Multi-Product Utility Maximization for Economic Recommendation

Qi Zhao[†], Yongfeng Zhang[‡*], Yi Zhang[†], Daniel Friedman[♯]
[†]School of Engineering, [♯]Economics Department, University of California, Santa Cruz, CA 95060
[‡]College of Information and Computer Science, University of Massachusetts Amherst, MA 01003
{qzhao2,yiz,dan}@ucsc.edu, yongfeng@cs.umass.edu

ABSTRACT

Basic economic relations such as substitutability and complementarity between products are crucial for recommendation tasks, since the utility of one product may depend on whether or not other products are purchased. For example, the utility of a camera lens could be high if the user possesses the right camera (complementarity), while the utility of another camera could be low because the user has already purchased one (substitutability). We propose *multi-product utility maximization* (MPUM) as a general approach to recommendation driven by economic principles. MPUM integrates the economic theory of consumer choice with personalized recommendation, and focuses on the utility of *sets* of products for individual users. MPUM considers what the users already have when recommending additional products. We evaluate MPUM against several popular recommendation algorithms on two real-world E-commerce datasets. Results confirm the underlying economic intuition, and show that MPUM significantly outperforms the comparison algorithms under top-K evaluation metrics.

Categories and Subject Descriptors

M.5.4 [**Applied Computing**]: Law, Social and Behavioral Sciences- *Economics*; H.3.3 [**Information Search and Retrieval**]: Information Filtering

Keywords

Recommender Systems; Utility Maximization; Collaborative Filtering; Computational Economics

1. INTRODUCTION

Recommender Systems (RS) [29, 11, 19] play a major role in connecting producers to consumers in the burgeoning online economy [35]. These systems discover the products that might interest consumers by learning personalized consumer preferences.

So far, computer scientists have dominated the design and interpretation of recommendation algorithms which, for example, estimate various latent representations of user preferences [18, 17, 38, 13], or make recommendations based on content profiles [25, 19, 8], social relations [21, 20], knowledge graphs [6], or user-generated comments [43, 22]. Despite their practical importance in the economy, RS have seldom been designed (or interpreted) with economic principles in mind.

In particular, consider the economic concept of substitutes and complements [39]. Two products, A and B, are *substitutes* for an individual if A is is less valuable once B is already purchased; and they are *complements* if A is more valuable if B is also purchased. Substitutability and complementarity are basic product relations extensively investigated by economists [39]. Identifying and leveraging such relationships should bring major benefits. For example, the system should better avoid recommending more SLR digital cameras to a user if she has just purchased one, while instead recommending the matching camera lenses or batteries could be wise choices.

Consider also the economic concept of *cardinal utility*, which can measure the monetary value of a product (or set of products) to a consumer (e.g., [39] p. 166). Recommending the product set that brings highest utility to a user is the goal, but multi-product utility modeling is a non-trivial task. For example, how much more utility would each lens product give a particular consumer who has just purchased a particular camera? A principled approach is needed to quantify the total utility of sets of products for a given consumer.

In this paper, we propose just such a measure, called Multi-Product Utility Maximization (MPUM). Beginning with the crucial economic idea of Marginal Rate of Substitution (MRS) [39], we show how to construct pairwise utility functions that generalize well beyond standard Cobb-Douglas utility function [9]. Then we extend the pairwise utility functions to cover product sets with more than two products. Based on multinomial consumer choice modeling, MPUM conducts parameter learning by maximizing the utility of users' historical purchasing records, and then provides recommendations of the maximized total utility when combined with previous items.

The rest of the paper is organized as follows: we review the related work in Section 2, and introduce some basic definitions and concepts in Section 3. In Section 4, we propose our MPUM framework as well as the personalized transaction-based recommendation strategy. We further present extensive experimental results based on two different real-world datasets in Section 5. We conclude, and note some of the future research directions in Section 6.

*The first two authors contributed equally to this work.

WSDM 2017, February 06-10, 2017, Cambridge, United Kingdom

DOI: http://dx.doi.org/10.1145/3018661.3018674

2. RELATED WORK

The advent of internet has resulted in large sets of user behaviour records, which makes it possible for automatic recommendation by collaborative filtering, content-based filtering, or hybrid algorithms, etc.

Collaborative Filtering (CF) is based on the assumption that users with similar tastes for previous items would have similar preferences for new items, so the algorithm tends to recommend the highly ranked items by those users deemed similar to the current user [11, 38, 34, 33]. Such algorithms fall into two main categories. Memory-based CF predicts the unknown ratings for a user based on the weighted aggregation of ratings from other (usually the top-K most similar) users for the same item. Model-based CF adopts the collection of ratings to fit model parameters, and then makes predictions based on the fitted model, e.g., aspect models, flexible mixture models, or factorization models [4, 18, 24]. **Content-based filtering** adopts the information-rich features (meta data, words in description, price, tags, visual features, etc.) to describe the items that a user likes or dislikes so as to estimate the user preferences [25, 19, 8]. It usually recommends new items similar to previous items the user liked. The underlying research focuses on estimating a user's profile from her explicit feedback on previous items. **Hybrid recommendation** combines the advantages of collaborative filtering and content based filtering, and usually performs better than either filtering method alone [43, 22, 29].

Most such recommendation methods predict individual product scores for each user and rank the products accordingly. One major problem is that the top ranked recommendations might be very similar or even duplicated, which usually is not desirable. To address this issue, researchers proposed to diversify the recommendation results [15, 26, 19], and the diversification problem has been extensively studied in different scenarios such as news, movies or music recommendation [5][1]. A typical approach is to introduce certain diversity measures such as the number of categories in recommendations, the relative share of recommendations above or below a certain popularity rank percentile [2], or measures over product graphs [1]. Another approach is to measure and achieve diversity indirectly, such as the risk of a user portfolio of multiple products [36]. Although diversity is not the main focus of this paper, the proposed method indeed leads to diversity following the result of Diminishing Marginal Utility.

Perhaps the most closely related work to ours is [16], which seeks to classify product relationship into substitutes and complements based on data associated to products. It formulates the problem as supervised learning of the substitutes and complements relationship from observing co-purchased and co-viewed products. Our MPUM approach does not classify as such, but rather learns parameters of flexible utility models.

Another line of related research concerns the *next basket* recommendation, which models the sequential pattern of user purchases and recommends a set of items for user's next visit based on previous purchases. A series of methods have been developed for next basket recommendation [28, 41, 14, 7, 42], among which the Hierarchical Representation Model (HRM) [42] represents the state of the art. HRM combines general taste by conventional CF and information from previous transaction aggregated by a nonlinear function. Although our focus is not next basket recommendation, the proposed multi-product utility model can be applied to solve this problem, by assuming the products that the user purchased before as already in the basket, and to recommend more products so as to optimize the total utility for the user.

Several recent papers have tried in various ways to incorporate economics principles into E-commerce recommendation systems. In [40], the authors propose to adopt the law of diminishing marginal utility at individual product level, so that perishable and durable products are treated differently. In [45], the authors propose to estimate consumer's Willingness-To-Pay (WTP) in E-commerce setting, and the estimated WTP is used to price the products at individual level, so that seller's profit is maximized. In [44], a total surplus based recommendation framework is proposed to match producers and consumers so that the total social benefit is maximized. Our research falls into this direction and tries to handle the multi-product recommendation problem based on solid economics principles and practical recommendation techniques.

In particular, recognition of product substitutability and complementarity has been considered important to the study the market demand of one product affected by other products [3, 31, 32, 37]. Our proposed research is motivated by these existing and widely accepted economics principles.

3. BASIC COMPONENTS

We begin by introducting some key economic ideas that will play a central role in later sections.

3.1 Utility

In economics, *utility* is a measure of a consumer's preferences over alternative sets of goods or services. It is a basic building block of rational choice theory [10]. A consumer's total utility for a given set of goods is the consumer's satisfaction experienced from consuming these goods as a whole.

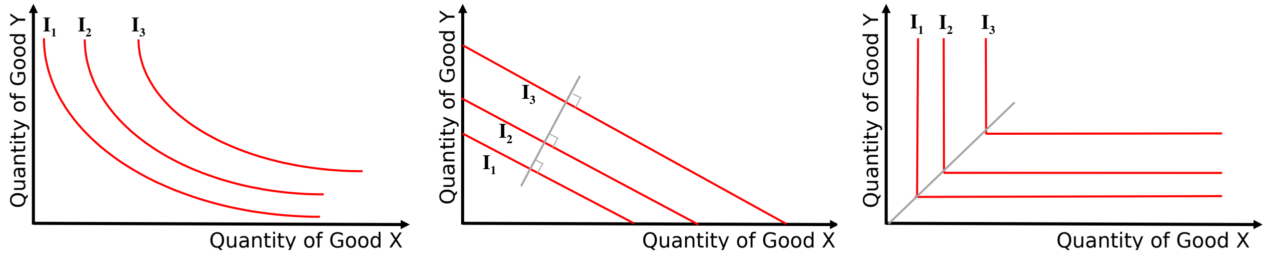
Utility $U(q)$ for a single good is a function of the consumed quantity q. It typically obeys the *Law Of Diminishing Marginal Utility* [30], which states that marginal utility, i.e., $U'(q) > 0$, is a decreasing function of the quantity consumed, i.e., $U''(q) < 0$. For example, a very hungry person may obtain huge satisfaction when consuming the first slice of bread, but consuming an additional slice at some point brings less additional satisfaction.

Economists have several standard functional forms for utility, including *Cobb-Douglas* utility, *Constant Elasticity of Substitution* (CES) utility, and *quasilinear* utility. As each utility function has its own assumptions and limitations, it seems worthwhile to explain and motivate our choice of the utility functional form.

3.2 Indifference Curves

In economics, *Indifference curves* are used to describe the preference relationship between a desirable pair of products, e.g., how increasing consumption of one product affects the relative marginal utilities of both products. As illustrated in Figure 1, each indifference curve represents the quantities of two given products that give the same level of utility. By definition, indifference curves for two goods possess the following properties,

- Any two points of the same curve give the same utility.



(a) General case (b) *Perfect substitute* products (c) *Perfect complementary* products

Figure 1: Illustrative indifference curves for common product pairs, perfect substitute product pairs, and perfect complementary product pairs. The utilities of the three illustrative curves satisfy $I_1 < I_2 < I_3$.

- Curves do not intersect.
- The tangent to any point on the curve has negative slope because, to keep total utility unchanged, increasing of quantity for one desirable product requires a decrease in the quantity of the other product.

Let q_j, q_k denote the consumed quantity of product j and product k, respectively. Since points of the same curve have the same utility, the total derivative at any point should be 0, we have:

$$\begin{aligned} \mathrm{d}U(q_j, q_k) &= \frac{\partial U}{\partial q_j}\mathrm{d}q_j + \frac{\partial U}{\partial q_k}\mathrm{d}q_k \\ &= U'_{q_j}\mathrm{d}q_j + U'_{q_k}\mathrm{d}q_k = 0 \end{aligned} \tag{1}$$

Let $h(q_j, q_k) = \frac{\mathrm{d}q_j}{\mathrm{d}q_k}$ denote the Marginal Rate of Substitution (MRS) at point (q_j, q_k). Then by Eq. (1) we have

$$h(q_j, q_k) = \frac{\mathrm{d}q_j}{\mathrm{d}q_k} = -\frac{U'_{q_k}}{U'_{q_j}} \tag{2}$$

Intuitively, the larger $|h(q_j, q_k)|$, the more consumption of product j is needed to compensate for a given decrease of the consumption of product k.

Economists have long recognized that the MRS function (2) completely captures the ordinal properties (the shape of the indifference curves) of a utility function. Figure 1 illustrates some of the possibilites. Figure 1a shows the generic case where MRS gets closer to zero smoothly as x increases. Figure 1b shows *perfect substitutes* where the MRS is a constant. In this case, a consumer is willing to exchange two products a fixed rate everywhere, for example, two kinds of pens that differ only in color, for a consumer who doesn't care about color. Figure 1c shows *perfect complements* where the utility is determined by the minimum of the two product quantities, for example left and right shoes. Given, say, 2 left shoes, the utility will not change by having more than 2 right shoes, and vice versa. The MRS of perfect complements is discontinuous — it changes from infinity to zero at certain point. Except in the limit of *perfect* complements, we can model the substitutes and complements via a differentiable MRS function.

4. MULTIPLE PRODUCT UTILITY MAXIMIZATION (MPUM) FRAMEWORK

In this section, we put the aforementioned ingredients together in a particular way.

Table 1: Choices of $h(q_k, q_j)$ and its corresponding utility function. $z(\cdot)$ denotes any monotone function.

	Polynomial	Exponential
$h(q_j, q_k)$	$-\frac{a}{1-a}(\frac{q_j}{q_k})^b$ $0 < a < 1, b \geq 0$	$-\frac{a}{1-a}e^{b(q_j - q_k)}$ $0 < a < 1, b \geq 0$
$U(q_j, q_k)$	$z\left((1-a)q_j^{1-b} + aq_k^{1-b}\right)$	$z\left((1-a)e^{-bq_j} + ae^{-bq_k}\right)$

4.1 Modeling Marginal Rate of Substitution

The first step is to find a proper utility functional form for $U(q_j, q_k)$ so that it can capture all possible products relationships shown in Figure 1. However, the right form for the utility function is not obvious, and it is not practical for us to try all possible alternatives of $U(q_j, q_k)$ by testing them against the cases in Figure 1. Since product substitutes and complements are better illustrated by MRS, we propose to find a proper functional form for MRS, and then to recover the utility function by solving differential equations.

By the Implicit Function Theorem, from the equation for an indifference curve ($U(q_j, q_k) = $ const.), we can alternatively express q_j as a function of q_k, i.e., $q_j = f(q_k)$, and the MRS defined in Eq. (2) becomes,

$$\frac{\mathrm{d}q_j}{\mathrm{d}q_k} = f'(q_k) = h(q_j, q_k) \tag{3}$$

where h is the chosen MRS function.

When choosing h, we are mainly concerned about two aspects: mathematical convenience and flexibility. Thus we propose to consider two functional forms in Table 1: polynomial and exponential.

Regardless of the specific form of h, the problem of recovering $U(q_j, q_k)$, or $f(q_k)$, boils down to solving the differential differential Eq.(3) for $f(\cdot)$. Table 1 summarizes the solutions for each alternative of h, as we now shall explain.

4.1.1 *Polynomial Case*

A preliminary question is whether h is expressive enough, e.g., whether it can cover the three cases shown in Figure 1. For the polynomial entry it is easy to see that, when $b = 0$, the MRS is constant at $\frac{a}{1-a}$; thus h covers the case of perfect substitutes. When b gets very large ($\rightarrow +\infty$), h is large when $\frac{q_j}{q_k} > 1$ and immediately drops to near 0 when $\frac{q_j}{q_k} < 1$, corresponding to the perfect complements case; when $0 < b < 1$, the MRS is for the general case shown in Figure 1a.

After applying standard differential function solution techniques (separation of variables and integrating) to Eq. (3), we reach the following equation,

$$\left((1-a)q_j^{1-b} + aq_k^{1-b}\right) = const. \tag{4}$$

Let's remind ourselves that MRS is defined when utility is set to unknown constant. The above equation suggests that the utility function is some monotone function of the left side of the above equation, namely,

$$U(q_j, q_k) = z\left((1-a)q_j^{1-b} + aq_k^{1-b}\right) \tag{5}$$

where $z(\cdot)$ is any monotone increasing function such as log and power. In particular, when $z(x) = x^{\frac{1}{1-b}}$, we obtain the well known *Constant Elasticity Substitution* (CES) utility function. Here $s = \frac{1}{b}$ is called the *Elasticity of Substitution*, and denotes the degree of substitutability between a pair of products. Specifically, the utility function models (perfect) complement product pairs when s is sufficiently large (towards $+\infty$), and (perfect) substitute pairs when s is sufficiently small (towards 0).

4.1.2 Exponential Case

A similar analysis applies to the exponential function form in Table 1). When $b = 0$, the MRS is constant $\frac{a}{1-a}$; when b goes to ∞, the MRS goes to infinity when $q_j > q_k$ and drops to zero when $q_j < q_k$. This suggests that the exponential functional form also can capture complements and substitutes.

Solving the differential equation Eq. (3) now yields:

$$ae^{-bq_k} + (1-a)e^{-bq_j} = const, \tag{6}$$

so the corresponding utility function is:

$$U(q_j, q_k) = z\left(ae^{-bq_k} + (1-a)e^{-bq_j}\right). \tag{7}$$

Here it is convenient to set the monotone function to be $z(x) = -\frac{1}{b}\log(x)$, to ensure that utility increases with an increase in the goods quantities. Using L'Hospital's rule, one can verify that we get perfect substitutes in the the limit $b \downarrow 0$.

Recall that we want to get perfect complements in the limit $b \to +\infty$, so the limit function should depend on the minimum of the two product quantities, namely,

$$U(q_1, q_2) = U\left(\min(q_1, q_2), \min(q_1, q_2)\right) \tag{8}$$

Like the polynomial function, the larger the b parameter, the better Eq. (7) can approximate the perfect complementary utilities, but which is a better approximation for finite b? Figure 2a shows the fitting errors (Root Mean Squared Error) of utility function defined in Eq. (5) and (7), respectively, and Figure 2b shows the indifference curves for polynomial and exponential utility function with $b = 5$. It can be seen that exponential function is closer to the "L" shape as shown in Figure 1c, and generally has smaller fitting error. On the other hand, the CES function is more familiar and seems better at approximating close substitutes (which are more common in our data), so we will rely on the polynomial utility function defined in Eq. (5) in the rest of the paper.

4.2 Multi-product Utility Modeling

In practice, it is very common that there are more than two products in a single transaction/order, and it is desirable for us to represent the utility of an arbitrary number of products. Let Ω_{it} be the set of products purchased by user i at time t. We consider the utility of Ω_{it} as the sum of the utility of all product pairs within Ω_{it}, namely,

$$\begin{aligned} U(\Omega_{it}) &= \tfrac{1}{|\Omega_{it}|-1} \sum_{j,k\in\Omega_{it}, j\neq k} U(q_j, q_k) \\ &= \tfrac{1}{|\Omega_{it}|-1} \sum_{j,k\in\Omega_{it}, j\neq k} \left(a_{jk}q_j^{1-b_{jk}} + (1-a_{jk})q_k^{1-b_{jk}}\right)^{\frac{1}{1-b_{jk}}} \end{aligned} \tag{9}$$

where a_{ij} and b_{ij} are specific parameters regarding a product pair, and $|\Omega_{it}|$ is the number of products in set Ω_{it}. The denominator is $|\Omega_{it}|-1$ because we want to count each product (not each product-pair) once when obtaining the total utility for the product set.

4.3 CF-based Re-Parameterization

As seen from Eq. (9), there are two unknown parameters a_{jk}, b_{jk} for product j and k. In general, we can reparamterize a_{jk} and b_{jk} as below,

$$a_{jk} = f(\vec{x_{jk}}, \vec{\theta_{jk}}) \tag{10}$$
$$b_{jk} = g(\vec{x_{jk}}, \vec{\eta_{jk}}) \tag{11}$$

where $\vec{x_{jk}}$ represents information related to product j and k, and $\vec{\theta_{jk}}, \vec{\eta_{jk}}$ are function parameters. $\vec{x_{jk}}$ can be anything such as product category, brand, reviews, etc. In [16], the authors propose to predict product substitutes and complements links using product information. It is possible for us to adopt similar idea as [16]. But for simplicity, we apply Collaborative Filtering (CF) to model a_{jk} and b_{jk}, sharing the same spirit of modeling ratings between user and item.

$$a_{jk} = \sigma\left(\alpha + \beta_j + \beta_k + \vec{x}_j^T\vec{x}_k\right) \tag{12}$$
$$b_{jk} = \exp\left(\mu + \gamma_j + \gamma_k + \vec{p}_j^T\vec{p}_k\right) \tag{13}$$
$$\vec{x}_j, \vec{p}_j \in \mathbb{R}^d, \beta_j, \gamma_j, \alpha, \mu \in \mathbb{R}$$

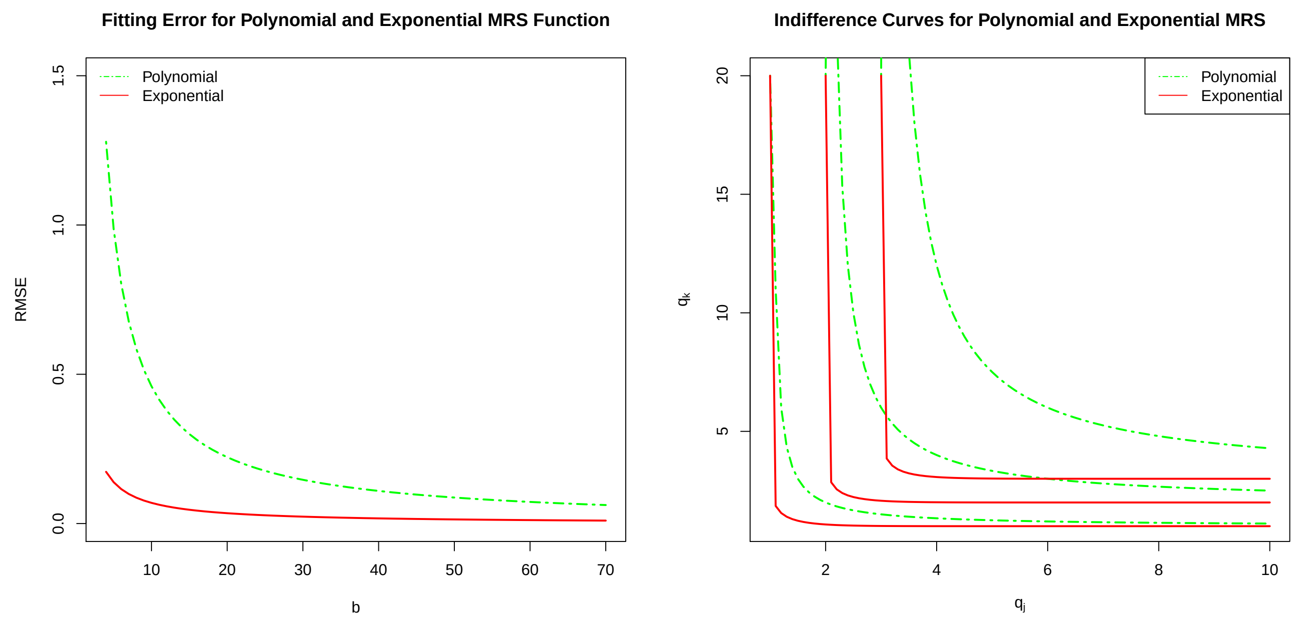
where $\sigma(\cdot)$ is Sigmoid function that ensures $0 < a_{jk} < 1$ and exponential function ensures $b_{ij} > 0$. Under CF representation, the parameters now are $\Theta = \{\vec{x}_j, \vec{p}_j, \beta_j, \gamma_j, \alpha, \mu\}$. Clearly, CF is a special case of f and g, as in which $\vec{x_{jk}}$ is empty.

4.4 Discrete Choice Modeling

In economics, *discrete choice models* characterize and predict consumer's choices between two or more alternatives, such as buying Coke or Pepsi, or choosing between different hotels for traveling. In this paper, at each time point t, consumer chooses product set Ω_{it} over other unobserved alternative product sets – $g(\Omega_{it})$. Let $\Pi_{it} = \{\Omega_{it}, g(\Omega_{it})\}$ represent all product sets and its k-th element is Π_{it}^k. Researchers in economics have developed *random utility models* (RUMs) for the discrete choice problem [23]. RUMs attach each alternative utility with a random value:

$$\widetilde{U_i}(\Pi_{it}^k) = U_i(\Pi_{it}^k) + \epsilon_k \tag{14}$$

where ϵ_k is a random variable that follows a certain probability distribution. The probability that a consumer chooses



(a) Fitting Error

(b) Indifference curves

Figure 2: Fitting error and indifference curves for utility functions defined in Eq. (5) and (7). In (a), one product quantity is fixed as 100 and the other product quantity varies from 1 to 10 with a tuning step of 1. In (b), the utility value of the curves are $1, 2$ and 3, respectively.

Π_{it}^1 (i.e. Ω_{it}) over other alternatives is:

$$P\left(\widetilde{U_i}(\Pi_{it}^1) > \widetilde{U_i}(\Pi_{it}^k)\right) = P\left(\epsilon_k - \epsilon_1 < U_i(\Pi_{it}^1) - U_i(\Pi_{it}^k)\right) \tag{15}$$

where $k = 2, \ldots, |\Pi_{it}|$. If ϵ_1 and ϵ_k follow *iid* extreme value distribution, it can be shown that the probability of choosing Π_{it}^1 is the following multinomial logistic model (MNL):

$$P(y_{it} = 1) = \frac{\exp\left(U_i(\Pi_{it}^1)\right)}{\sum_{k=1}^{|\Pi_{it}|} \exp\left(U_i(\Pi_{it}^k)\right)} \tag{16}$$

Alternatively, if ϵ_k follows a Gaussian distribution, then $P(y_{it})$ turns into a Probit model [23]. In the rest of this paper, we adopt the frequently used multinomial logistic model as in Eq.(16).

At each time point for a given user, the system usually observes a chosen product set (e.g. an order with multiple products, or a wishlist) Ω_{it}. The alternative product sets $g(\Omega_{it})$, however, are not observed. We can simply view Ω_{it} and $g(\Omega_{it})$ as positive and negative training records, respectively. Generation of negative samples will be introduced in Section 4.6.

4.5 Budget Constraint

The theory of consumer choice in microeconomics [12] concerns how consumers maximize their utility of consumption subject to their budget constraint. The utility of consumption is determined by consumer preferences and their corresponding utility functions as explained in Section 4.2. The consumer choice problem is thus formalized as the following constrained economic optimization problem,

$$\begin{aligned} &\underset{\{q_1, q_2, \ldots, q_N\}}{\text{argmax}} \quad U_{it}(q_1, q_2, \ldots, q_N) \\ &\quad s.t. \sum\nolimits_{j=1}^{N} p_j \times q_j \leq W_{it} \end{aligned} \tag{17}$$

where p_j is the price of product j, q_j is the consumed quantity of product j, and W_{it} is the consumer's budget. Eq.(17) can be solved by standard constraint optimization methods if the quantity variables q_j are real numbers. However, q_j are discrete numbers in most of the cases, this turns the above optimization problem into an integer programming problem that is NP hard. Due to the exponential computational complexity, it is not feasible to consider all possible product combinations for the objective function in Eq.(17). As a result, we only generate a sample of candidate sets Π_{it} for each observed chosen product set Ω_{it} when training the utility model.

4.6 Generate Negative Samples

To reduce the number of combinations for Eq.(17), we only consider products from observed product set Ω_{it}. Doing this greatly reduces the size of negative samples. In addition, we further require the total price of each negative sample equals to that of Ω_{it}, namely,

$$\sum_{j \in \Omega_{it}} p_j \times q_j = W_{it}. \tag{18}$$

As the product quantity can not be fractional, it is very likely that we can only find very few or even none solutions for Eq.(18). To overcome this issue, we relax Eq.(18) as,

$$|\sum_{j \in \Omega_{it}} p_j \times q_j - W_{it}| \leq \epsilon W_{it}. \tag{19}$$

where $0 < \epsilon < 1$. It is easy to prove that the solution of Equation 19 can be effectively obtained through dynamic programming at $\mathcal{O}(W_{it} * |\Omega_{it}|)$ running time complexity. $|\Omega_{it}|$ is the number of products in Ω_{it}. As we shall see in Table 3, $|\Omega_{it}|$ is about 10 in average. In the mean time, as W_{it} is bounded, it can be replaced by a constant number. Thus, the running time complexity of generating $g(\Omega_{it})$ is constant.

4.7 Model Parameter Learning

Given the observed transactions/orders and the consumer discrete choice modeling framework, the model parameters Θ can be optimized by maximizing the following Log-Likelihood ($\ell\ell$) of training data:

$$\begin{aligned} &\underset{\Theta}{\operatorname{argmax}}\ \ell\ell(D;\Theta) \\ &= \sum_{i,t:I_{i,t}=1} \log\left(P(y_{it}=1)\right) - \eta||\Theta||^2 \end{aligned} \tag{20}$$

where D is the training dataset, $I_{it} = 1$ if user i places an order at time t, and $P(y_{it})$ is the multinomial logistic regression model described in Eq.(16). $\eta \in \mathbb{R}^+$ is the regularization coefficient which is determined using cross validation.

There is no closed form solution to Eq.(20), and the optimal model parameters can be found using gradient based methods such as stochastic gradient descent.

4.8 Multi-Product Recommendation

The objective of our recommendation algorithm is to recommend a set of products that give the maximum utility without violating the budget constraint, as defined in Eq.(17). As we will see later, once we have learned the utility functions with Eq.(20), we can use the principle of utility maximization to predict the purchase quantity of each product for a given user. Here Eq.(17) takes the form

$$\underset{\Omega_{it}}{\operatorname{argmax}}\, U(\{q_j|j \in \Omega_{it}\}) \tag{21}$$

$$s.t. \sum_{j\in\Omega_{it}} p_j \times q_j \leq W_{it} \tag{22}$$

where Ω_{it} is a subset of all products. In practice, it is reasonable to limit $|\Omega_{it}|$ based on the typical size of an order. Due to the large search space of candidate products, it is not computationally feasible to evaluate all product sets exhaustively. Thus, we adopt greedy strategy which extends Ω_{it} incrementally by adding a product that gives the maximum incremental utility.

5. EXPERIMENT

We investigate the proposed framework based on two real-world E-commerce datasets. In this section, we report the experimental design, empirical results, and further analyses.

5.1 Dataset Description

The following two real-world datasets are used in our experiments:

Table 3: Basic statistics of the two datasets

Dataset	#Transactions	#Products	Average Size	Train/Test
Shop.com	86k	370k	~ 8	80%/20%
Amazon	7.8k	18k	~ 12	

Shop.com Data: Each record in the dataset is a purchase transaction with consumer ID, product(s) price, product(s) quantity, and the purchasing time. Key data statistics are summarized in Table 3. We treat each transaction as a positive training data point for Eq.(20), except that we omit transactions with less than two products, since we are focusing on multiple products.

Amazon Baby Registry Data: Amazon's Baby Registry[1] allows consumers to add and manage products for babies. Each registry is a wishlist containing a list of products the list owner wants to purchase. As the lists are publicly available, we crawled the lists and their products to generate this data set. Each product comes with title, price, brand, and category information. Some of the key statistics of the dataset is also summarized in Table 3. We treat each wishlist as a positive training point for Eq.(20).

Each dataset involved can be viewed as a collection of transactions. Each transaction holds a set of products the consumer purchased or wanted at certain time. The transactions are randomly split into two subsets - 80% of them are used for model training and the rest 20% is for performance evaluation. For each testing transaction, a portion (also 20%) of the products are randomly masked and they are predicted by recommendation algorithm based on other observed products in the same transaction.

For the training transactions, we generate negative training data (i.e. product sets not chosen by a user) for each positive set, which are required in Eq.(20) for model learning. Negative samples are generated as described in Section 4.6. For computational efficiency, we limit the size of Π_{it} to 10 when selecting negative sets from $g(\Omega_{it})$.

5.2 Evaluation Metric

Precision and recall at top-K are the most widely-adopted ranking evaluation metrics in practice, so we shall employ them here. Let Γ_i be the masked items in the i-th testing transaction and let Γ'_i be the list of items recommended by the algorithm under consideration. The metrics are defined as follows:

$$\begin{aligned} \text{Precision@K} &= \frac{1}{N}\sum_{i=1}^{N}\frac{|\Gamma'_i \cap \Gamma_i|}{K} \\ \text{Recall@K} &= \frac{1}{N}\sum_{i=1}^{N}\frac{|\Gamma'_i \cap \Gamma_i|}{|\Gamma_i|} \\ F_1\text{-measure@K} &= \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \end{aligned} \tag{23}$$

where K is the length of the recommendation list, and N is the number of testing transactions.

5.3 Experimental Results

We investigate the performance of our MPUM framework for the task of product recommendation for a transaction. For performance comparison, we consider the CF-based algorithm described in Section 4.3 and the Bayesian Personalized Ranking (BPR) [27] method, which are typical and state-of-the-art rating and ranking based recommendation approaches, respectively. In both of these comparison algorithms, the transactions are treated as "users", and recommendation is modeled as predicting purchasing quantity directly for each "user" (transaction). In order to recommend

[1]https://www.amazon.com/babyregistry

Table 2: Evaluation results for Top-K recommendation performance on Precision, Recall, and F_1-measure. All bolded improvements are statistically significant on 0.01 level in two-tailed t-test.

Dataset	Amazon Baby Registry Transactions								
@K	1			5			10		
Method	CF	BPRMF	MPUM	CF	BPRMF	MPUM	CF	BPRMF	MPUM
Precision (%)	0.092	0.117	**0.275**	0.437	0.473	**0.609**	0.262	0.513	**0.669**
Recall (%)	0.074	**0.112**	0.103	0.761	0.866	**1.279**	1.178	1.150	**2.844**
F_1-measure (%)	0.082	0.114	**0.150**	0.555	0.612	**0.825**	0.429	0.710	**1.083**
Dataset	Shop.com Transactions								
@K	1			5			10		
Method	CF	BPRMF	MPUM	CF	BPRMF	MPUM	CF	BPRMF	MPUM
Precision (%)	0.022	0.076	**0.470**	0.012	0.038	**0.286**	0.012	0.026	**0.160**
Recall (%)	0.017	0.003	**0.465**	0.035	0.013	**1.390**	0.073	0.185	**1.531**
F_1-measure (%)	0.019	0.006	**0.467**	0.018	0.019	**0.474**	0.021	0.046	**0.290**

for testing "users", it is necessary to learn their profile during the training stage. Towards this end, for each testing "user", 80% of its products are included in the training dataset, and the remaining 20% is used for performance evaluation. $|\Pi_{it}|$ in Eq.(16) is set to 10 and SGD learning rate is set to 0.01. For the comparison algorithms, latent vector dimension and regularization coefficient are set to 10 and 0.01, as they give best performance in a 10-fold cross validation setting. For a fair comparison, our method adopts the same parameters.

The evaluation results on Amazon and Shop.com datasets are reported in Table 2, and the largest value on each dataset and for each evaluation measure is significant at 0.01 level.

It can be seen from the results that our proposed MPUM algorithm outperforms the comparison algorithms in nearly all the cases, and in particular, the performance advantage is more pronounced on Shop.com dataset. A possible reason is that Shop.com dataset has much lower density (0.00205%) than Amazon dataset (0.0655%). Our method is less sensitive to low density than CF and BPR because they both introduce latent vectors for users (i.e., transactions in our problem) and products, and then learn the vectors through user-product interaction pairs; while our MPUM algorithm only concerns product-product relationships and models the transactions indirectly through its products without the need of considering the very sparse user-product pairs. As a result, our MPUM requires far fewer model parameters than the comparison algorithms.

(a) A nipple product that is complementary with the feeding bottle product in the right side

(b) This nipple product is a very close substitute for other nipple products

(c) A feeding bottle product

Figure 3: Examples of complementary and substitute products from Amazon Baby Registry dataset.

5.4 Empirical Tests of Economic Intuition

Do the learned utility functions make economic sense? A convenient aspect of the CES utility functions we work with is that one can read off the Elasticity of Substitution (ES) for real-world products learned by our model.

As shown in Figure 3, we find that the product pair with the lowest ES in the Amazon Baby Registry dataset is a nipple together with a feeding bottle (Figure 3(a) and 3(c)), which indeed are very complementary products. The pair with the highest ES are two different brands of nipple products (Figure 3(a) and 3(b)), which are clearly very close substitutes. This is very reassuring.

Another finding suggests an economic regularity not featured in textbooks. We compute the average elasticity of substitution (AES) for each product by averaging over all the Amazon Baby Registry products its final learned ES with every other product. We find that the popularity of a product in the dataset is strongly negatively correlated with AES. This means popular products have relatively smaller ES values, which suggests popular items tend to be systematically more complementary with other products.

More specifically, Figure 4 shows the logarithm of popularity of a product (y axis) against the AES of the product (x axis). The correlation between log($popularity$) and ES values is -0.916 for these products. Because we care more about the product ranking lists for recommendation rather than the absolute ES values in practice, we further rank the products according to ES and investigate the relation between log($popularity$) and the rankings (Figure 4, right). The correlation is -0.931. Further analysis shows that the products with small AES values in Figure 4 are mostly baby care necessities (e.g., pacifier, plug, and teether) that are generally complementary with many products, which makes them generally popular in most of the transactions.

These findings are encouraging and suggest that our proposed utility maximization approach conforms with economic intuition. It seems possible to discover product substitute

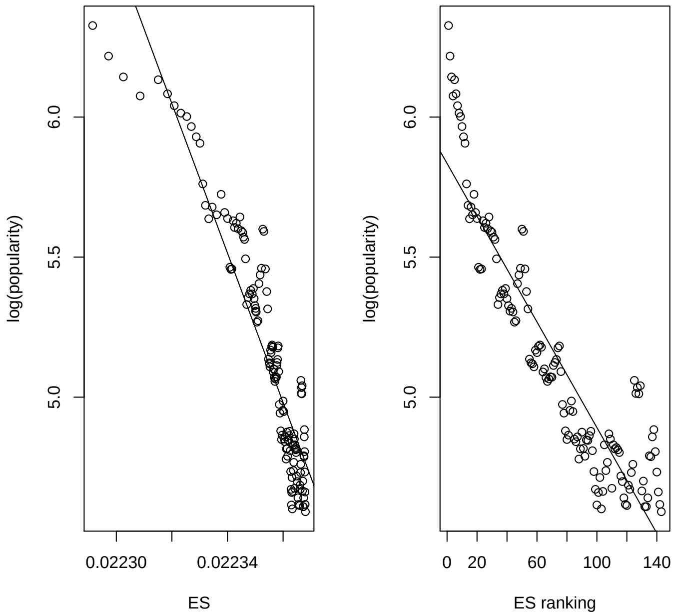


Figure 4: Scatterplots of product popularity vs. the average Elasticity of Substitution (ES) of the corresponding product as well as the the ranking of ES values.

or complementary relationships from real-world transaction data automatically, based on fusing machine learning techniques with economic principles.

6. CONCLUSIONS AND FUTURE WORK

Utility is commonly used by economists to characterize consumer preference over alternatives, and it serves as the cornerstone for consumer choice theory [12]. Motivated by existing research in economics, we introduced a general utility based framework for multiple product recommendation. Starting with the basic concept of Marginal Rate of Substitution (MRS) defined over product indifference curves, we derived several candidate utility function forms that can model both substitutability and complementarity. The model parameters are estimated based on observed consumer purchase data, and recommendations of multiple products are thus generated by maximizing the joint utility functions. Experimental results on both Amazon and Shop.com E-commerce data sets demonstrated the effectiveness of the proposed approach for recommendation. Further analysis also shows the underlying complementary and substitutability relations between products.

This is our first attempt toward multi-product utility modeling based on real-world purchasing records, and there is much room for further improvement. For example, the functional form of MRS could be adjusted to capture other products relationships beyond complementarity and substitutability. We can also introduce product and/or user features into this framework. In this work, we adopted a greedy method to generate top-K products, because the maximization of the utility function reduces to an integer linear programming problem which is NP-hard, and we will further investigate other heuristic methods that have been applied to 0-1 integer programming for better model learning accuracies. Besides, modeling the relationship between products is a fundamental problem for various recommendation tasks. Although we primarily focused on the most frequently concerned top-K recommendation task in this work, our proposed framework can be easily adapted for other usage scenarios such as package recommendation or next basket recommendation, which will be taken for further investigations in the future.

Acknowledgment

The authors thank the anonymous reviewers for the valuable reviews and constructive suggestions. This work was sponsored by the National Science Foundation under grant CCF-1101741 and IIS-0953908 of the United States, part of the work is sponsored by the Center for Intelligent Information Retrieval. Any opinions, findings, conclusions, or recommendations expressed in this paper are the authors', and do not necessarily reflect those of the sponsors.

Appendix

The key step for optimizing objective function Eq.(20) is to obtain the gradient of Eq.(5) w.r.t. $\Theta = \{\vec{x}_j, \vec{p}_j, \beta_j, \gamma_j, \alpha, \mu\}$. As Eq.(5) is the average of product pair utility $U(q_j, q_k)$, it is convenient to calculate the gradient per product pair utility $U(q_j, q_k)$ and aggregate to get the gradient of $U(\Omega_{it})$.

We first obtain the derivative of $U(q_j, q_k)$ w.r.t. the intermediate parameters a_{jk} and b_{jk},

$$
\begin{aligned}
\frac{\partial U(q_j,q_k)}{\partial a_{jk}} &= \frac{1}{1-b_{jk}}\left((1-a_{jk})q_j^{1-b_{jk}} + a_{jk}q_k^{1-b_{jk}}\right)^{\frac{b_{jk}}{1-b_{jk}}} \\
&\quad \left(q_k^{1-b_{jk}} - q_j^{1-b_{jk}}\right) \\
\frac{\partial U(q_j,q_k)}{\partial b_{jk}} &= U(q_j,q_k)\cdot\Bigg[\\
&\quad \frac{(a_{jk}-1)q_j^{1-b_{jk}}\log(q_j) - a_{jk}q_k^{1-b_{jk}}\log(q_k)}{(1-b_{jk})\left((1-a_{jk})q_j^{1-b_{jk}} + a_{jk}q_k^{1-b_{jk}}\right)} \\
&\quad + \frac{\log\left((1-a_{jk})q_j^{1-b_{jk}} + a_{jk}q_k^{1-b_{jk}}\right)}{(1-b_{jk})^2}\Bigg]
\end{aligned}
\tag{24}
$$

where a_{jk} and b_{jk} are functions of the model parameters Θ. We further obtain the gradient of the intermediate parameters w.r.t. the model parameters,

$$
\begin{aligned}
&(\nabla a_{jk})_{\vec{x}_j} = \sigma(.)'\vec{x}_k \qquad (\nabla a_{jk})_{\vec{x}_k} = \sigma(.)'\vec{x}_j \\
&(\nabla a_{jk})_{\alpha} = (\nabla a_{jk})_{\beta_k} = (\nabla a_{jk})_{\beta_j} = \sigma(.)'
\end{aligned}
\tag{25}
$$

where $\sigma(.) = 1/\left(1+\exp(-(\alpha+\beta_j+\beta_k+\vec{x}_j^T\vec{x}_k))\right)$ and $\sigma(.)'$ is the derivative of the Sigmoid function. Similarly, we can obtain the gradient for b_{jk},

$$
\begin{aligned}
&(\nabla b_{jk})_{\vec{p}_j} = \exp(.)\vec{p}_k \qquad (\nabla b_{jk})_{\vec{p}_k} = \exp(.)\vec{p}_j \\
&(\nabla b_{jk})_{\mu} = (\nabla b_{jk})_{\gamma_j} = (\nabla b_{jk})_{\gamma_k} = \exp(.)
\end{aligned}
\tag{26}
$$

where $\exp(.) = \exp(\mu + \gamma_j + \gamma_k + \vec{p}_j^T\vec{p}_k)$. The gradient of $U(q_j, q_k)$ w.r.t. Θ is available by applying the chain rule to the above results. Finally, we can thus optimize the objective based on commonly used Stochastic Gradient Descent (SGD) methods.

7. REFERENCES

[1] Y. K. Adomavicius, G. Toward more diverse recommendations: Item re-ranking methods for recommender systems. In *Proc. of the 19th Workshop on Information Technologies and Systems*, pages 657–666. ACM, 2009.

[2] E. Brynjolfsson, Y. J. Hu, and M. D. Smith. Research commentary—long tails vs. superstars: The effect of information technology on product variety and sales concentration patterns. *Info. Sys. Research*, 21(4):736–747, Dec. 2010.

[3] R. E. Bucklin and V. Srinivasan. Determining interbrand substitutability through survey measurement of consumer preference structures. *Journal of Marketing Research*, pages 58–71, 1991.

[4] J. Canny. Collaborative filtering with privacy via factor analysis. In *SIGIR*, pages 238–245, 2002.

[5] P. Castells, J. Wang, R. Lara, and D. Zhang. Novelty and diversity in recommender systems. In *RecSys*, 2011.

[6] R. Catherine and W. Cohen. Personalized recommendations using knowledge graphs: A probabilistic logic programming approach. *RecSys*, 2016.

[7] K. Christidis, D. Apostolou, and G. Mentzas. Exploring customer preferences with probabilistic topic models. In *ECML*, 2010.

[8] W. Chu and S. Park. Personalized Recommendation on Dynamic Content Using Predictive Bilinear Models. *WWW*, 2009.

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

[10] J. S. Coleman and T. J. Fararo. Rational Choice Theory. *Nueva York: Sage*, 1992.

[11] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan. Collaborative Filtering Recommender Systems. *Foundations and Trends in Human-Computer Interaction*, 4(2):81–173, 2011.

[12] J. Gans, S. King, and N. G. Mankiw. Principles of Microeconomics. *Cengage Learning*, 2011.

[13] Z. Gantner, S. Rendle, and L. Schmidt-Thieme. Factorization Models for Context-/Time-Aware Movie Recommendations. *CAMRa*, pages 14–19, 2010.

[14] A. Gatzioura and M. Sanchez-Marre. A case-based recommendation approach for market basket data. *Intelligent Systems, IEEE*, 30(1):20–27, 2015.

[15] N. Hurley and M. Zhang. Novelty and diversity in top-n recommendation–analysis and evaluation. *Transactions on Internet Technology (TOIT)*, 10(4):14, 2011.

[16] J. McAuley, R. Pandey, J. Leskovec. Inferring networks of substitutable and complementary products. In *KDD*, 2015.

[17] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse Recommendation: N-dimensional Tensor Factorization for Context-aware Collaborative Filtering. *RecSys*, pages 79–86, 2010.

[18] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[19] P. Lops, M. De Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*. Springer, 2011.

[20] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008.

[21] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.

[22] J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM, 2013.

[23] D. McFadden. Econometric models for probabilistic choice among products. *Journal of Business*, pages S13–S29, 1980.

[24] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD Cup and Workshop*, 2007.

[25] M. J. Pazzani and D. Billsus. Content-Based Recommendation Systems. *The Adaptive Web LNCS*, pages 325–341, 2007.

[26] P. Pu, L. Chen, and R. Hu. A user-centric evaluation framework for recommender systems. In *RecSys*, 2011.

[27] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press, 2009.

[28] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820. ACM, 2010.

[29] F. Ricci, L. Rokach, and B. Shapira. Introduction to Recommender Systems Handbook. *Springer US*, 2011.

[30] M. C. Rios, C. R. McConnell, and S. L. Brue. *Economics: Principles, problems, and policies*. McGraw-Hill, 2013.

[31] G. J. Russell and R. N. Bolton. Implications of market structure for elasticity structure. *Journal of Marketing Research*, pages 229–241, 1988.

[32] G. J. Russell and A. Petersen. Analysis of cross category dependence in market basket selection. *Journal of Retailing*, 76(3):367–392, 2000.

[33] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. *ICML*, pages 880–887, 2008.

[34] B. Sarwar, G. Karypis, et al. Item-based collaborative filtering recommendation algorithms. *WWW*, 2001.

[35] J. B. Schafer, J. Konstan, and J. Riedl. Recommender systems in e-commerce. In *EC*, pages 158–166, 1999.

[36] Y. Shi, X. Zhao, J. Wang, M. Larson, and A. Hanjalic. Adaptive diversification of recommendation results via latent factor portfolio. In *SIGIR*, SIGIR '12, pages 175–184, New York, NY, USA, 2012. ACM.

[37] A. D. Shocker, B. L. Bayus, and N. Kim. Product complements and substitutes in the real world: The relevance of "other products". *Journal of Marketing*, 68(1):28–40, 2004.

[38] G. Takacs, I. Pilaszy, B. Nemeth, and D. Tikk. Investigation of Various Matrix Factorization Methods for Large Recommender Systems. *Proc. ICDM*, 2008.

[39] H. R. Varian. *Intermediate Microeconomics: A Modern Approach*. WW Norton & Company, New York, eighth edition, 2010.

[40] J. Wang and Y. Zhang. Utilizing marginal net utility for recommendation in e-commerce. In *SIGIR*, pages 1003–1012. ACM, 2011.

[41] P. Wang, J. Guo, and Y. Lan. Modeling retail transaction data for personalized shopping recommendation. In *CIKM*, pages 1979–1982, 2014.

[42] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng. Learning hierarchical representation model for nextbasket recommendation. In *SIGIR*, 2015.

[43] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma. Explicit Factor Models for Explainable Recommendation based on Phrase-level Sentiment Analysis. *SIGIR*, 2014.

[44] Y. Zhang, Q. Zhao, Y. Zhang, D. Friedman, M. Zhang, Y. Liu, and S. Ma. Economic recommendation with surplus maximization. *WWW*, pages 73–83, 2015.

[45] Q. Zhao, Y. Zhang, D. Friedman, and F. Tan. E-commerce Recommendation with Personalized Promotion. *RecSys*, pages 219–226, 2015.