4) and (5.5) as one inequality for each binary choice scenario:

$$y(r - \alpha_0 - \boldsymbol{\alpha}^\top \boldsymbol{x}) \geq 0. \tag{5.6}$$

Here $\boldsymbol{x}$, $r$ and $y$ are the observed attributes, reward, and the final decision, respectively for each binary choice scenario. $\alpha_0$ and $\boldsymbol{\alpha}$ are individual-specified parameters to be estimated. This reformulation removes the need to involve $D$, and Eq (5.6) encodes all the constraints that we need the estimated LDT model to satisfy.

To enforce regularization to overcome the risk of overfitting, we adopt the principle of max-margin and develop the following formulation (for each individual with all his/her data points):

$$\begin{aligned} \min_{\alpha_0, \boldsymbol{\alpha}, \xi} \quad & \frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{\alpha} + \lambda \sum_j \xi_j \\ \text{s.t.} \quad & y_j(r_j - \alpha_0 - \boldsymbol{\alpha}^\top \boldsymbol{x}_j) \geq 1 - \xi_j, \text{ for all } j, \\ & \xi_j \geq 0, \text{ for all } j, \end{aligned} \tag{5.7}$$

by introducing small violations $\xi_j$'s. This follows the idea of non-separable Support Vector Machine (SVM) to tolerate violations on a certain level [26, 23]. The objective of this learning formulation is similar as in SVM, to balance the margin of classifying hyperplane and the small violations, controlled by the tuning parameter of $\lambda$. This algorithm naturally fits into the LDT model and the comparison nature in decision-making when there are incentives as shown in Figure 5.2. In addition to that, it can also help us incorporate robustness into the model learning to be resilient against noise, i.e., with the max-margin objective [21] and the introduction of the violation parameters ($\xi$) to prevent overfitting of the model, especially

when data is limited [35]. The max-margin learning also has very good interpretability, which is very important for understanding user behavior. There are multiple solvers available for this convex optimization problem. In our work, we use `CVXR` for estimating the model [44, 43].

The constraint structure in Eq.(5.7) is almost the same as that in the SVM formulations [26]. The difference between the two here lies on the fact that, if we rewrite the LDT formulation in the standard form of SVM, we will have a varying offset in the constraints, i.e., here, because $r_j$ varies from case to case. In summary, the max-margin formulation in Eq.(5.7) effectively addresses the computational issues in our graphic model that have a unique mechanism of the latent variable. It does not demand additional assumptions in variable distribution or conditional probabilities. Furthermore, the maximized soft margin is capable of obtaining stable estimations with the existence of errors in data. The combination of the advantages from both the graphical model and the computationally efficient max-margin algorithm makes LDT model suitable and effective for modeling the interactions.

## *5.5 Simulation Studies*

We then evaluate the performances of the proposed LDT model, and examine our hypothesis about the data-generating mechanism that is beyond the user behavior data to explain the counter-intuitive results as mentioned in Section 5.2. Our approach is to design an experiment that generates data by the data-generating mechanism as shown in Figure 5.2 and compare the performances of LDT with the logit models.

### *5.5.1 Data Generation*

The simulation design follows the data-generating mechanism described in Section 5.3 where each alternative consists of a set of explanatory attributes $\boldsymbol{x}$ and a certain amount of reward $r$, and the users will make the decision based on his/her own preferences $\boldsymbol{\alpha}$ towards the attributes. To account for the various complexities that can be encountered in real world, our simulation experiments include several different aspects.

First, in real-world decision behaviors, some attributes are usually disliked (or always

welcomed) by the users, e.g., in transportation, changing of scheduled time and extra toll cost can be regard as barrier attributes (commuters usually do not like it), while time or fuel saving can be encouraging attributes which most people would prefer [13, 14]. This characteristic can be reflected in the parameters $\boldsymbol{\alpha}$ as always being positive or negative. For example, we design three attributes in the following study where two of them are barriers $(x_1, x_2)$ and another is an encouraging attribute $x_3$, and the corresponding parameters will follow:

$$\boldsymbol{\alpha} = [\alpha_1 \geq 0,\ \alpha_2 \geq 0,\ \alpha_3 \leq 0]^\top. \tag{5.8}$$

This constraint also provides a perspective for evaluating the interpretability of the model, which is further discussed in performance metrics.

Another layer of complexity is the heterogeneity or diverse cohorts in user preferences, i.e., diverse personal preferences for $N$ different users $(\{\alpha_0^{(i)}, \boldsymbol{\alpha}^{(i}\},\ n = 1, \ldots, N)$. Multiple methods can achieve this setting. In our work, we utilize the Gaussian Mixture Distribution to generate $\boldsymbol{\alpha}^{(i)}$. To be specific, $\boldsymbol{\alpha} \sim N(\boldsymbol{\mu}_{\boldsymbol{\alpha}}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}})$ where $\boldsymbol{\mu}_{\boldsymbol{\alpha}}$ has multiple choices, for example:

$$\boldsymbol{\mu}_{\boldsymbol{\alpha}}^{(1)} = [1, 2, -2]^\top,\ \boldsymbol{\mu}_{\boldsymbol{\alpha}}^{(2)} = [2, 1, -2]^\top,\ \boldsymbol{\mu}_{\boldsymbol{\alpha}}^{(3)} = [2, 2, -1]^\top.$$

Other choices will also work as well. After we have obtained all the $\{\alpha_0^{(i)}, \boldsymbol{\alpha}^{(i)}\}$ from these distributions, we truncate the $\boldsymbol{\alpha}$ that does not fit the condition outlined in Eq.(5.8) to be 0. With the simulated personal preferences for a user, we further generate the alternatives for the user to choose. An alternative is defined by the attributes $(x_1, x_2, x_3)$, which we generate from uniform distributions, i.e., $x_p \sim Unif(0, 30), p = 1, 2, 3$.

The reward mechanism, i.e., the strategy to assign $r$ for the promotions may differ in different apps or systems, or with different levels of knowledge about the users. To determine the rewards for the alternatives, we design three different reward approaches as follows.

- **Random Reward approach** It randomly gives out rewards regardless of the attributes of the choices. This may represent the situations where very few knowledge about user preferences are known so that the rewards are determined randomly. We

generate it using normal distribution where the mean is equal to the average of all generated decision thresholds. The variance of the normal distribution is the same as that of all generated decision thresholds. In this way the overall acceptance rate for the promotions would be close to 0.5, which will ensure we have a balanced dataset, i.e., about half of the promotions would be accepted by users. Based on the aforementioned approach in generating the data, the random rewards are generated via $r \sim N(35, 45^2)$.

- **Contribution Reward approach** Although users preferences vary and are unknown, the "contribution" of any alternative to the transportation system can be determined. For example, we can generate more reward points to those who leave home 20 minutes earlier than those who leave home 2 minutes earlier. Specifically, here we can define $r = 10 + 1.67x_1 + 1.67x_2 - 1.67x_3 + \epsilon_c$ with $\epsilon_c \sim N(0, 5^2)$ to ensure we have a balanced dataset (for the same reason explained in the random reward approach).

- **Predictive Reward approach** This reward mechanism builds on the assumption that the app system has obtained adequate knowledge about every user's preferences and can assign rewards that equal the decision threshold. Here, we design $r_P = \alpha_0^{(i)} + \boldsymbol{\alpha}^{(i)\top}\boldsymbol{x} + \epsilon_P$ for any user $i$. The random term $\epsilon_P \sim N(0, 5^2)$ is used here for the same reason mentioned in the contribution reward approach.

### *5.5.2 Performance metrics*

We will compare the proposed LDT model with MLM, since both build models at the individual level. A population-level logit model is also estimated to provide another baseline for the potential interest of readers. As all the parameters involved in our models are numerical, we use the root of mean squared error ($RMSE$) to evaluate estimation performances of the models. For example, to evaluate how well LDT estimates the latent decision threshold ($D$), the $RMSE$ is defined as:

$$RMSE = \sqrt{\frac{1}{N \times T} \sum_{i,j} (D_j^{(i)} - \hat{D}_j^{(i)})^2}, \tag{5.9}$$

where $i$ refers to the $i$-th user and $j$ refers to the $j$-th decision made by this user. Smaller $RMSE$ indicates that the model can retrieve the latent variable (decision thershold, $D$) better, in other words, the model performs better. For logit models, we estimate the reward where a user will accept the promotion (decision threshold, $D$), i.e., by solving $r$ for equation $Pr(y = 1) = Pr(y = -1) = 0.5$ with estimated coefficients. This seems similar to the concept "tipping point" used in the LDT model, but it is worthy of pointing out that the hypothesized data-generating mechanisms of the two models are quite different as shown in Figure 5.1 and Figure 5.2. Besides $RMSE$, we also use the mean absolute error ($MAE$) and mean absolute percentage error ($MAPE$) to evaluate how well the models can estimate the latent variables and user preferences. For example, for a parameter $\alpha$, $MAE$ and $MAPE$ are defined as follows:

$$MAE = \frac{1}{N}\sum_{n} |\alpha^{(i)} - \hat{\alpha}^{(i)}|, \quad MAPE = \frac{1}{N}\sum_{n} \frac{|\alpha^{(i)} - \hat{\alpha}^{(i)}|}{|\alpha^{(i)}|}. \tag{5.10}$$

In order to have a fair comparison between LDT and the logit models, we normalize $\hat{\boldsymbol{\beta}}$'s by $\hat{\beta}_r$ so that it also represents the "cost" of change in terms of reward.

On the other hand, as shown in Table 5.2, for this personalized TDM system where users interact with the system, the logit models meet difficulties in interpreting the estimated coefficients, i.e., a large proportion of them are counter-intuitive. We also evaluate this aspect of interpretability in our study using the sign of the estimated coefficients, i.e., as outlined in Eq.(5.8). If $\hat{\boldsymbol{\alpha}}$ shows different signs as in Eq.(5.8), it is a sign error. For each element in $\hat{\boldsymbol{\alpha}}$, we calculate the fraction of sign errors (sign error rate, $SER$). Note that logit models also have a parameter for the reward, $\beta_r$, which should be positive [13, 14]. Thus, we also calculate the $SER$ for the reward coefficient for the logit models.

Finally, since the decision-making behavior is cast as a binary classification form, for each learned model, we also report the classification accuracy, recall, and precision on the testing set. Classification accuracy is an overall evaluation, which is defined as the rate of correct classifications. Recall and precision represent two different aspects of the accuracy. Recall, also known as sensitivity, indicates how well the model can detect the positive. It is defined

as the fraction of true positives (actually positive and also classified as positive) among all those cases which are actually positive. Precision represents the accuracy within positive classifications. It is defined as the fraction of true positives among those cases which are classified as positive.

All the experiments are conducted on R (version 4.0.2) on the platform of x86_64-apple-darwin17.0 (64-bit) under macOS 10.15.6 (2.2 GHz 6-Core Intel Core i7, 6 GB 2400 MHz DDR4). The run times for the simulation experiments are reported in the results as well, i.e., in Tables 5.5 to 5.7. Our code is publicly available on Github repository (https://github.com/feng-jings/LDTmodel).

### *5.5.3 Results*

For each reward approach, we run the simulation experiment including the data generation and model fitting for 100 replicates. The overall performance over the 100 replicates is reported using the average of the aforementioned performance metrics. Simulation results on $N = 1000$ users are shown in Tables 5.5 to 5.7 that correspond to the three reward approaches, respectively.

An overall observation is that the proposed LDT model outperforms MLM in parameter estimation (reflected by $RMSE$, $MAE$ and $MAPE$) and interpretability (reflected by $SER$). Both models have similar classification performances (reflected by classification accuracy, recall and precision), and are superior than the population-level logit model. Specifically, from the perspective of coefficient estimation, the estimation errors of LDT model is smaller than that of MLM. The $SER$ also shows MLM often lead to more counter-intuitive results. Among the three reward approaches, the mixed logit model performs worse under the predictive reward approach case than the other two in terms of the coefficient estimation and interpretability.

It is worthy of pointing out that, when normalizing the parameters and calculating the tipping points (decision threshold) for MLM, it could suffer from numerical instability because of very low $\hat{\beta}_r$, so that in deriving the statistics reported in Tables 5.5 to 5.7 we have

Table 5.5: Model performances of Logit, MLM, and LDT models, over 100 replicates of simulation using random reward approach

| | Logit | MLM | LDT |
|---|---|---|---|
| Time (secs) | **0.8754** | 191.5911 | 602.3220 |
| Testing Classification Accuracy | 0.7908 | 0.8862 | **0.8909** |
| Recall | 0.8044 | 0.8820 | **0.8918** |
| Precision | 0.8068 | 0.8811 | **0.8942** |
| RMSE: Decision Threshold | 41.7415 | 38.4610 | **16.1519** |
| RMSE: Coefficient #1 | 1.6482 | 1.3611 | **0.9928** |
| RMSE: Coefficient #2 | 1.6436 | 1.3572 | **0.9930** |
| RMSE: Coefficient #3 | 1.6448 | 1.3573 | **1.0669** |
| MAE: Decision Threshold | 34.8529 | 31.8577 | **12.9827** |
| MAE: Coefficient #1 | 1.4157 | 1.1649 | **0.8230** |
| MAE: Coefficient #2 | 1.4154 | 1.1647 | **0.8264** |
| MAE: Coefficient #3 | 1.4197 | 1.1590 | **0.8720** |
| MAPE: Decision Threshold | 0.9307 | 0.8277 | **0.4509** |
| MAPE: Coefficient #1 | 0.9086 | 0.7341 | **0.5533** |
| MAPE: Coefficient #2 | 0.9086 | 0.7351 | **0.5561** |
| MAPE: Coefficient #3 | 0.9078 | 0.7325 | **0.5699** |
| SER: Coefficient #1 | - | 0.1188 | **0.1090** |
| SER: Coefficient #2 | - | 0.1193 | **0.1070** |
| SER: Coefficient #3 | - | 0.1210 | **0.1100** |
| SER: Coefficient Reward | - | 0.0260 | - |

to drop some experiments that have enormously large metrics. In other words, the $RMSE$, $MAE$ and $MAPE$ values of the logit models shown in Tables 5.5 to 5.7 are actually underestimated (some will be over $10^4$).

Table 5.6: Model performances of Logit, MLM, and LDT models, over 100 replicates of simulation using contribution reward approach

| | Logit | MLM | LDT |
|---|---|---|---|
| Time (secs) | **0.8014** | 167.1454 | 621.2343 |
| Testing Classification Accuracy | 0.5699 | 0.8797 | **0.8869** |
| Recall | 0.6472 | 0.8858 | **0.8924** |
| Precision | 0.5831 | 0.8833 | **0.8945** |
| RMSE: Decision Threshold | 41.7543 | 38.1875 | **15.5102** |
| RMSE: Coefficient #1 | 1.6493 | 1.3339 | **0.8838** |
| RMSE: Coefficient #2 | 1.6444 | 1.3248 | **0.8895** |
| RMSE: Coefficient #3 | 1.6480 | 1.3238 | **0.9383** |
| MAE: Decision Threshold | 34.8650 | 31.6239 | **11.9106** |
| MAE: Coefficient #1 | 1.4168 | 1.1471 | **0.6950** |
| MAE: Coefficient #2 | 1.4162 | 1.1464 | **0.6993** |
| MAE: Coefficient #3 | 1.4161 | 1.1345 | **0.7034** |
| MAPE: Decision Threshold | 0.9313 | 0.8236 | **0.4230** |
| MAPE: Coefficient #1 | 0.9097 | 0.7215 | **0.5100** |
| MAPE: Coefficient #2 | 0.9093 | 0.7256 | **0.5110** |
| MAPE: Coefficient #3 | 0.9110 | 0.7250 | **0.4989** |
| SER: Coefficient #1 | - | 0.0817 | **0.0615** |
| SER: Coefficient #2 | - | 0.0753 | **0.0613** |
| SER: Coefficient #3 | - | 0.0806 | **0.0614** |
| SER: Coefficient Reward | - | 0.0401 | - |

The observations aforementioned are consistent with the analysis presented in Section 5.2. According to Lemma 5.2, when variables are correlated, i.e., the reward is related to other attributes, the estimated coefficients by the logit and mixed logit models may not accurately

Table 5.7: Model performances of Logit, MLM, and LDT models, over 100 replicates of simulation using predictive reward approach

| | Logit | MLM | LDT |
|---|---|---|---|
| Time (secs) | **0.8084** | 176.4998 | 610.6803 |
| Testing Classification Accuracy | 0.6296 | 0.6581 | **0.7566** |
| Recall | 0.7846 | 0.7639 | **0.7927** |
| Precision | 0.6558 | 0.6905 | **0.7947** |
| RMSE: Decision Threshold | 43.6269 | 42.9456 | **4.3299** |
| RMSE: Coefficient #1 | 1.7137 | 1.6719 | **0.2380** |
| RMSE: Coefficient #2 | 1.7086 | 1.6667 | **0.2387** |
| RMSE: Coefficient #3 | 1.6988 | 1.6521 | **0.2948** |
| MAE: Decision Threshold | 36.6210 | 35.9834 | **3.3852** |
| MAE: Coefficient #1 | 1.4790 | 1.4397 | **0.1918** |
| MAE: Coefficient #2 | 1.4787 | 1.4396 | **0.1929** |
| MAE: Coefficient #3 | 1.4655 | 1.4213 | **0.2290** |
| MAPE: Decision Threshold | 0.9968 | 0.9736 | **0.1441** |
| MAPE: Coefficient #1 | 0.9795 | 0.9424 | **0.1367** |
| MAPE: Coefficient #2 | 0.9790 | 0.9424 | **0.1382** |
| MAPE: Coefficient #3 | 0.9661 | 0.9373 | **0.1638** |
| SER: Coefficient #1 | - | 0.0635 | **0.0556** |
| SER: Coefficient #2 | - | 0.0576 | **0.0537** |
| SER: Coefficient #3 | - | 0.0621 | **0.0451** |
| SER: Coefficient Reward | - | 0.3327 | - |

report the true preference parameters as shown by the high estimation errors measured by $RMSE$, $MAE$, $MAPE$, and suffer from an interpretability problem presented by $SER$. However, although its estimated parameters may be inaccurate individually, the estimated

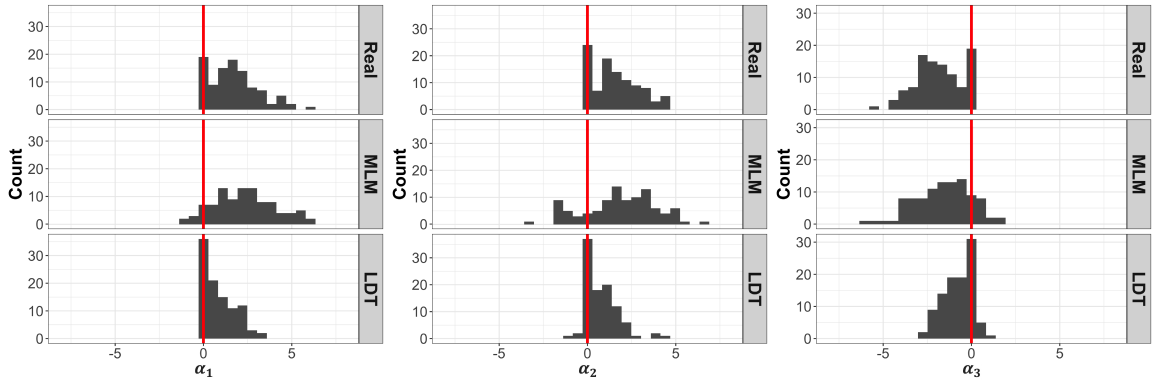


Figure 5.3: Coefficient estimation performances of MLM and LDT model, on the simulated dataset using random reward approach.

model itself may still lead to the same utility so that the classification performances are still comparable. This observation is further reinforced by the experiment using the predictive reward approach (i.e., that computes the reward based on each user's own preferences, so it holds the highest correlation of the three reward approaches) which shows the model performances of MLM under this approach is the worst according to Tables 5.5 to 5.7.

A closer look at the coefficients estimated by MLM and LDT models provide an additional explanation of the model performances. Figure 5.3 illustrates the details of coefficient estimations of MLM and LDT model using random reward approach (in one simulation replicate). The extreme values from unstable estimates of the MLM model have been eliminated for better presentation. Note that, for reason that has been explained in Section 5.3 (i.e., the paragraph below Eq.(5.5)), in order to be comparable with the coefficients of the LDT model, the coefficients in MLM need to be flipped and normalized (by $\hat{\beta}_r^{(i)}$). The red lines refer to zero. Compared with the real values on the top, it can be observed that besides the higher fraction of sign violation, the estimations from MLM are more dispersed.

Figure 5.4 further shows how well the two models perform on the estimation of the latent decision threshold ($D$) using predictive reward approach. The first two panels correspond to MLM, where the middle panel zooms in the shadowed area of the left panel to give a

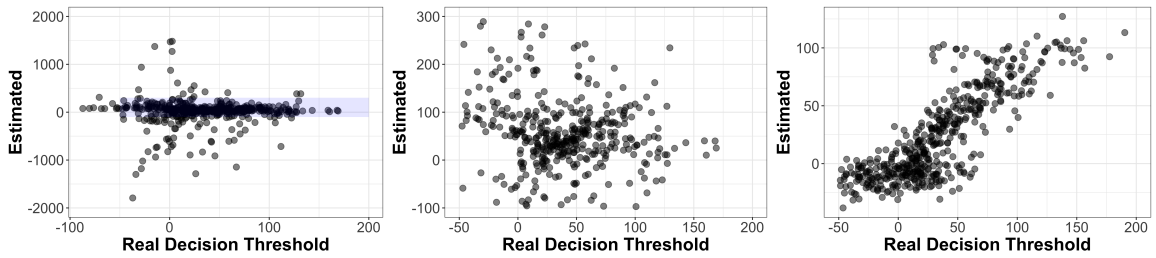


Figure 5.4: Latent variable (the decision threshold) estimation performances of MLM and LDT model, on the simulated dataset using predictive reward approach.

better view of how the majority of the results of the MLM estimation deviates from the true model. The right panel corresponds to the LDT model. It can be seen that the proposed LDT method could yield a nice estimation of the latent decision thresholds ($D$) of the alternatives. It implies that the LDT model is a better choice in explaining user decision behavior and has promising potential to be used as the analytic foundation where personalized promotion algorithms could be developed.

### *5.6 Real-World Case Study*

In this section, we apply our new LDT model on the real-world dataset depicted in Section 2.5 and Section 5.2. After proper cleaning of the data, it consists of 828 respondents and each of them was asked to make 13 rounds of decisions between two alternatives, i.e., whether accept a promotion or stick to the original travel plan [132]. There were 387 female respondents and 441 male respondents. Commuters with different daily commuting times are investigated (range from 10 to 60 minutes) and their average daily commuting time was 31.58 minutes [132]. Promotions for a user are designed based on the user's self-reported background. The attributes that characterize the alternatives include departure time change, i.e., either several minutes earlier ($x_{SDE}$) or several minutes later ($x_{SDL}$), and travel time saving ($x_{TTS}$). The system will determine the amount of reward points $r$ needed for the potential acceptance of the promoted alternative to encourage the user to accept the choice. For each user, we use the first 10 rounds as the training data and the last 3 as the testing data. Among the 828

individuals, there are 174 of them who accepted all the first 10 promotions or rejected all. These users are also excluded from our analysis.

### *5.6.1 Results of the logit and LDT Models*

Table 5.8 shows the results of the population-level logit model, MLM, and the LDT model. From the perspective of classification accuracy, MLM and the LDT model are similar. The LDT model slightly outperforms MLM. But in terms of interpretability, our LDT method is much better. For example, based on MLM there were 38.84% of the users who do not want or even dislike rewards (i.e., $\hat{\beta}_r < 0$). This is counter-intuitive and implies that more rewards only discourage users in accepting promotions. Equally puzzled is that based on MLM there were 21.25% of the users who do not like travel time saving (i.e., $\hat{\beta}_{TTS} < 0$). The LDT model also shows that some users had this counter-intuitive signs, but the percentage is much smaller, later, we will dive in this paradox and identify the peculiar data characteristics of this dataset that is probably responsible for why even LDT also shows a slight percentage of counter-intuitive result. Last but not least, the result shows that the standard deviations of the estimated parameters based on the LDT model are much smaller than those of MLM. This is consistent with our observation in Figure 5.3 that the coefficients estimated by MLM are more dispersed.

Note here that the metric $SER$ (sign error rate) does not necessarily mean that the sign of the estimated coefficient is wrong, because the underlying true preferences are unknown. It indicates results that are counter-intuitive, and points out directions for more investigations.

### *5.6.2 Discover Behavior Patterns*

To show why for some users even the LDT model shows counter-intuitive signs, it is worthy of examining their raw data. Tables 5.9 and 5.10 are data from two users.

For User 625, we notice that $\hat{\alpha}_{SDE} < 0$, indicating that the user preferred departing earlier. It is easy to see why the LDT model learned a negative value of $\hat{\alpha}_{SDE}$, i.e., this user only rejected promotion No.2 and accepted all the others. By comparing this rejected

Table 5.8: Model performances of Logit, MLM, and LDT models, on the smart TDM system data.

| | Logit | MLM | LDT |
|---|---|---|---|
| Testing Classification Accuracy | 0.6677 | 0.7834 | **0.7920** |
| Recall | 0.8152 | 0.8389 | **0.8640** |
| Precision | 0.7329 | 0.8300 | **0.8392** |
| SER: Coefficient SDE | - | 0.1581 | **0.1422** |
| SER: Coefficient SDL | - | **0.0176** | 0.1314 |
| SER: Coefficient TTS | - | 0.2125 | **0.1315** |
| SER: Coefficient Reward | - | 0.3884 | – |
| Standard Deviation: Intercept | - | 98.9605 | **24.3670** |
| Standard Deviation: Coefficient SDE | - | 7.6052 | **1.0097** |
| Standard Deviation: Coefficient SDL | - | 7.9815 | **1.7143** |
| Standard Deviation: Coefficient TTS | - | 17.6733 | **2.0290** |

Table 5.9: The answers of User ID 625 (negative $\hat{\alpha}_{SDE}$)

| Question No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Delay early (min) | 30 | 0 | 10 | 0 | 0 | 0 | 0 | 60 | 0 | 30 | 60 | 30 | 10 |
| Delay late (min) | 0 | 30 | 0 | 30 | 10 | 30 | 10 | 0 | 30 | 0 | 0 | 0 | 0 |
| Time save (min) | 2 | 2 | 2 | 5 | 2 | 2 | 5 | 2 | 5 | 2 | 5 | 5 | 2 |
| Reward (points) | 20 | 20 | 20 | 20 | 20 | 40 | 20 | 20 | 40 | 10 | 20 | 20 | 20 |
| Decision | 1 | -1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

promotion with choices No.1 and No.8, we can see that they all offered the same time saving and reward points, and the only difference was the change of departure time. The respondent rejected No.2 that asked to departure later but accepted the other two which required departing earlier. Therefore, it is reasonable that $\hat{\alpha}_{SDE}$ is negative in the estimated LDT

Table 5.10: The answers of User ID 2078 (negative $\hat{\alpha}_{SDL}$ and positive $\hat{\alpha}_{TTS}$)

| Question No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Delay early (min) | 30 | 0 | 30 | 10 | 10 | 10 | 30 | 30 | 0 | 0 | 0 | 10 | 30 |
| Delay late (min) | 0 | 30 | 0 | 0 | 0 | 0 | 0 | 0 | 30 | 10 | 10 | 0 | 0 |
| Time save (min) | 5 | 5 | 5 | 5 | 5 | 10 | 10 | 10 | 10 | 5 | 5 | 5 | 2 |
| Reward (points) | 20 | 20 | 30 | 20 | 30 | 70 | 20 | 30 | 20 | 20 | 20 | 80 | 30 |
| Decision | -1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |

model, although it is unknown if the user behaved rationally when making these decisions. This is one of the obstacles of small sample size that the estimation may largely dependent on only one or few observations.

Similarly, User 2078 accepted only one promotion which is No.2. It is interesting to notice that the same user rejected No.9 which was the same as No.2 except that No.9 had more time saving. This is the cause of the counter-intuitive value of $\hat{\alpha}_{TTS}$, i.e., which turned out to be negative. Further, if we compare No.2 with No.10, we can see that the user rejected the one requiring less schedule delay late and this might be the cause of a negative $\hat{\alpha}_{SDL}$. Having seen these patterns, we conclude that these unexpected signs were largely caused by the irrational behavior of the users.

Among all the 654 investigated users, User 2361 is the only one whose coefficients are all counter-intuitive based on the LDT model. Table 5.11 shows the data of this user. It can be seen that this user clearly followed a pattern, that was to accept the choice when the given reward is higher than the choice before, i.e., for No.3 and No.4, he/she accepted them because they offered higher reward points than the first two rounds. Then for No.5, however, even though it was the same as the second promotion except for less departure time change, he/she rejected it because he/she had seen higher reward points (20 and 30). Similar behavior can be observed in later decisions. The user accepted the ones offering 40, 50 and 60 reward points sequentially and rejected those with only 10 points. Thus, the learned LDT

Table 5.11: The answers of User ID 2361 (the only user with all three coefficients counter-intuitive))

| Question No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Depart early (min) | 30 | 0 | 30 | 30 | 0 | 30 | 30 | 30 | 0 | 30 | 60 | 60 | 30 |
| Depart late (min) | 0 | 30 | 0 | 0 | 10 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 |
| Time save (min) | 10 | 10 | 10 | 10 | 10 | 10 | 20 | 10 | 20 | 10 | 10 | 10 | 10 |
| Reward (points) | 10 | 10 | 20 | 30 | 10 | 40 | 10 | 50 | 10 | 60 | 40 | 50 | 20 |
| Decision | 1 | 1 | 1 | 1 | -1 | 1 | -1 | 1 | -1 | 1 | -1 | 1 | 1 |

model, while showing counter-intuitive results, captured what was in the data.

## *5.7 Conclusion*

In this chapter, we propose the Latent Decision Threshold (LDT) model to provide a fair characterization of the user-system interaction process and the decision-making behavior by the user in the new environment of personalized interactive systems. The LDT model is a novel graphical model, which depicts the fact that the system is designed to interfere with user behaviors. We develop an efficient learning algorithm based on an interesting connection between the graphical model with max-margin learning. Extensive simulation studies and a real-world application show that the LDT model outperforms the logit models in both model estimation and interpretability.

# Chapter 6
# CONCLUSIONS AND FUTURE WORKS

This dissertation focus on two major data analytic challenges faced by emerging personalized apps in learning user behaviors. Innovative statistical and optimization models are proposed for each aspects.

## *6.1 Personalized Modeling*

A learning framework is proposed for modeling user behaviors at the individual-level, called Logistic Collaborative Model (LogCM), based on the Random Utility Maximization (RUM) theory and the Collaborative Learning Framework. LogCM is capable of learning distinct individual behavioral models from data, even when each individual's data is limited. Further, on the basis of LogCM framework, we extend it in several different ways to better fit specific real-world applications. We develop LogSCM for incorporating similarity information and enhancing the model performance. We propose LogPCM as a knowledge discovery tool to reveal possible hidden canonical structures in heterogeneous populations. For time-varying preferences, LogCM-T is capable of modeling preference changes. Extensive numerical studies on simulation and real-world application show promising performances of all these proposed models.

Another strand of personalized modeling in our research is handling the complex composition of a large heterogeneous population. Due to the complex heterogeneity, a single canonical model layer may not sufficiently express the population's collaborative structure. Therefore, we extend the collaborative learning into a multi-layer model and propose the Hierarchical Collaborative Model (HCM), where multiple levels of canonical models with different degrees of detail can be learned simultaneously. The proposed Contextual HCM

can reflect the population's compositional structure so that characteristic information can help enhance personalized modeling.

### *6.2 User Behavior Modeling*

The interactive nature of personalized systems/apps raises an unprecedented challenge in user behavior modeling, which is that the data collection procedure also interfere with the user behavior. We propose a graphical model called Latent Decision Threshold (LDT) model to characterize the user-system interaction process and the decision-making behavior of the users in this new environment. The graphical model deconstructs the entanglement with a carefully designed model structure to mimic such interaction mechanism. Further analysis of its structure leads to an efficient learning algorithm with max-margin learning. Extensive simulation studies and a real-world application with a new reward-based smart TDM system illustrate the effectiveness and good interpretability of the LDT model.

### *6.3 Future Research: Collaborative Latent Decision Threshold Model (C-LDT)*

The developed models in this thesis have separately addressed the challenges in personalized modeling and user behavior modeling. In reality, it could be the case that both challenges are present. In other words, we will need to learn users' decision-making behaviors in an interactive system, with limited individual data and the system intervention as well.

In facing such a challenge, we can add collaborative structures to the LDT model, so that individual models are not learned independently but related to each other with a shared set of canonical models. To be concrete, with the notation defined in Chapters 2 and 5,

$$\begin{aligned} \text{Latent Decision Threshold: } & y(r - D) \geq 0, \\ & D(\boldsymbol{x}) = \boldsymbol{\alpha}^\top \boldsymbol{x} + \epsilon; \\ \text{Collaborative Structure: } & \boldsymbol{\alpha}_i = \mathbf{Q}\boldsymbol{c}_i. \end{aligned} \tag{6.1}$$

Consolidating these, and we can have the following aggregated inequality of Collaborative

Latent Decision Threshold (C-LDT).

$$y_{ij}(r_{ij} - \boldsymbol{x}_{ij}^{\top}\mathbf{Q}\boldsymbol{c}_i) \geq 0. \tag{6.2}$$

Here, $\boldsymbol{x}_{ij}, r_{ij}, y_{ij}$ are attribute vector, reward, and the outcome of the $j$-th decision of individual $i$ respectively, same as in LDT model. $\mathbf{Q}$ is the shared canonical matrix and $\boldsymbol{c}_i$ is the membership vector of individual $i$, same as in collaborative models. Due to the structure of collaborative learning, the max-margin formulation is not an efficient way to estimate the parameters anymore with the existence of quadruple items. To develop formulation and learning algorithm for C-LDT, we recommend that future studies apply loss functions to Eq.(6.2) such as huber loss and exponential loss [61, 98, 104] can derive the objectives to optimize.

The C-LDT model can learn LDT model for every individual in a heterogeneous population, even when each individual's data is limited. With the good interpretability granted by LDT model, the C-LDT would be an essential block for such interactive personalized systems or apps [132, 6] to coordinate and change user behavior.

### *6.4 Future research: Recommendation System with Preference Updater*

Recommendation systems are usually developed to deal with information overload and provide personalized recommendations, content and services to users in e-commerce domain. They have proven to be valuable means for online users because they can identify a subset of items from a much larger set, i.e., item pool, that best matches a user's interest. Typically, a recommendeation system employs relevancy or similarity based (similarity among users or items) techniques such as collaborative filtering and content-based filtering [1, 96]. These approaches are successful but suffer from a lack of theoretical understanding of the behavioral process that led to a particular choice. With more individual data becoming available, the ability of discrete choice models to predict individual choices has attracted interest in recommendation systems [109, 19, 64, 95]. The application of choice models that can account well for consumer heterogeneity enables the recommendation systems to utilize predicted

choice probability or utility to generate recommending list. Unlike the relevancy metrics, it directly measures the quality or result of the recommendation and is easier to interpret. On the other hand, it provides a unified approach to achieve both relevancy and diversity in recommendation which is also important for good recommendation system [97]. Widely used discrete choice models include multinomial logit, nest logit, and logit mixture models, and the problem of recommendation is formulated as a integer optimization or assortment optimization problem within the item list [64, 109, 68].

With our collaborative learning framework and latent decision threshold model, the heterogeneity in user preferences is explicitly presented and modeled so that it suits for recommendation systems. With the well-studied individual models, it could be easier to find the "best" next recommendation in terms of most revenue gain, or highest acceptance possibility [60, 74, 109, 112].

On the other hand, as the user-system interaction is a dynamic progress, determining the next recommendation aiming to improve the accuracy of individual models may also be beneficial. The model accuracy may be limited by the available data size, especially when a user is new, whose preference is shifting. Recommendations based on inaccurate model may lead to inefficiency. Recommendation systems built on user preferences should have the ability to update their user preference model as more data becomes available [109]. Bayesian sequential experimental design could be useful to solve this problem that is the state-of-art method to measure the parameter estimation uncertainties and can generate next-step experiment to reduce the uncertainty under some criteria such as D-optimality [9, 119, 126].

Finally, the two targets of the next recommendation intrinsically raise a trade-off: whether to exploit the existing estimated model to maximize revenue gain, or to explore more about the user preference to update the individual model. A multi-armed bandit approach can be used to deal with such trade-offs [40, 16].

# BIBLIOGRAPHY

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[2] Jong-Hoon Ahn, Seungjin Choi, and Jong-Hoon Oh. A multiplicative up-propagation algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 3. ACM, 2004.

[3] Christopher P Ames, Justin S Smith, Ferran Pellisé, Michael Kelly, Ahmet Alanay, Emre Acaroglu, Francisco Javier Sánchez Pérez-Grueso, Frank Kleinstück, Ibrahim Obeid, Alba Vila-Casademunt, et al. Artificial intelligence based hierarchical clustering of patient types and intervention categories in adult spinal deformity surgery: towards a new classification scheme that predicts quality and value. *Spine*, 44(13):915–926, 2019.

[4] Ali Arian, Alireza Ermagun, Xiaoyu Zhu, and Yi-Chang Chiu. An empirical investigation of the reward incentive and trip purposes on departure time behavior change. In *Advances in Transport Policy and Planning*, volume 1, pages 145–167. Elsevier, 2018.

[5] Hossein Azari, David Parks, and Lirong Xia. Random utility theory for social choice. In *Advances in Neural Information Processing Systems*, pages 126–134, 2012.

[6] Carlos Lima Azevedo, Ravi Seshadri, Song Gao, Bilge Atasoy, Arun Prakash Akkinepally, Eleni Christofa, Fang Zhao, Jessika Trancik, and Moshe Ben-Akiva. Tripod: sustainable travel incentives with prediction, optimization, and personalization. In *the 97th Annual Meeting of Transportation Research Board*, 2018.

[7] Sebastian Bamberg, Daniel Rölle, and Christoph Weber. Does habitual car use not lead to more resistance to change of travel mode? *Transportation*, 30(1):97–108, 2003.

[8] Fabian Bastin, Cinzia Cirillo, and Philippe L Toint. Estimating nonparametric random utility models with an application to the value of time in heterogeneous populations. *Transportation science*, 44(4):537–549, 2010.

[9] Felix Becker, Mazen Danaf, Xiang Song, Bilge Atasoy, and Moshe Ben-Akiva. Bayesian estimator for logit mixtures with inter-and intra-consumer heterogeneity. *Transportation Research Part B: Methodological*, 117:1–17, 2018.

[10] Moshe Ben-Akiva, André de Palma, Daniel McFadden, Maya Abou-Zeid, Pierre-André Chiappori, Matthieu de Lapparent, Steven N Durlauf, Mogens Fosgerau, Daisuke Fukuda, Stephane Hess, et al. Process and context in choice models. *Marketing Letters*, 23(2):439–456, 2012.

[11] Moshe Ben-Akiva, Daniel McFadden, Kenneth Train, Joan Walker, Chandra Bhat, Michel Bierlaire, Denis Bolduc, Axel Boersch-Supan, David Brownstone, David S Bunch, et al. Hybrid choice models: Progress and challenges. *Marketing Letters*, 13(3):163–175, 2002.

[12] Moshe E Ben-Akiva, Steven R Lerman, and Steven R Lerman. *Discrete choice analysis: theory and application to travel demand*, volume 9. MIT press, 1985.

[13] Eran Ben-Elia and Dick Ettema. Changing commuters' behavior using rewards: A study of rush-hour avoidance. *Transportation research part F: traffic psychology and behaviour*, 14(5):354–368, 2011.

[14] Eran Ben-Elia and Dick Ettema. Rewarding rush-hour avoidance: A study of commuters' travel behavior. *Transportation Research Part A: Policy and Practice*, 45(7):567–582, 2011.

[15] James R Bettman, Mary Frances Luce, and John W Payne. Constructive consumer choice processes. *Journal of consumer research*, 25(3):187–217, 1998.

[16] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.

[17] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1548–1560, 2010.

[18] Joseph Chee Chang, Nathan Hahn, Adam Perer, and Aniket Kittur. Searchlens: Composing and capturing complex user interests for exploratory search. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 498–509, 2019.

[19] Bassam H Chaptini. *Use of discrete choice models with recommender systems*. PhD thesis, Massachusetts Institute of Technology, 2005.

[20] Yves Chauvin and David E Rumelhart. *Backpropagation: theory, architectures, and applications*. Psychology press, 2013.

[21] Gal Chechik, Geremy Heitz, Gal Elidan, Pieter Abbeel, and Daphne Koller. Max-margin classification of data with absent features. *Journal of Machine Learning Research*, 9(Jan):1–21, 2008.

[22] Cynthia Chen, Jingtao Ma, Yusak Susilo, Yu Liu, and Menglin Wang. The promises of big data and small data for travel behavior (aka human mobility) analysis. *Transportation research part C: emerging technologies*, 68:285–299, 2016.

[23] Di-Rong Chen, Qiang Wu, Yiming Ying, and Ding-Xuan Zhou. Support vector machine soft margin classifiers: error analysis. *Journal of Machine Learning Research*, 5(Sep):1143–1175, 2004.

[24] Caspar G Chorus. A new model of random regret minimization. *European Journal of Transport and Infrastructure Research*, 10(2), 2010.

[25] Caspar G Chorus, John M Rose, and David A Hensher. Regret minimization or utility maximization: it depends on the attribute. *Environment and Planning B: Planning and Design*, 40(1):154–169, 2013.

[26] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[27] Zaixu Cui and Gaolang Gong. The effect of machine learning regression algorithms and sample size on individualized behavioral prediction with functional connectivity features. *Neuroimage*, 178:622–637, 2018.

[28] Arthur P Dempster, Nan M Laird, and Donald 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.

[29] Adele Diederich. Dynamic stochastic models for decision making under time constraints. *Journal of Mathematical Psychology*, 41(3):260–274, 1997.

[30] Chris HQ Ding, Tao Li, and Michael I Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):45–55, 2010.

[31] Wafa Elias and Yoram Shiftan. The influence of individual's risk perception and attitudes on travel behavior. *Transportation research part A: policy and practice*, 46(8):1241–1251, 2012.

[32] Martin S Feldstein. Multicollinearity and the mean square error of alternative estimators. *Econometrica: Journal of the Econometric Society*, pages 337–346, 1973.

[33] Jingshuo Feng, Shuai Huang, and Cynthia Chen. Modeling user interaction with app-based reward system: A graphical model approach integrated with max-margin learning. *Transportation Research Part C: Emerging Technologies*, 120:102814, 2020.

[34] Jingshuo Feng, Xi Zhu, Feilong Wang, Shuai Huang, and Cynthia Chen. A learning framework for personalized random utility maximization (rum) modeling of user behavior. *IEEE Transactions on Automation Science and Engineering*, 2020.

[35] A Floares, MARIUS Ferisgan, DANIELA Onita, ANDREI Ciuparu, G Calin, and F Manolache. The smallest sample size for the desired diagnosis accuracy. *International Journal of Oncology and Cancer Therapy*, 2017.

[36] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[37] Anqi Fu, Balasubramanian Narasimhan, and Stephen Boyd. Cvxr: An r package for disciplined convex optimization. *arXiv preprint arXiv:1711.07582*, 2017.

[38] Andrzej Gałecki and Tomasz Burzykowski. *Linear mixed-effects models using R: A step-by-step approach*. Springer Science & Business Media, 2013.

[39] William V Giannobile, Thomas M Braun, Anna K Caplis, Lynn Doucette-Stamm, Gordon W Duff, and Kenneth S Kornman. Patient stratification for preventive care in dentistry. *Journal of dental research*, 92(8):694–701, 2013.

[40] John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

[41] Genevieve Giuliano. Transportation demand management: promise or panacea? *Journal of the American Planning Association*, 58(3):327–335, 1992.

[42] Donald Goldfarb and Ashok Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical programming*, 27(1):1–33, 1983.

[43] Michael Grant and Stephen Boyd. Cvx: Matlab software for disciplined convex programming, version 2.1, 2014.

[44] Michael C Grant and Stephen P Boyd. Graph implementations for nonsmooth convex programs. In *Recent advances in learning and control*, pages 95–110. Springer, 2008.

[45] William H Greene and David A Hensher. A latent class model for discrete choice analysis: contrasts with mixed logit. *Transportation Research Part B: Methodological*, 37(8):681–698, 2003.

[46] C Angelo Guevara and Stephane Hess. A control-function approach to correct for endogeneity in discrete choice models estimated on sp-off-rp data and contrasts with an earlier fiml approach by train & wilson. *Transportation Research Part B: Methodological*, 123:224–239, 2019.

[47] Cristian Angelo Guevara and Moshe Ben-Akiva. Addressing endogeneity in discrete choice models: Assessing control-function and latent-variable methods. *Choice Modelling: The State-of-the Art and the State-of-practice*, pages 353–371, 2010.

[48] Paulo Guimaraes, Octávio Figueiredo, and Douglas Woodward. Industrial location modeling: Extending the random utility framework. *Journal of Regional Science*, 44(1):1–20, 2004.

[49] Meeghat Habibian and Mohammad Kermanshah. Coping with congestion: Understanding the role of simultaneous transportation demand management policies on commuters. *Transport Policy*, 30:229–237, 2013.

[50] Thomas O Hancock, Stephane Hess, and Charisma F Choudhury. Decision field theory: Improvements to current methodology and comparisons with standard choice modelling techniques. *Transportation Research Part B: Methodological*, 107:18–40, 2018.

[51] Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.

[52] Donald Hedeker. A mixed-effects multinomial logistic regression model. *Statistics in medicine*, 22(9):1433–1446, 2003.

[53] David A Hensher. Stated preference analysis of travel choices: the state of practice. *Transportation*, 21(2):107–133, 1994.

[54] David A Hensher and William H Greene. The mixed logit model: the state of practice. *Transportation*, 30(2):133–176, 2003.

[55] Stephane Hess, Andrew Daly, and Richard Batley. Revisiting consistency with random utility maximisation: theory and implications for practical work. *Theory and Decision*, 84(2):181–204, 2018.

[56] Stephane Hess and Marek Giergiczny. Intra-respondent heterogeneity in a stated choice survey on wetland conservation in belarus: first steps towards creating a link with uncertainty in contingent valuation. *Environmental and Resource Economics*, 60(3):327–347, 2015.

[57] Stephane Hess and Amanda Stathopoulos. A mixed random utility—random regret model linking the choice of decision rule to latent character traits. *Journal of choice modelling*, 9:27–38, 2013.

[58] R Carter Hill and George Judge. Improved prediction in the presence of multicollinearity. *Journal of Econometrics*, 35(1):83–100, 1987.

[59] Fushing Y Hsieh, Daniel A Bloch, and Michael D Larsen. A simple method of sample size calculation for linear and logistic regression. *Statistics in medicine*, 17(14):1623–1634, 1998.

[60] Cheng-Lung Huang and Wei-Liang Huang. Handling sequential pattern decay: Developing a two-stage collaborative recommender system. *Electronic Commerce Research and Applications*, 8(3):117–129, 2009.

[61] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.

[62] Lun-ping Hung. A personalized recommendation system based on product taxonomy for one-to-one marketing online. *Expert systems with applications*, 29(2):383–392, 2005.

[63] Brian M Iacoviello, Joshua R Steinerman, David B Klein, Theodore L Silver, Adam G Berger, Sean X Luo, and Nicholas J Schork. Clickotine, a personalized smartphone app for smoking cessation: initial evaluation. *JMIR mHealth and uHealth*, 5(4):e56, 2017.

[64] Hai Jiang, Xin Qi, and He Sun. Choice-based recommender systems: a unified approach to achieving relevancy and diversity. *Operations Research*, 62(5):973–993, 2014.

[65] Richard Katzev. Car sharing: A new approach to urban transportation problems. *Analyses of social issues and public policy*, 3(1):65–86, 2003.

[66] Jinhee Kim, Soora Rasouli, and Harry JP Timmermans. Investigating heterogeneity in social influence by social distance in car-sharing decisions under uncertainty: A regret-minimizing hybrid choice model framework based on sequential stated adaptation experiments. *Transportation Research Part C: Emerging Technologies*, 85:47–63, 2017.

[67] Ran Kivetz, Oded Netzer, and Rom Schrift. The synthesis of preference: Bridging behavioral decision research and marketing science. *Journal of Consumer Psychology*, 18(3):179–186, 2008.

[68] A Gürhan Kök, Marshall L Fisher, and Ramnath Vaidyanathan. Assortment planning: Review of literature and industry practice. In *Retail supply chain management*, pages 99–153. Springer, 2008.

[69] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[70] Pigi Kouki, Ilias Fountalis, Nikolaos Vasiloglou, Nian Yan, Unaiza Ahsan, Khalifeh Al Jadda, and Huiming Qu. Product collection recommendation in online retail. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 486–490, 2019.

[71] Rico Krueger, Prateek Bansal, Michel Bierlaire, Ricardo A Daziano, and Taha H Rashidi. Variational bayesian inference for mixed logit models with unobserved inter- and intra-individual heterogeneity. *arXiv preprint arXiv:1905.00419*, 2019.

[72] Sumit Kunnumkal. Randomization approaches for network revenue management with customer choice behavior. *Production and Operations Management*, 23(9):1617–1633, 2014.

[73] Sik-Yum Lee and Jian-Qing Shi. Maximum likelihood estimation of two-level latent variable models with mixed continuous and polytomous data. *Biometrics*, 57(3):787–794, 2001.

[74] Jovian Lin, Kazunari Sugiyama, Min-Yen Kan, and Tat-Seng Chua. Addressing cold-start in app recommendation: latent user models constructed from twitter followers. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 283–292, 2013.

[75] Ying Lin, Shuai Huang, Gregory E Simon, and Shan Liu. Analysis of depression trajectory patterns using collaborative learning. *Mathematical biosciences*, 282:191–203, 2016.

[76] Ying Lin, Kaibo Liu, Eunshin Byon, Xiaoning Qian, and Shuai Huang. Domain-knowledge driven cognitive degradation modeling for alzheimer's disease. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 721–729. SIAM, 2015.

[77] Ying Lin, Kaibo Liu, Eunshin Byon, Xiaoning Qian, Shan Liu, and Shuai Huang. A collaborative learning framework for estimating many individualized regression models in a heterogeneous population. *IEEE Transactions on Reliability*, 67(1):328–341, 2018.

[78] Ying Lin, Shan Liu, and Shuai Huang. Selective sensing of a heterogeneous population of units with dynamic health conditions. *IISE Transactions*, 50(12):1076–1088, 2018.

[79] Siwei Lyu and Xin Wang. On algorithms for sparse multi-factor nmf. In *Advances in Neural Information Processing Systems*, pages 602–610, 2013.

[80] Shujie Ma and Jian Huang. A concave pairwise fusion approach to subgroup analysis. *Journal of the American Statistical Association*, 112(517):410–423, 2017.

[81] Inge Mayeres, Sara Ochelen, and Stef Proost. The marginal external costs of urban transport. *Transportation Research Part D: Transport and Environment*, 1(2):111–130, 1996.

[82] Daniel McFadden et al. Conditional logit analysis of qualitative choice behavior. 1973.

[83] Daniel McFadden et al. The revealed preferences of a government bureaucracy: Theory. *Bell Journal of Economics*, 6(2):401–416, 1975.

[84] Michael D Meyer. Demand management as an element of transportation policy: using carrots and sticks to influence travel behavior. *Transportation Research Part A: Policy and Practice*, 33(7-8):575–599, 1999.

[85] John C Milton, Venky N Shankar, and Fred L Mannering. Highway accident severities and the mixed logit model: an exploratory empirical analysis. *Accident Analysis & Prevention*, 40(1):260–266, 2008.

[86] Tom Minka. Algorithms for maximum-likelihood logistic regression. Technical report, Technical report, CMU, Department of Statistics, TR 758, 2001.

[87] Renato DC Monteiro, Ilan Adler, and Mauricio GC Resende. A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension. *Mathematics of Operations Research*, 15(2):191–214, 1990.

[88] Guido Möser and Sebastian Bamberg. The effectiveness of soft transport policy measures: A critical assessment and meta-analysis of empirical evidence. *Journal of Environmental Psychology*, 28(1):10–26, 2008.

[89] Peter Nijkamp and Dani Shefer. Urban transport externalities and pigouvian taxes: A network approach. In *Road Pricing, Traffic Congestion and the Environment*, pages 171–189. Edgar Elgar, 1998.

[90] Sewoong Oh and Devavrat Shah. Learning mixed multinomial logit model from ordinal data. In *Advances in Neural Information Processing Systems*, pages 595–603, 2014.

[91] George R Parsons, Erik C Helm, and Tim Bondelid. Measuring the economic benefits of water quality improvements to recreational users in six northeastern states: an application of the random utility maximization model. *For the EPA Office of Policy Economics and Innovation*, 2003.

[92] Elazar J Pedhazur and Liora Pedhazur Schmelkin. *Measurement, design, and analysis: An integrated approach*. Psychology Press, 2013.

[93] Ludovico Pedullà, Giampaolo Brichetto, Andrea Tacchino, Claudio Vassallo, Paola Zaratin, Mario Alberto Battaglia, Laura Bonzano, and Marco Bove. Adaptive vs. non-adaptive cognitive training by means of a personalized app: a randomized trial in people with multiple sclerosis. *Journal of neuroengineering and rehabilitation*, 13(1):88, 2016.

[94] Sebastian Petry, Claudia Flexeder, and Gerhard Tutz. Pairwise fused lasso. 2011.

[95] Amalia Polydoropoulou and Maria A Lambrou. Development of an e-learning recommender system using discrete choice models and bayesian theory: a pilot case in the

shipping industry. *Security Enhanced Applications for Information Systems*, page 35, 2012.

[96] Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K Lam, Sean M McNee, Joseph A Konstan, and John Riedl. Getting to know you: learning new user preferences in recommender systems. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 127–134, 2002.

[97] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.

[98] Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana, and Alessandro Verri. Are loss functions all the same? *Neural computation*, 16(5):1063–1076, 2004.

[99] Talib Rothengatter. The effects of police surveillance and law enforcement on driver behaviour. *Current Psychological Reviews*, 2(3):349–358, 1982.

[100] Jan Rouwendal and Erik T Verhoef. Basic economic principles of road pricing: From theory to applications. *Transport policy*, 13(2):106–114, 2006.

[101] Sam Roweis and Zoubin Ghahramani. A unifying review of linear gaussian models. *Neural computation*, 11(2):305–345, 1999.

[102] Geertje Schuitema and Linda Steg. The role of revenue use in the acceptability of transport pricing policies. *Transportation Research Part F: Traffic Psychology and Behaviour*, 11(3):221–231, 2008.

[103] Jianzhao Shen and Sujuan Gao. A solution to separation and multicollinearity in multiple logistic regression. *Journal of data science: JDS*, 6(4):515, 2008.

[104] Yi Shen. *Loss functions for binary classification and class probability estimation*. PhD thesis, University of Pennsylvania, 2005.

[105] Delsey M Sherrill, Marilyn L Moy, John J Reilly, and Paolo Bonato. Using hierarchical clustering methods to classify motor activities of copd patients from wearable sensor data. *Journal of NeuroEngineering and Rehabilitation*, 2(1):16, 2005.

[106] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

[107] Brent Smith and Greg Linden. Two decades of recommender systems at amazon. com. *Ieee internet computing*, 21(3):12–18, 2017.

[108] Junbo Son, Qiang Zhou, Shiyu Zhou, Xiaofeng Mao, and Mutasim Salman. Evaluation and comparison of mixed effects model based prognosis for hard failure. *IEEE Transactions on Reliability*, 62(2):379–394, 2013.

[109] Xiang Song, Mazen Danaf, Bilge Atasoy, and Moshe Ben-Akiva. Personalized menu optimization with preference updater: a boston case study. *Transportation Research Record*, 2672(8):599–607, 2018.

[110] Peter R Stopher. Reducing road congestion: a reality check. *Transport Policy*, 11(2):117–131, 2004.

[111] Jimeng Sun, Daby Sow, Jianying Hu, and Shahram Ebadollahi. Localized supervised metric learning on temporal physiological data. In *2010 20th International Conference on Pattern Recognition*, pages 4149–4152. IEEE, 2010.

[112] John K Tarus, Zhendong Niu, and Abdallah Yousif. A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. *Future Generation Computer Systems*, 72:37–48, 2017.

[113] Thomas R Ten Have and A Russell Localio. Empirical bayes estimation of random effects parameters in mixed effects logistic regression models. *Biometrics*, 55(4):1022–1029, 1999.

[114] R Thomas, Ten Have, Allen R Kunselman, Erik P Pulkstenis, and J Richard Landis. Mixed effects logistic regression models for longitudinal binary response data with informative drop-out. *Biometrics*, pages 367–383, 1998.

[115] Mikkel Thorhauge, Elisabetta Cherchi, Joan L Walker, and Jeppe Rich. The role of intention as mediator between latent effects and behavior: application of a hybrid choice model to study departure time choices. *Transportation*, 46(4):1421–1445, 2019.

[116] Kenneth E Train. *Discrete choice methods with simulation*. Cambridge university press, 2009.

[117] George Trigeorgis, Konstantinos Bousmalis, Stefanos Zafeiriou, and Bjoern Schuller. A deep semi-nmf model for learning hidden representations. In *International Conference on Machine Learning*, pages 1692–1700, 2014.

[118] George Trigeorgis, Konstantinos Bousmalis, Stefanos Zafeiriou, and Björn W Schuller. A deep matrix factorization method for learning attribute representations. *IEEE transactions on pattern analysis and machine intelligence*, 39(3):417–429, 2016.

[119] Miikka Väntänen, Joona Vaara, Jukka Aho, Jukka Kemppainen, and Tero Frondelius. Bayesian sequential experimental design for fatigue tests. *Rakenteiden Mekaniikka*, 50(3):201–205, 2017.

[120] Jeroen K Vermunt. Mixed-effects logistic regression models for indirectly observed discrete outcome variables. *Multivariate Behavioral Research*, 40(3):281–301, 2005.

[121] Rosalie Viney, Emily Lancsar, and Jordan Louviere. Discrete choice experiments to measure consumer preferences for health and healthcare. *Expert review of pharmacoeconomics & outcomes research*, 2(4):319–326, 2002.

[122] Fei Wang, Jimeng Sun, Jianying Hu, and Shahram Ebadollahi. imet: interactive metric learning in healthcare applications. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 944–955. SIAM, 2011.

[123] Greg CG Wei and Martin A Tanner. A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85(411):699–704, 1990.

[124] Clifford Winston. On the performance of the us transportation system: Caution ahead. *Journal of Economic Literature*, 51(3):773–824, 2013.

[125] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[126] Cao Xiao, Yan Jin, Ji Liu, Bo Zeng, and Shuai Huang. Optimal expert knowledge elicitation for bayesian network structure identification. *IEEE Transactions on Automation Science and Engineering*, 15(3):1163–1177, 2018.

[127] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. Deep matrix factorization models for recommender systems. In *IJCAI*, pages 3203–3209, 2017.

[128] Handong Zhao, Zhengming Ding, and Yun Fu. Multi-view clustering via deep matrix factorization. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[129] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiusi Chen, and Jun Gao. Atrank: An attention-based user behavior modeling framework for recommendation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[130] Mo Zhou, Yoshimi Fukuoka, Yonatan Mintz, Ken Goldberg, Philip Kaminsky, Elena Flowers, and Anil Aswani. Evaluating machine learning–based automated personalized daily step goals delivered through a mobile phone app: Randomized controlled trial. *JMIR mHealth and uHealth*, 6(1):e28, 2018.

[131] Xi Zhu, Jingshuo Feng, Shuai Huang, and Cynthia Chen. An online updating method for time-varying preference learning. *Transportation Research Part C: Emerging Technologies*, 121:102849, 2020.

[132] Xi Zhu, Feilong Wang, Cynthia Chen, and Derek D Reed. Personalized incentives for promoting sustainable travel behaviors. *Transportation Research Part C: Emerging Technologies*, 113:314–331, 2020.

# Appendix A

## *A.1 Proof to Theorem 2.2*

In the following, we will show the equivalence between the objective function of logistic MEM and the objective function of proposed LogSCM, given the number of latent classes $K$, and the penalty hyperparameter $\lambda$.

**Theorem 2.2.** *The objective function of the LogSCM is equivalent to the objective function of logistic MEM when* $\mathbf{W}$ *is a matrix with all entries being* $1/\lambda N$ *and* $\boldsymbol{\Sigma} = \mathbf{Q}\mathbf{Q}^\top$.

***Proof*** Logistic MEM assumes that for individual $i$, fixed effect $\boldsymbol{\beta}_f$ and random effect $\boldsymbol{\beta}_{ri}$ will together influence the probability of being in a certain category, where the random effect $\boldsymbol{\beta}_{ri}$ follows a multivariate normal distribution $N(\mathbf{0}, \boldsymbol{\Sigma})$.

$$y_{ij} \sim Ber(P(y = 1|\boldsymbol{\beta}_f, \boldsymbol{\beta}_{ri}, \boldsymbol{x}_{ij}));$$

$$P(y = 1|\boldsymbol{\beta}_f, \boldsymbol{\beta}_{ri}, \boldsymbol{x}_{ij}) = \pi_{ij} = logit^{-1}(\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}_f + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}_{ri});$$

$$\boldsymbol{\beta}_{ri} \sim N(\mathbf{0}, \boldsymbol{\Sigma}).$$

where $Ber$ refers to the Bernoulli distribution and $logit$ refers to the logit function.

Based on the conditional distribution of $y_{ij}$, we can derive the likelihood function for logistic MEM as:

$$\begin{aligned} L(\boldsymbol{\beta}, \boldsymbol{\Sigma}) &= \prod_{i=1}^{N} \prod_{j=1}^{n_i} \left\{ [\pi_{ij}]^{y_{ij}} [1 - \pi_{ij}]^{1-y_{ij}} P(\boldsymbol{\beta}_{ri}|\boldsymbol{\Sigma}) \right\} \\ &= \prod_{i=1}^{N} \prod_{j=1}^{n_i} \left\{ [\pi_{ij}]^{y_{ij}} [1 - \pi_{ij}]^{1-y_{ij}} \right. \\ &\qquad \left. \left[ \frac{1}{\sqrt{(2\pi)^p |\boldsymbol{\Sigma}|}} \exp(-\frac{1}{2} \boldsymbol{\beta}_{ri}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\beta}_{ri}) \right] \right\}, \end{aligned}$$

where

$$\pi_{ij} = \frac{\exp(\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}_f + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}_{ri})}{1 + \exp(\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}_f + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}_{ri})}.$$

Use $\boldsymbol{\beta}_i = \boldsymbol{\beta}_f + \boldsymbol{\beta}_{ri}$ to replace $\boldsymbol{\beta}_{ri}$ and the log-likelihood can be derived as:

$$\begin{aligned} l(\boldsymbol{\beta}, \boldsymbol{\Sigma}) =& \sum_{i=1}^{N} \sum_{j=1}^{n_i} \{y_{ij} \log \pi_{ij} + (1 - y_{ij}) \log(1 - \pi_{ij})\} \\ & - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{n_i} (\boldsymbol{\beta}_i - \boldsymbol{\beta}_f)^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{\beta}_i - \boldsymbol{\beta}_f) \\ =& \sum_{i=1}^{N} \sum_{j=1}^{n_i} \left\{ y_{ij} (\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}_i) - \log(1 + \exp(x_{ij}^\top \boldsymbol{\beta}_i)) \right\} \\ & - \frac{1}{2} \sum_{i=1}^{N} n_i (\boldsymbol{\beta}_i - \boldsymbol{\beta}_f)^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{\beta}_i - \boldsymbol{\beta}_f). \end{aligned}$$

If all $n_i$'s are same or we simply weight the log-likelihood with respect to $n_i$, and then flip it to a loss function, we will have:

$$\begin{aligned} \text{Obj}_{MEM} =& \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \left\{ \log(1 + \exp(x_{ij}^\top \boldsymbol{\beta}_i)) - y_{ij} (\boldsymbol{x}_{ij}^\top \boldsymbol{\beta}_i) \right\} \\ & + \frac{1}{2} \sum_{i=1}^{N} (\boldsymbol{\beta}_i - \boldsymbol{\beta}_f)^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{\beta}_i - \boldsymbol{\beta}_f). \end{aligned}$$

Given $K$ and $\lambda$, the objective function we have for LogSCM is

$$\begin{aligned} \text{Obj}_{SCM} =& \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \left\{ \log(1 + \exp(\boldsymbol{x}_{ij}^\top \mathbf{Q} \boldsymbol{c}_i)) - y_{ij} (\boldsymbol{x}_{ij}^\top \mathbf{Q} \boldsymbol{c}_i) \right\} \\ & + \frac{\lambda}{2} \sum_{l,m} \|\boldsymbol{c}_l - \boldsymbol{c}_m\|^2 w_{lm}. \end{aligned}$$

By denoting $\boldsymbol{\beta}_i = \mathbf{Q} \boldsymbol{c}_i$ as the parameters for LogSCM, we can tell that the first term of two objective functions are equivalent. If we let $\boldsymbol{c}_i = (\mathbf{Q}^\top \mathbf{Q})^{-1} \mathbf{Q}^\top \boldsymbol{\beta}_i = \mathbf{Q}^\dagger \boldsymbol{\beta}_i$ and assume $\boldsymbol{\Sigma}^{-1} = (\mathbf{Q}^\dagger)^\top \mathbf{Q}^\dagger$, $\boldsymbol{\beta}_f = \bar{\boldsymbol{\beta}} = (\sum_i \boldsymbol{\beta}_i)/N$, the similarity regularization term for LogSCM can be derived starting from the second term of logistic MEM objective function:

$$\text{S}_{MEM} = \sum_{i=1}^{N} (\boldsymbol{\beta}_i - \boldsymbol{\beta}_f)^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{\beta}_i - \boldsymbol{\beta}_f)$$

$$
\begin{aligned}
&= \sum_{i=1}^{N} (\boldsymbol{c}_i - \bar{\boldsymbol{c}})^\top \mathbf{Q}^\top (\mathbf{Q}^\dagger)^\top \mathbf{Q}^\dagger \mathbf{Q} (\boldsymbol{c}_i - \bar{\boldsymbol{c}}) \\
&= \sum_{i=1}^{N} (\boldsymbol{c}_i - \bar{\boldsymbol{c}})^\top (\boldsymbol{c}_i - \bar{\boldsymbol{c}}) = \sum_{i=1}^{N} \boldsymbol{c}_i^\top \boldsymbol{c}_i - N \bar{\boldsymbol{c}}^\top \bar{\boldsymbol{c}} \\
&= \sum_{i=1}^{N} \boldsymbol{c}_i^\top \boldsymbol{c}_i - \sum_{i,j} \frac{1}{N} \boldsymbol{c}_i^\top \boldsymbol{c}_j \\
&= \lambda \sum_{i} (\boldsymbol{c}_i)^\top \boldsymbol{c}_i d_{ii} - \lambda \sum_{i,j} (\boldsymbol{c}_i)^\top \boldsymbol{c}_j w_{ij} \\
&\quad \text{if } w_{ij} = \frac{1}{\lambda N} \text{ and } d_{ii} = \sum_{j} w_{ij} = \frac{1}{\lambda}, \\
&= \lambda \sum_{i,j} \|\boldsymbol{c}_i - \boldsymbol{c}_j\|^2 w_{ij} = \mathrm{S}_{SCM}.
\end{aligned}
$$

Then we can see that when $w_{ij} = 1/\lambda N$ for all entries in similarity matrix $\mathbf{W}$, the objective functions are equivalent, with the constraint that $rank(\mathbf{\Sigma}) \leq rank(\mathbf{Q}) = K$. Hence, our model shows more flexibility in terms of utilizing the similarity information and low-rank structure.

### *A.2 Derivation of Updating Rule in C Step*

Following we provide a detailed derivation of the updating rule in $\mathbf{C}$ Step. The formulation of LogSCM is as following:

$$
\begin{aligned}
\min_{\mathbf{C},\mathbf{Q}} \quad & \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \left\{ \log(1 + \exp(\boldsymbol{x}_{ij}^\top \mathbf{Q} \boldsymbol{c}_i)) - y_{ij} (\boldsymbol{x}_{ij}^\top \mathbf{Q} \boldsymbol{c}_i) \right\} + \lambda \operatorname{Tr}(\mathbf{C} \mathbf{L} \mathbf{C}^\top), \\
\text{s.t.} \quad & \boldsymbol{c}_i \geq 0, \ \boldsymbol{c}_i^\top \mathbf{1} = 1 \quad i = 1, ..., N.
\end{aligned}
\tag{A.1}
$$

In $\mathbf{C}$ step, we focus on solving $\mathbf{C}$ with a given $\mathbf{Q}^*$, i.e., $\mathbf{Q}^*$ could be the latest estimation of $\mathbf{Q}$ or from prior knowledge. Given $\mathbf{Q}^*$, the Lagrangian function could be derived as:

$$
L = \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} \left\{ \log(1 + \exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)) - y_{ij} (\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i) \right\} + \lambda \operatorname{Tr}(\mathbf{C} \mathbf{L} \mathbf{C}^\top) + \sum_{i=1}^{N} \eta_i (\boldsymbol{c}_i^\top \mathbf{1} - 1),
$$

by introducing the Lagrangian multiplier $\eta_i$ for normalization constraint $\boldsymbol{c}_i^\top \mathbf{1} = 1$. To solve for the optimal $\mathbf{C}$, we derive the gradient of the objective function regarding $\boldsymbol{c}_i$, as shown in

below:

$$\frac{\partial L}{\partial \boldsymbol{c}_i} = \frac{1}{n_i}\sum_{j=1}^{n_i}\left[\frac{\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)}{1+\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)} - y_{ij}\right]\mathbf{Q}^{*\top}\boldsymbol{x}_{ij} + 2\lambda(\mathbf{CL})_i + \eta_i \mathbf{1} = 0.$$

We use the complementary condition, i.e., $(\partial L/\partial c_{ik})c_{ik} = 0$, and get the following equation for $c_{ik}$:

$$\frac{1}{n_i}\sum_{j=1}^{n_i}\left[\frac{\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)}{1+\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)} - y_{ij}\right]\left(\mathbf{Q}^{*\top}\boldsymbol{x}_{ij}\right)_k c_{ik} + 2\lambda\left((\mathbf{CL})_i\right)_k c_{ik} + \eta_i c_{ik} = 0. \tag{A.2}$$

Then, with the normalization constraint $\sum_k c_{ik} = 1$, we can sum up the equations in Eq.(A.2) shown immediately above over $k$ and it will lead to:

$$\frac{1}{n_i}\sum_{j=1}^{n_i}\left[\frac{\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)}{1+\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)} - y_{ij}\right]\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i + 2\lambda(\mathbf{CL})_i^\top \boldsymbol{c}_i + \eta_i = 0.$$

Thus, with $\mathbf{L} = \mathbf{D} - \mathbf{W}$:

$$\eta_i = \frac{1}{n_i}\sum_{j=1}^{n_i} y_{ij}(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i) - \frac{1}{n_i}\sum_{j=1}^{n_i}\frac{\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)}{1+\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)}(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i) - 2\lambda(\mathbf{CD})_i^\top \boldsymbol{c}_i + 2\lambda(\mathbf{CW})_i^\top \boldsymbol{c}_i. \tag{A.3}$$

Using the expression of multiplier $\eta_i$ in Eq.(A.3), the complementary condition in Eq.(A.2) can be rewritten as:

$$\begin{aligned} c_{ik}\times \Bigg\{ &\frac{1}{n_i}\sum_{j=1}^{n_i}\left[\frac{\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)}{1+\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)} - y_{ij}\right]\left(\mathbf{Q}^{*\top}\boldsymbol{x}_{ij}\right)_k + 2\lambda\left((\mathbf{CD})_i\right)_k - 2\lambda\left((\mathbf{CW})_i\right)_k \\ &+\frac{1}{n_i}\sum_{j=1}^{n_i}\left[y_{ij} - \frac{\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)}{1+\exp(\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i)}\right](\boldsymbol{x}_{ij}^\top \mathbf{Q}^* \boldsymbol{c}_i) - 2\lambda(\mathbf{CD})_i^\top \boldsymbol{c}_i + 2\lambda(\mathbf{CW})_i^\top \boldsymbol{c}_i \Bigg\} = 0. \end{aligned}$$

Then separate this equation into positive part and negative part:

$$
\begin{aligned}
&\Bigg\{\frac{1}{n_i}\sum_{j=1}^{n_i}\left[\delta_+\left(\frac{\exp(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i)}{1+\exp(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i)}\left(\mathbf{Q}^{*\top}\boldsymbol{x}_{ij}\right)_k\right)-\delta_-\left(y_{ij}\left(\mathbf{Q}^{*\top}\boldsymbol{x}_{ij}\right)_k\right)\right]\\
&+\frac{1}{n_i}\sum_{j=1}^{n_i}\left[\delta_+\left(y_{ij}(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i)\right)-\delta_-\left(\frac{\exp(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i)}{1+\exp(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i)}(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i)\right)\right]\\
&+2\lambda\left((\mathbf{CD})_i\right)_k+2\lambda(\mathbf{CW})_i^\top\boldsymbol{c}_i\Bigg\}c_{ik}\\
-&\Bigg\{\frac{1}{n_i}\sum_{j=1}^{n_i}\left[-\delta_-\left(\frac{\exp(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i)}{1+\exp(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i)}\left(\mathbf{Q}^{*\top}\boldsymbol{x}_{ij}\right)_k\right)+\delta_+\left(y_{ij}\left(\mathbf{Q}^{*\top}\boldsymbol{x}_{ij}\right)_k\right)\right]\\
&+\frac{1}{n_i}\sum_{j=1}^{n_i}\left[-\delta_-\left(y_{ij}(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i)\right)+\delta_+\left(\frac{\exp(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i)}{1+\exp(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i)}(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i)\right)\right]\\
&+2\lambda\left((\mathbf{CW})_i\right)_k+2\lambda(\mathbf{CD})_i^\top\boldsymbol{c}_i\Bigg\}c_{ik}\\
=&0.
\end{aligned}
$$

Here, $\delta_+(\cdot)$ is a function defined as $\delta_+(x) := \max(x, 0)$ and $\delta_-(\cdot)$ is defined as $\delta_-(x) := \min(x, 0)$. The equation above can leads to our updating rule:

$$
\begin{aligned}
c_{ik}^{(m+1)} = c_{ik}^{(m)}\times&\Bigg\{\frac{1}{n_i}\sum_{j=1}^{n_i}\Bigg[-\delta_-\left(\frac{\exp(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i^{(m)})}{1+\exp(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i^{(m)})}\left(\mathbf{Q}^{*\top}\boldsymbol{x}_{ij}\right)_k\right)+\delta_+\left(y_{ij}\left(\mathbf{Q}^{*\top}\boldsymbol{x}_{ij}\right)_k\right)\Bigg]\\
&+\frac{1}{n_i}\sum_{j=1}^{n_i}\Bigg[-\delta_-\left(y_{ij}(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i^{(m)})\right)+\delta_+\left(\frac{\exp(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i^{(m)})}{1+\exp(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i^{(m)})}(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i^{(m)})\right)\Bigg]\\
&+2\lambda\left((\mathbf{C}^{(m)}\mathbf{W})_i\right)_k+2\lambda(\mathbf{C}^{(m)}\mathbf{D})_i^\top\boldsymbol{c}_i^{(m)}\Bigg\}\Bigg/\\
&\Bigg\{\frac{1}{n_i}\sum_{j=1}^{n_i}\left[\delta_+\left(\frac{\exp(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i^{(m)})}{1+\exp(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i^{(m)})}\left(\mathbf{Q}^{*\top}\boldsymbol{x}_{ij}\right)_k\right)-\delta_-\left(y_{ij}\left(\mathbf{Q}^{*\top}\boldsymbol{x}_{ij}\right)_k\right)\right]\\
&+\frac{1}{n_i}\sum_{j=1}^{n_i}\left[\delta_+\left(y_{ij}(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i^{(m)})\right)-\delta_-\left(\frac{\exp(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i^{(m)})}{1+\exp(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i^{(m)})}(\boldsymbol{x}_{ij}^\top\mathbf{Q}^*\boldsymbol{c}_i^{(m)})\right)\right]\\
&+2\lambda\left((\mathbf{C}^{(m)}\mathbf{D})_i\right)_k+2\lambda(\mathbf{C}^{(m)}\mathbf{W})_i^\top\boldsymbol{c}_i^{(m)}\Bigg\}.
\end{aligned}
\tag{A.4}
$$

Eq.(A.4) is derived from the complementary condition and the original constraint for the normalization of the membership vector. Therefore, it is a necessary condition to solve Eq.(A.1) and is a stationary point. In addition, the introducing of the $\delta$-functions ensures that the numerator and the denominator are both non-negative. Thus, given any positive initial membership matrix $\mathbf{C}^{(0)}$, the non-negativity of $\mathbf{C}^{(m)}$ is guaranteed.

Eq.(A.4) is the updating rule for every $c_{ik}$ entry in $\mathbf{C}$, given $\mathbf{Q}^*$. This completes our two-step iterative algorithm for LogCM and LogSCM.

## *A.3 Note for Data Generation in Simulation Studies*

The data generation note in this section is used in our simulation studies for LogCM (Section 2.4).

### *A.3.1 Parameters*

We first generate heterogeneous individual models ($\boldsymbol{\beta}_i$'s). Under collaborative learning framework, the parameters required are the canonical matrix $\mathbf{Q}$ which contains $K$ canonical models, and the membership matrix $\mathbf{C}$, which includes membership vectors $\boldsymbol{c}_i$'s for every individual.

With any given $K$, we manually set the parameters of the canonical models encoded in $\mathbf{Q}$ to make sure that they are different enough. It is because the canonical models, by assumption, represent different preferences, different behavior or mechanism patterns. For instance, in our study with $p = 5$ predicting factors, we randomly pick one factor parameter to be 0 for each canonical model (and they are not the same element) while the absolute values of other parameters are all generated to be at least 1. In this way, we will generate a matrix $\mathbf{Q}$ without homogeneous canonical structure.

Other generating procedures may also be suitable, for example, different canonical models have different direction on the same factors.

Then, for generating $\mathbf{C}$, we notice that $\boldsymbol{c}_i$'s are membership vectors with constraints $\boldsymbol{c}_i^\top \mathbf{1} = 1$ and $\boldsymbol{c}_i \geq 0$ for all $i$. The Dirichlet distribution is designed to model this type of data.

The Dirichlet distribution is a family of continuous multivariate probability distributions parameterized by a $p$-length vector $\boldsymbol{\alpha}$ of positive real values. It is a multivariate generalization of the Beta distribution and will generate $p$ probabilities which add up to 1. With larger element in the parameter vector $\boldsymbol{\alpha}$, the corresponding probability generated tends to be larger.

In our study, for $K = 3$, we design three distinct Dirichlet distributions as: $F_1(\boldsymbol{c}) \sim Dir(\nu, 1, 1)$, $F_2(\boldsymbol{c}) \sim Dir(1, \nu, 1)$, and $F_3(\boldsymbol{c}) \sim Dir(1, 1, \nu)$. For each individual, we first randomly assign him/her into one group, i.e., select one of the three Dirichlet distributions. Then, generate a random vector following this specified Dirichlet distribution. The tuning parameter element $\nu$ controls the the degree of heterogeneous in model structure. For $F_1(\boldsymbol{c})$, large $\nu$ (like 20) indicates that canonical model #1 will domain the individual model with a high probability because the first element of the membership vector is quite likely to be much larger than the other two. The larger $\nu$ is, the more dominant the corresponding canonical model is. On the contrary, if $\nu$ is close to 1, the difference between canonical models will probably vanish in final individual models, which indicating homogeneity in the population. Therefore, we set our $\nu = 20$ and this makes sure that we can see a canonical structure in individual models.

With $\mathbf{Q}$ and $\mathbf{C}$, we can obtain the parameter vectors of all the individuals by $\boldsymbol{\beta}_i = \mathbf{Q}\boldsymbol{c}_i$.

### *A.3.2 Samples*

After having individual models, we can generate our training and testing samples with $\boldsymbol{\beta}_i$'s. We generate $\boldsymbol{x}_{ij}$'s for individual $i$ from a normal distribution. Since we consider the individuals in population are heterogeneous, for individuals generated from different Dirichlet distributions, we use different normal distributions to generate $\boldsymbol{x}_{ij}$. And $y_{ij}$'s are calculated accordingly with a small normal distributed noise.

Here, we consider one more layer of complexity, which is the balance of the two classes in the binary outcomes. In practice, it is known that imbalanced labels could result in great difficulty for many methods to effectively learn the model from the data, as the dominant

class would overwhelm the parameter estimation in its own favor. Thus, we also would like to generate imbalanced data in our simulation experiments. There are various ways to achieve this. For instance, it could be done by changing the mean vectors that is used in generating $\boldsymbol{x}_{ij}$, i.e., for balanced data, we design $\boldsymbol{x}_{ij}$ to make the two labels of similar sizes (50% for each); for imbalanced data, we make the percentage of one label to be around 80%.

### *A.3.3 Division of the Simulated Data*

Finally, for the data points generated for each individual, we divide them into different usages to further incorporate sparsity in our simulation studies.

For each individual, using the process mentioned above, we generate 40 data points and randomly pick 10 data points for testing. As we have the option to use the remaining 30 data points for training, we further consider two realistic scenarios that we call them as dense sampling and sparse sampling. For dense sampling we set the data size $M \sim Unif(21, 30)$ to mimic the scenario that personal data is sufficient, while for sparse sampling we set $M \sim Unif(6, 12)$. In other words, sparse sampling refers to the application contexts in which there are only a few data points for each individual.

# Appendix B

## *B.1 Derivation of Updating Rule in C Step of LogCM-T*

Following, we provide a detailed derivation of the updating rule in Online LogCM-T. The formulation of the subproblem of updating membership vectors given $\mathbf{Q}$ is as following:

$$\begin{aligned} \min_{\boldsymbol{c}_{ip},\forall p} \quad & \sum_{t=1}^{n_i}\left\{\log\left(1+\exp\left(\sum_{p=1}^{P}(x_{it,p}(\mathbf{Q}_p\boldsymbol{c}_{ip})^\top\boldsymbol{v}_t)\right)\right)-y_{it}\left(\sum_{p=1}^{P}(x_{it,p}(\mathbf{Q}_p\boldsymbol{c}_{ip})^\top\boldsymbol{v}_t))\right)\right\}, \\ \text{s.t.} \quad & \boldsymbol{c}_{ip}\geq 0,\ \boldsymbol{c}_{ip}^\top\mathbf{1}=1 \quad p=1,\ldots,P. \end{aligned} \tag{B.1}$$

The decision variables are the membership vectors for individual $i$, and the Lagrangian function could be written as:

$$\begin{aligned} L=&\sum_{t=1}^{n_i}\left\{\log\left(1+\exp\big(\sum_{p=1}^{P}x_{it,p}(\mathbf{Q}_p\boldsymbol{c}_{ip})^\top\boldsymbol{v}_t\big)\right)-y_{it}\big(\sum_{p=1}^{P}x_{it,p}(\mathbf{Q}_p\boldsymbol{c}_{ip})^\top\boldsymbol{v}_t\big)\right\} \\ &+\sum_{p=1}^{P}\eta_p(\boldsymbol{c}_{ip}^\top\mathbf{1}-1). \end{aligned} \tag{B.2}$$

In Eq.(B.2), we introduce the Lagrangian multiplier $\eta_p$ for constraint $\boldsymbol{c}_{ip}^\top\mathbf{1}=1$. To get the optimal $\boldsymbol{c}_{ip}, p=1,2,...,P$, we could derive the gradient of the objective function regarding $\boldsymbol{c}_{ip}$,

$$\frac{\partial L}{\partial \boldsymbol{c}_{ip}}=\sum_{t=1}^{n_i}\left\{\left(\frac{\exp(\sum_{p=1}^{P}x_{it,p}(\mathbf{Q}_p\boldsymbol{c}_{ip})^\top\boldsymbol{v}_t)}{1+\exp(\sum_{p=1}^{P}x_{it,p}(\mathbf{Q}_p\boldsymbol{c}_{ip})^\top\boldsymbol{v}_t)}-y_{it}\right)\times x_{it,p}\boldsymbol{Q_p}^\top\boldsymbol{v}_t\right\}+\eta_p\mathbf{1}=0. \tag{B.3}$$

According to complementary condition, we also have $(\partial L/\partial c_{ip,k})c_{ip,k}=0$ $(k=1,2,\ldots,K_p;$ $p=1,2,\ldots,P)$, which could lead to the following equation,

$$\begin{aligned} \frac{\partial L}{\partial c_{ip,k}}c_{ip,k}=&\sum_{t=1}^{n_i}\left\{\left(\frac{\exp(\sum_{p=1}^{P}x_{it,p}(\mathbf{Q}_p\boldsymbol{c}_{ip})^\top\boldsymbol{v}_t)}{1+\exp(\sum_{p=1}^{P}x_{it,p}(\mathbf{Q}_p\boldsymbol{c}_{ip})^\top\boldsymbol{v}_t)}-y_{it}\right)\times x_{it,p}(\mathbf{Q}_p{}^\top\boldsymbol{v}_t)_k c_{ip,k}\right\}+\eta_p c_{ip,k} \\ &=0. \end{aligned} \tag{B.4}$$

Given $\boldsymbol{c}_{ip}^{\top}\mathbf{1} = 1$, i.e., $\sum_{k=1}^{K_p} c_{ip,k} = 1$, it could be further modified as,

$$\sum_{t=1}^{n_i} \left\{ \left( \frac{\exp(\sum_{p=1}^{P} x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^{\top} \boldsymbol{v}_t)}{1 + \exp(\sum_{p=1}^{P} x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^{\top} \boldsymbol{v}_t)} - y_{it} \right) \times x_{it,p}(\mathbf{Q}_p{}^{\top} \boldsymbol{v}_t)^{\top} \boldsymbol{c}_{ip} \right\} + \eta_p = 0. \tag{B.5}$$

With Eq.(B.5), we could write $\eta_p$ in the following way:

$$\begin{aligned} \eta_p = & \sum_{t=1}^{n_i} y_{it} \times x_{it,p}(\mathbf{Q}_p{}^{\top} \boldsymbol{v}_t)^{\top} \boldsymbol{c}_{ip} \\ & - \sum_{t=1}^{n_i} \frac{\exp(\sum_{p=1}^{P} x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^{\top} \boldsymbol{v}_t)}{1 + \exp(\sum_{p=1}^{P} x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^{\top} \boldsymbol{v}_t)} \times x_{it,p}(\mathbf{Q}_p{}^{\top} \boldsymbol{v}_t)^{\top} \boldsymbol{c}_{ip}. \end{aligned} \tag{B.6}$$

Replace $\eta_p$ in Eq.(B.5) with Eq.(B.6), and we will get,

$$\begin{aligned} c_{ip,k} \times \Bigg\{ & \sum_{t=1}^{n_i} \left( \frac{\exp(\sum_{p=1}^{P} x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^{\top} \boldsymbol{v}_t)}{1 + \exp(\sum_{p=1}^{P} x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^{\top} \boldsymbol{v}_t)} - y_{it} \right) \times x_{it,p}(\mathbf{Q}_p{}^{\top} \boldsymbol{v}_t)_k \\ & + \sum_{t=1}^{n_i} \left( y_{it} - \frac{\exp(\sum_{p=1}^{P} x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^{\top} \boldsymbol{v}_t)}{1 + \exp(\sum_{p=1}^{P} x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^{\top} \boldsymbol{v}_t)} \right) \times x_{it,p}(\mathbf{Q}_p{}^{\top} \boldsymbol{v}_t)^{\top} \boldsymbol{c}_{ip} \Bigg\} = 0 \end{aligned} \tag{B.7}$$

Since $c_{ip,k}$ should be non-negative, we define $\delta_+(x) := \max(x, 0)$ and $\delta_-(x) := \min(x, 0)$, with which the equation above could be separated into a positive part and a negative part:

$$\begin{aligned} & c_{ip,k} \times \Bigg\{ \sum_{t=1}^{n_i} \Bigg[ \delta_+ \left( \frac{\exp(\sum_{p=1}^{P} x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^{\top} \boldsymbol{v}_t)}{1 + \exp(\sum_{p=1}^{P} x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^{\top} \boldsymbol{v}_t)} \times x_{it,p}(\mathbf{Q}_p^{\top} \boldsymbol{v}_t)_k \right) \\ & \qquad - \delta_- \left( \frac{\exp(\sum_{p=1}^{P} x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^{\top} \boldsymbol{v}_t)}{1 + \exp(\sum_{p=1}^{P} x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^{\top} \boldsymbol{v}_t)} \times x_{it,p}(\mathbf{Q}_p^{\top} \boldsymbol{v}_t)^{\top} \boldsymbol{c}_{ip} \right) \\ & \qquad + \delta_+ \left( y_{it} \times x_{it,p}(\mathbf{Q}_p^{\top} \boldsymbol{v}_t)^{\top} \boldsymbol{c}_{ip} \right) - \delta_- \left( y_{it} \times x_{it,p}(\mathbf{Q}_p^{\top} \boldsymbol{v}_t)_k \right) \Bigg] \Bigg\} \\ & - c_{ip,k} \times \Bigg\{ \sum_{t=1}^{n_i} \Bigg[ \delta_+ \left( \frac{\exp(\sum_{p=1}^{P} x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^{\top} \boldsymbol{v}_t)}{1 + \exp(\sum_{p=1}^{P} x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^{\top} \boldsymbol{v}_t)} \times x_{it,p}(\mathbf{Q}_p^{\top} \boldsymbol{v}_t)^{\top} \boldsymbol{c}_{ip} \right) \\ & \qquad - \delta_- \left( \frac{\exp(\sum_{p=1}^{P} x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^{\top} \boldsymbol{v}_t)}{1 + \exp(\sum_{p=1}^{P} x_{it,p}(\mathbf{Q}_p \boldsymbol{c}_{ip})^{\top} \boldsymbol{v}_t)} \times x_{it,p}(\mathbf{Q}_p^{\top} \boldsymbol{v}_t)_k \right) \\ & \qquad + \delta_+ \left( y_{it} \times x_{it,p}(\mathbf{Q}_p^{\top} \boldsymbol{v}_t)_k \right) - \delta_- \left( y_{it} \times x_{it,p}(\mathbf{Q}_p^{\top} \boldsymbol{v}_t)^{\top} \boldsymbol{c}_{ip} \right) \Bigg] \Bigg\} \\ & = 0. \end{aligned} \tag{B.8}$$

Thus, we could derive an iteratively updating rule for $\boldsymbol{c}_{ip}$ similarly as in Chapter 2:

$$
\begin{aligned}
c_{ip,k}^{(m+1)} = c_{ip,k}^{(m)} \times \Bigg\{ & \sum_{t=1}^{n_i} \Bigg[ \delta_+ \Bigg( \frac{\exp(\sum_{p=1}^{P} x_{it,p} (\mathbf{Q}_p \boldsymbol{c}_{ip}^{(m)})^\top \boldsymbol{v}_t)}{1 + \exp(\sum_{p=1}^{P} x_{it,p} (\mathbf{Q}_p \boldsymbol{c}_{ip}^{(m)})^\top \boldsymbol{v}_t)} \times x_{it,p} (\mathbf{Q}_p^\top \boldsymbol{v}_t)^\top \boldsymbol{c}_{ip}^{(m)} \Bigg) \\
& - \delta_- \Bigg( \frac{\exp(\sum_{p=1}^{P} x_{it,p} (\mathbf{Q}_p \boldsymbol{c}_{ip}^{(m)})^\top \boldsymbol{v}_t)}{1 + \exp(\sum_{p=1}^{P} x_{it,p} (\mathbf{Q}_p \boldsymbol{c}_{ip}^{(m)})^\top \boldsymbol{v}_t)} \times x_{it,p} (\mathbf{Q}_p^\top \boldsymbol{v}_t)_k \Bigg) \\
& + \delta_+ \Big( y_{it} \times x_{it,p} (\mathbf{Q}_p^\top \boldsymbol{v}_t)_k \Big) - \delta_- \Big( y_{it} \times x_{it,p} (\mathbf{Q}_p^\top \boldsymbol{v}_t)^\top \boldsymbol{c}_{ip}^{(m)} \Big) \Bigg] \Bigg\} \Bigg/ \\
& \Bigg\{ \sum_{t=1}^{n_i} \Bigg[ \delta_+ \Bigg( \frac{\exp(\sum_{p=1}^{P} x_{it,p} (\mathbf{Q}_p \boldsymbol{c}_{ip}^{(m)})^\top \boldsymbol{v}_t)}{1 + \exp(\sum_{p=1}^{P} x_{it,p} (\mathbf{Q}_p \boldsymbol{c}_{ip}^{(m)})^\top \boldsymbol{v}_t)} \times x_{it,p} (\mathbf{Q}_p^\top \boldsymbol{v}_t)_k \Bigg) \\
& - \delta_- \Bigg( \frac{\exp(\sum_{p=1}^{P} x_{it,p} (\mathbf{Q}_p \boldsymbol{c}_{ip}^{(m)})^\top \boldsymbol{v}_t)}{1 + \exp(\sum_{p=1}^{P} x_{it,p} (\mathbf{Q}_p \boldsymbol{c}_{ip}^{(m)})^\top \boldsymbol{v}_t)} \times x_{it,p} (\mathbf{Q}_p^\top \boldsymbol{v}_t)^\top \boldsymbol{c}_{ip}^{(m)} \Bigg) \\
& + \delta_+ \Big( y_{it} \times x_{it,p} (\mathbf{Q}_p^\top \boldsymbol{v}_t)^\top \boldsymbol{c}_{ip}^{(m)} \Big) - \delta_- \Big( y_{it} \times x_{it,p} (\mathbf{Q}_p^\top \boldsymbol{v}_t)_k \Big) \Bigg] \Bigg\}
\end{aligned}
\tag{B.9}
$$

In Eq.(B.9), the superscript $m$ refers to the order of iteration. By introducing $\delta$-functions, we ensure that the numerator and the denominator are both non-negative. Therefore, given any non-negative initial membership vectors $\boldsymbol{c}_{ip}, p = 1, 2, ..., P$, the non-negativity requirement of the membership vectors is guaranteed.

# Appendix C

## *C.1 Proof to Lemma 4.2*

**Lemma 4.2:** *For two non-negative matrices* $\mathbf{A}_1 \in \mathbb{R}^{p\times q}$ *and* $\mathbf{A}_2 \in \mathbb{R}^{q\times r}$*, whose column sums are all 1, the product* $\mathbf{A}_1\mathbf{A}_2$ *is also a non-negative matrix with all column sums being 1.*

***Proof*** For matrices $\mathbf{A}_1 = \{a_{1,ij}\}$ and $\mathbf{A}_2 = \{a_{2,ij}\}$, we have:

$$a_{1,ij} \geq 0,\ a_{2,ij} \geq 0,\ \forall i, j;$$

$$\sum_{i=1}^{p} a_{1,ij} = 1,\ \forall j = 1, \ldots, q;$$

$$\sum_{i=1}^{q} a_{2,ij} = 1,\ \forall j = 1, \ldots, r.$$

For the product matrix $\mathbf{A}_1\mathbf{A}_2 \in \mathbb{R}^{p\times r} = \{x_{ij}\}$, each entry $x_{ij}$ satisfies:

$$x_{ij} = \sum_{k=1}^{q} a_{1,ik}a_{2,kj} \geq 0.$$

$$\sum_{i=1}^{p} x_{ij} = \sum_{i=1}^{p}\sum_{k=1}^{q} a_{1,ik}a_{2,kj} = \sum_{k=1}^{q}\big(\sum_{i=1}^{p} a_{1,ik}\big)a_{2,kj} = \sum_{k=1}^{q} 1 \cdot a_{2,kj} = 1$$

Hence, the product matrix $\mathbf{A}_1\mathbf{A}_2$ is non-negative, and all column sums are 1.

# Appendix D

## *D.1 Proof to Lemma 5.2*

We provide the proof of Lemma 5.2 here.

**Lemma 5.2:** *In logit models, when the reward $r$ is linearly related to the attributes $\boldsymbol{x}$, there will be an infinite set of estimated coefficients $\hat{\boldsymbol{\beta}}$ that can lead to same utilities, and therefore, all maximize the log-likelihood of the model. In some of these solutions, $\hat{\beta}_r$ may be negative.*

***Proof*** To prove this, we first rewrite the coefficient $\boldsymbol{\beta}$ as $\boldsymbol{\beta} = [\boldsymbol{\beta}_{\boldsymbol{x}}^\top, \beta_r]^\top$, so the systematic utility $V = \beta_0 + \boldsymbol{\beta}_{\boldsymbol{x}}^\top \boldsymbol{x} + \beta_r r$. For the logit models like the one in Eq.(5.1), the log-likelihood to be maximized is:

$$\max_{\boldsymbol{\beta}_{\boldsymbol{x}}, \beta_r} \quad \sum_j \log\left[1 + \exp(\beta_0 + \boldsymbol{\beta}_{\boldsymbol{x}}^\top \boldsymbol{x}_j + \beta_r r_j)\right] - y_i(\beta_0 + \boldsymbol{\beta}_{\boldsymbol{x}}^\top \boldsymbol{x}_j + \beta_r r_j), \tag{D.1}$$

where $t$ indicates the data points. Suppose that $\{\beta_0^*, \boldsymbol{\beta}_{\boldsymbol{x}}^*, \beta_r^*\}$ is one optimal solution of the log-likelihood function. When $r_t$ is linearly related to $\boldsymbol{x}_t$, which we can denote as $r_t = \gamma_0 + \boldsymbol{\gamma}^\top \boldsymbol{x}_t$, for any real number $\delta \neq 1$, we will have:

$$r_j = \frac{\gamma_0}{1-\delta} + \frac{\boldsymbol{\gamma}^\top}{1-\delta}\boldsymbol{x}_j - \frac{\delta}{1-\delta} r_j. \tag{D.2}$$

Thus, we can define an infinite set of optimal solutions by noticing that

$$\beta_0^* + \boldsymbol{\beta}_{\boldsymbol{x}}^{*\top} \boldsymbol{x}_j + \beta_r^* r_j = (\beta_0^* + \frac{\beta_r^* \gamma_0}{1-\delta}) + (\boldsymbol{\beta}_{\boldsymbol{x}}^* + \frac{\beta_r^* \boldsymbol{\gamma}}{1-\delta})^\top \boldsymbol{x}_j + \frac{-\beta_r^* \delta}{1-\delta} r_j, \tag{D.3}$$

i.e., here, $\{\beta_0^{**} = (\beta_0^* + \frac{\beta_r^* \gamma_0}{1-\delta}), \boldsymbol{\beta}_{\boldsymbol{x}}^{**} = (\boldsymbol{\beta}_{\boldsymbol{x}}^* + \frac{\beta_r^* \boldsymbol{\gamma}}{1-\delta})^\top, \beta_r^{**} = \frac{-\beta_r^* \delta}{1-\delta}\}$ is a different optimal solution that achieves the same optimality as $\{\beta_0^*, \boldsymbol{\beta}_{\boldsymbol{x}}^*, \beta_r^*\}$.

Because of the arbitrariness of the optimal solutions, it is possible that $\hat{\beta}_r = \frac{-\beta_r^* \delta}{1-\delta} < 0$ (e.g., when $0 < \delta < 1$ and $\beta_r^* > 0$). The estimated coefficients for other attributes $\hat{\boldsymbol{\beta}}_{\boldsymbol{x}}$ will also be shifted by the amount of $\frac{\beta_r^* \boldsymbol{\gamma}}{1-\delta}$, and cause some unexpected sign errors. □