RELATIVE PRICE RECOMMENDATION: IMPLEMENTING CATEGORY RELATIONS AND PRICE OF PRODUCTS TO MAP PREFERENCE

A Thesis

by

TAE JUN JEON

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	James Caverlee
Co-Chair of Committee,	Ruihong Huang
Committee Member,	Tie Liu
Head of Department,	Dilma Da Silva

August 2018

Major Subject: Computer Engineering

ABSTRACT

In an e-commerce setting, a successful recommender system needs to incorporate the needs of consumers based on previous purchases by users. Current recommender systems incorporate different features and images of products to recommend products to consumers. Interestingly, the price of an object is one of the biggest constraints that the user faces before making a purchase. However, there is a research gap in our understanding of how to incorporate price into recommender systems.

This thesis explores the price aspect of products and how to incorporate price into a relative price-based recommendation. This work is different from modern approaches that observe the price elasticity and price sensitivity of products and to understand consumer behavior. This thesis will highlight how price as a comparable feature can be used to understand consumer interests and how relative price can be used to help narrow down products a consumer will be interested in.

This thesis will initially observe the performance of classic models such as user recommender and latent factor models such as Probabilistic Matrix Factorization. Than I will combine the category of relationships based on an economic theory of substitutes and complements to improve the accuracy of currently used models. This framework will address the issues of the long-tail problem inherent in data distribution.

From testing with Amazon review data, it has been observed that my framework can levitate the long-tail problem inherent in the dataset. Combined with previous works on price sensitivity, my framework can be used to explain purchase strategies of consumers along with consumer interest.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of advisor Professor James Caverlee and co-advisor Professor Ruihong Huang of the Department of Computer Science and Computer Engineering and Professor Tie Liu of the Department of Electrical Engineering and Computer Engineering.

The data analyzed for Chapter 4 was provided by Professor Julian McAuley from University of University of California San Diego.

All other work conducted for the thesis was completed by the student, under the advisement of Professor James Caverlee of the Department of Computer Science and Computer Engineering.

Funding Sources

There are no outside funding contributions to acknowledge related to the research and compilation of this document.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Existing Recommender Systems

Recommender systems have assumed an important position as human interactions have moved to the Internet. In particular, modern recommender systems are critical to many areas including social media, video, news, and e-commerce. A recommender system is designed to help users navigate a wide variety of content, typically by modeling the preferences of individual users. The core recommendation task can then be broken down into (i) understanding the large amount of available content (e.g., social media posts, videos, products); and (ii) delivering recent or unknown contents to the user based on personal preferences [1, 2].

Traditional recommender systems typically rely on two strategies to tackle the recommender problem. The first strategy – the content filtering approach – relies on building a content profile of each item. For movies, a profile for each film may contain information about the genre, leading actors, or director [2]. After collecting the profiles of users and items, the recommender system connects users to contents with shared interests (e.g., recommending a movie based on having a director of a previously liked movie). The second strategy – the collaborative filtering method – relies on users' past transactions, toward identifying items that were liked by similar users (e.g., recommending a movie liked by a group of other users who have rated other movies in common with you). The collaborative filtering model can be divided into the neighborhood method and the latent factorization model. The neighborhood model relies on analyzing relationships between users and items based on dependence on similar users or items [3]. The latent factor model is a matrix factorization-based method that creates latent feature vectors for users and items to capture different rating factors [2].

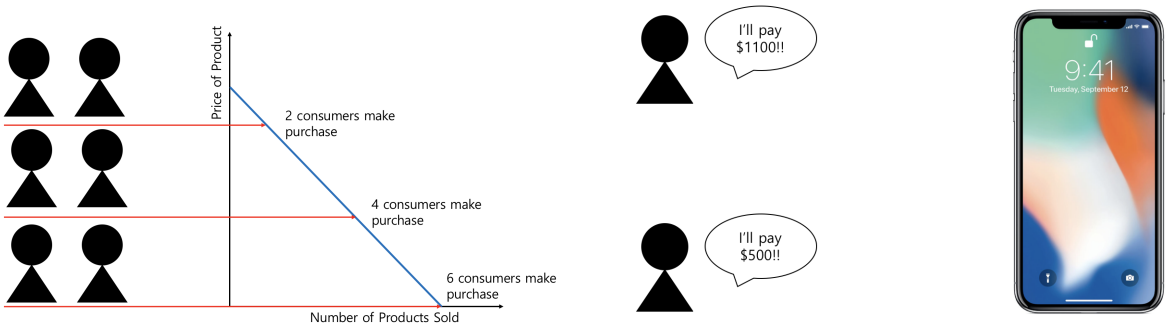
Many traditional recommender systems rely heavily on explicit data such as rating data to understand users' preferences, whereas modern recommender systems try to incorporate implicit data such as number of clicks to better understand user behaviors [4]. This is due to the fact that

explicit ratings are sometimes unreliable, and many users do not participate in rating the contents consumed. The general consensus is that successful modern recommender systems should consider multiple features when providing intuitive predictions.

1.1.1 Recommendations in E-commerce Setting

Recommender systems in e-commerce are slightly different than those in domains like social media, video, and news posts. While most recommender systems focus on personalized recommendations [2], a recommender system in e-commerce needs to consider the price of each item. For example, the above examples of social media and news posts do not require the user to pay money to connect to friends or read a news article. The cost of purchasing a movie ticket is the same regardless of the production cost of the movie. Users in the Netflix Challenge were paying a subscription fee-based membership that allowed them to watch a movie regardless of how successful the movie performed at the box office. In e-commerce, however, a user will not be able to purchase an item if he or she cannot afford it. This means that while the preference of the user is relevant, it may not be the determining factor that leads to a transaction.

While cost is a very important factor, there is a research gap in our understanding of how to incorporate price into recommenders. Currently, there are two perspectives of on going research in recommender systems in e-commerce using price. The first involves using the price sensitivity method. This concept is well studied in the field of economics and involves observing the change of demand in consumers when the price changes [5]. A visualization of price sensitivity is shown in Figure 1.1. Businesses use this concept in making personalized promotions to persuade users to consume products that are slightly out of budget [6]. The second method involves predicting the willingness of a consumer to pay for a certain product [7]. This is based on the concept that people have different perspectives toward putting value on the same objects. This concept is visualized in Figure 1.1.



(a) Visualization of Price Sensitivity

(b) Visualization of Willingness to Pay

Figure 1.1: Price sensitivity and willingness to pay

Most work on price in this context has focused on experiments and surveys with a small group of people. The most noteworthy work that was conducted on a large-scale dataset was proposed based on the economic concept of three-stage purchase decision [8]. This framework studies the stage of grocery category selection, and product choice along with the number of products purchased by customers. The primary insight comes from the use of price sensitivity for the purpose of understanding effective promotional strategies for customers. The framework concludes with the economic insight that personalized promotional strategies should be provided for customers in the product selection stage. While this economic insight is novel, grocery products have a tendency to be cheap, and the model depends heavily on periodic purchases by the users.

1.1.2 Relative Price Recommender System

This thesis proposes a new framework for incorporating price into recommendation that differs from previous studies. Instead of focusing on making personalized promotions, price can also be used as an indicator of consumers' interest. I hypothesize that consumers with specific interests will be willing to spend more money on related products. However, considering the magnitude of the money spent might not be an ideal method. Because different product categories have a tendency to have different price ranges, it is difficult to conclude that a consumer is more interested in electronic items or cars when a typical car will always be more expensive than electronics.

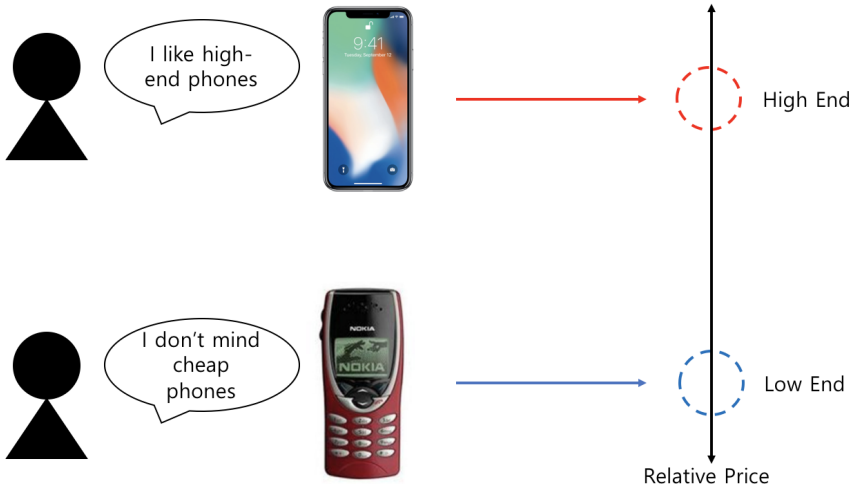
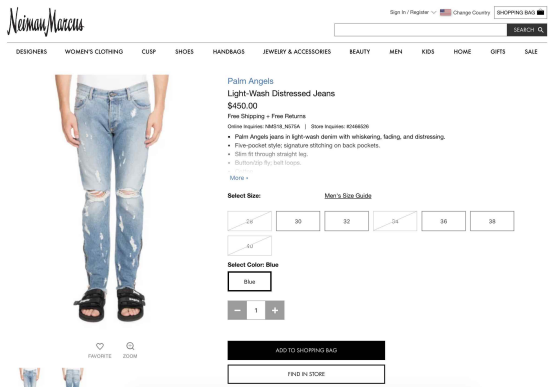


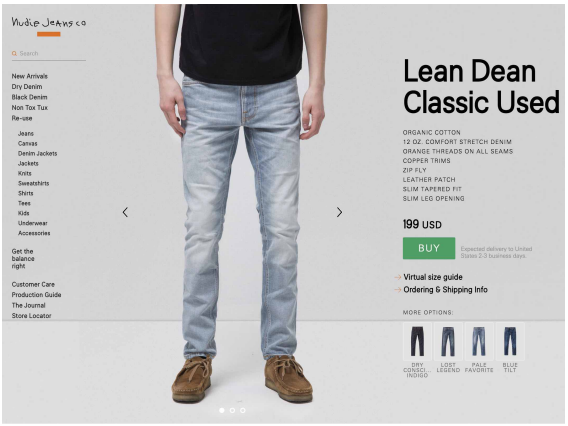
Figure 1.2: Visualization of relative price

Therefore, we need to consider the relative pricing of products within their associated categories to understand if they are considered relatively expensive or not. An example of relative price can be observed in Figure 1.2.

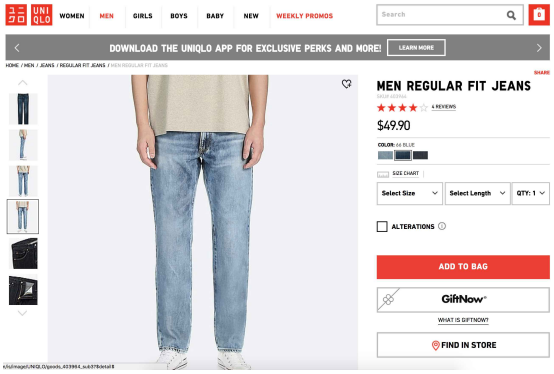
The proposed relative price can only be viable if the industry is well developed. A developed industry should contain multiple companies that strategically position themselves to target specific consumers [9]. This allows the user to have the necessary information to compare the different products that the companies in the industry provide and purchase ones that best fit his or her needs. A good example is the fashion industry.



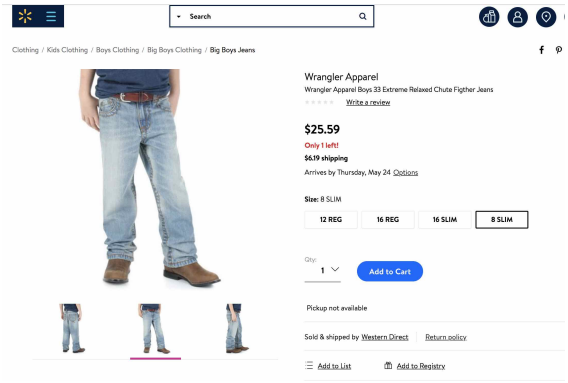
(a) Very expensive jeans by Palm Angels



(b) Expensive jeans by Nudie Jeans



(c) Medium-priced jeans by Uniqlo



(d) Cheap jeans by Walmart

Figure 1.3: Similar jeans with different price range

Figure 1.3 shows four different prices for jeans of a similar color. The quality and style of the jeans are different and the prices span a wide range. This is an example of the limitations of using visual images or relying on product relations when making recommendations [10, 11]. Without going through the trouble of analyzing images or crawling missing product features [10, 11], we can rely on the linear metric, price, to determine the preferences of users.

The challenge of utilizing relative price comes from the distribution of this relative price rating. Because cheaper products are purchased more often than expensive products, most users will have a low relative price purchase made on many categories, making it more difficult for the recommender system to predict higher relatively priced categories. This phenomenon, known as the

long-tail problem, is inherently caused by the tendency of users to purchase cheaper products.

For this thesis, I will be using an Amazon dataset that contains product and their accompanying reviews. I explore the use of relative price with the dataset with three baselines based on traditional models and test how the long-tail problem affects the initial models. Then, to address the long-tail problem, I propose a model that incorporates product relations when making recommendations. This allows the model to take into account product categories that are related to the target category when making a purchase. This has shown to be more effective in making predictions for higher relative-priced purchases.

The overall contributions of this thesis are listed below:

- Propose a method that utilizes price as a comparable variable in making recommendations;
- Embed relative price into traditional recommender system;
- Implement product relations to attack the long-tail problem faced in using relative price;
- Explore relative price in meta-data of Amazon Electronics.

The rest of the thesis is structured as follow. In Section 2, I will go over the baseline models used to compare the effects of long-tail distribution on traditional models. Section 3 presents the proposed model that alleviates the long-tail problem. Section 4 will cover the dataset explanation, filtering, and distribution observation. The experiment and results are discussed in Section 5, followed by the conclusion and future work directions in Section 6.

2. BASELINE RECOMMENDATION MODELS

Most previous recommender systems have ignored price as a factor, making it difficult to identify a baseline to compare with. However, the nature of converting actual price into a relative price makes it possible to treat relative price as if were a rating. This allows the collaborative filtering model to be used to predict the preferences of users in e-commerce setting. In this section, I will go over the three baselines that I have determined to test how traditional and modern models perform when facing the long tail problem.

Before going to the details of the baselines, I will go over the definitions of relative price and price level. The following is the definition of relative price:

$$r_j = \frac{p_j - min(C_j)}{max(C_j) - min(C_j)} \tag{2.1}$$

where r_j is the relative price of product j, p_j. C_j is the category product j belongs to, $p_j \in C_j$. In this formula, the price of product,j , is divided by, the maximum priced in the category to get the relative position of product j. The minimum price value is subtracted in the numerator and denominator to bind the relative price to [0,1].

The following is the definition of price level:

$$b_j = \lfloor \frac{r_j}{0.1} \rfloor \tag{2.2}$$

where b_j is the price level obtained from the relative price r_j. In this formula, the relative price is divided into 10 groups. This is necessary to visualize the performance of the models in which I use relative price as the main measurement of users' preference in a category.

2.1 Predict Most Popular Price Level

The first baseline is the naive approach of predicting the price level with the most frequent purchase for individual categories. Because the overall distribution of the relative price is skewed

towards cheaper products, it is necessary to set up this baseline to see if traditional models can outperform intuitive predictions.

The following is the equation for the first baseline:

$$b^{(C_i)} = \max_{b_k \in C_i} freq(b_k) \tag{2.3}$$

Here, $b^{(C_i)}$ represents the predicted price level of each category, C_i. $b_k \in C_i$ represents each price level in each category, and $freq(b_k)$ represents the frequency of price level b_k. This equation yields the price level with the highest frequency for each category.

2.2 User Based Recommender System

The second baseline represents the classical memory based collaborative filtering method. User based recommender system have been a popular approach for recommenders [12]. The algorithm was modeled on the assumption that consumers with similar interests will consume similar content. With a strong assumption that consumer preference will not change, past transactions can be used to predict future consumption. The transactions of all pairs of consumers will be compared to identify groups of similar products. This list of consumers will be used to make predictions for current consumers.

2.2.1 Model

The first step of the user based recommender system is to calculate the similarity between consumers and find k nearest neighbors. While there are different similarity measurements, I will be using cosine similarity for the user based recommender system. Equation 2.4 is the cosine similarity formula along with notations explained in Table 2.1.

$$sim(x, y) = \frac{\sum_{s \in C_{xy}} r_{x,c} * r_{y,c}}{\sqrt{\sum_{c \in C_x} r_{x,c}^2} * \sqrt{\sum_{c \in C_y} r_{y,c}^2}} \tag{2.4}$$

Equation 2.4 takes in the input of user's relative price for different categories. The numerator is the product of the relative price of a same category purchase and the denominator represents

Notation	Description
$r_{x,c}$	Rating of consumer x on category c
$r_{y,c}$	Rating of consumer y on category c
C_{xy}	Category that both consumers consumed

Table 2.1: Notation for user based recommender system

the values obtained from all the purchases users have maid. The output will represent how similar these two users are based on similar category purchases and relative prices.

The second step is to calculate the predicted category rating based on the average of the relative price by the neighbors. The equation is as follows:

$$r_{x,c} = \frac{\sum_{y \in T_{x,c}} sim(x,y) * r_{y,c}}{\sum_{y \in T_{x,c}} sim(x,y)} \tag{2.5}$$

where $r_{x,c}$ is the predicted relative price of user x on category c. $T_{x,c}$ represents the top k similar users, neighbors, of user x who made purchase for category c. Equation 2.5 represents the weighted sum of neighbors' ratings. The weight is multiplied by the ratings of the neighbors and divided by the sum of all the weights of the neighbors' to the target user. This way the ratings of the most similar neighbors will be valued more than that of the neighbors with less value.

2.3 Probabilistic Matrix Factorization (PMF)

The third baseline is the probabilistic matrix factorization model. This model tries to perform matrix decomposition as SVD but it has some advantages that make it more appealing. PMF performs its calculations based on the non-zero elements of the matrix which allows better performance in a sparse setting. It also has the advantage of computing in linear time [13].

2.3.1 Model

Figure2.1 represents the probabilistic graphical model of PMF (Table 2.2 explains the variables in the model). The PMF model captures the interaction of user and category with a probabilistic approach. PMF depends on the concept that if the given data has a Gaussian distribution prior

Notation	Description
b	Number of consumers
c	Number of categories
r_{ij}	Rating of consumer i on category j
$U \in r^{db}$	Latent consumer feature matrices with dimension d
$V \in r^{dc}$	Latent category feature matrices with dimension d
I^{ij}	Indicator function equal to 1 if consumer i made purchase for category j

Table 2.2: Notations for PMF

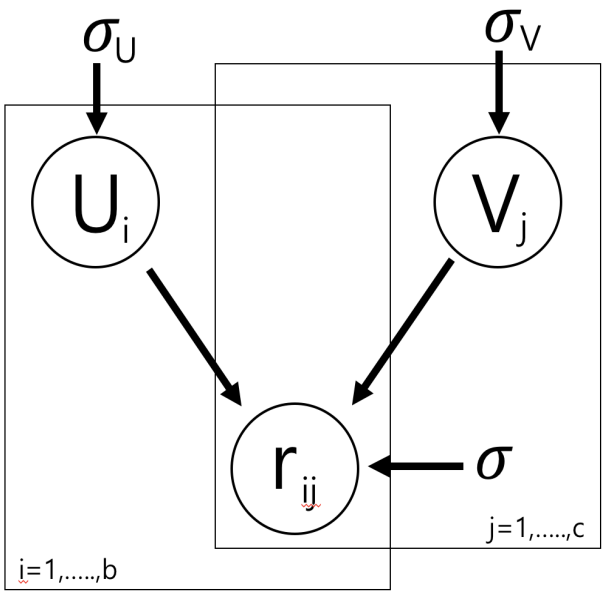


Figure 2.1: Graphical representation of PMF

with mean centered at zero, this problem can be treated like a matrix factorization problem trained with L2-loss with regularization. The following equation is the conditional distribution of the transactions of the relative price.

$$p(r|U,V,\sigma^2) = \prod_{i=1}^{b}\prod_{j=1}^{c}[\mathcal{N}(r_{ij}|g(U_i^T * V_j),\sigma^2)]^{I_{ij}} \tag{2.6}$$

$\mathcal{N}(x|\mu,\sigma^2)$ is the probability density function of the Gaussian distribution with mean μ and variance σ^2. $g(x) = \frac{1}{1+exp(-x)}$ represents the logistic function binding the predicted ratings to

[0:1]. Zero-mean spherical Gaussian priors were placed on consumer and category feature vectors as shown in equations 2.7:

$$p(U|\sigma_U^2) = \prod_{i=1}^{b} \mathcal{N}(U_i|0, \sigma_U^2 I)$$
$$p(V|\sigma_V^2) = \prod_{j=1}^{c} \mathcal{N}(V_j|0, \sigma_V^2 I) \tag{2.7}$$

Taking the log of the posterior distribution over both consumer and category features is shown as follows:

$$lnp(U, V|r, \sigma^2, \sigma_V^2, \sigma_U^2) = -\frac{1}{2\sigma^2} \sum_{i=1}^{b} \sum_{j=1}^{c} I_{ij}(r_{ij} - g(U_i^T V_j))^2 - \frac{1}{2\sigma_U^2} \sum_{i=1}^{b} U_i^T U_i$$
$$-\frac{1}{2\sigma_V^2} \sum_{j=1}^{c} V_j^T V_j - \frac{1}{2}((\sum_{i=1}^{b} \sum_{j=1}^{c} I_{ij}) ln\sigma^2 + bd * ln\sigma_U^2 + cd * ln\sigma_V^2) + C \tag{2.8}$$

The log of the posterior distribution can be transformed into a equivalent objective function of minimizing the sum of squared errors. The new objective function will contain quadratic regularization terms.

$$L(r, U, V) = \frac{1}{2} \sum_{i=1}^{b} \sum_{j=1}^{c} I_{ij}(r_{ij} - g(U_i^T V_j))^2 + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_V}{2} \|V\|_F^2 \tag{2.9}$$

$\|\|_F^2$ represents the Frobenius norm. The above loss function was optimized by performing gradient descent on the latent features of consumer and category.

In this section, I will explain my proposed model based on including category relations to PMF to improve prediction. This method was first implemented to improve predictions by implementing social information of target user in making prediction [14]. I channel this method but use category relations that I define instead of users in making predictions. The following subsections will cover the overall structure of the proposed model. I will explain category relations, how category relations are obtained, and go over the intuition behind it.

3.1 Model

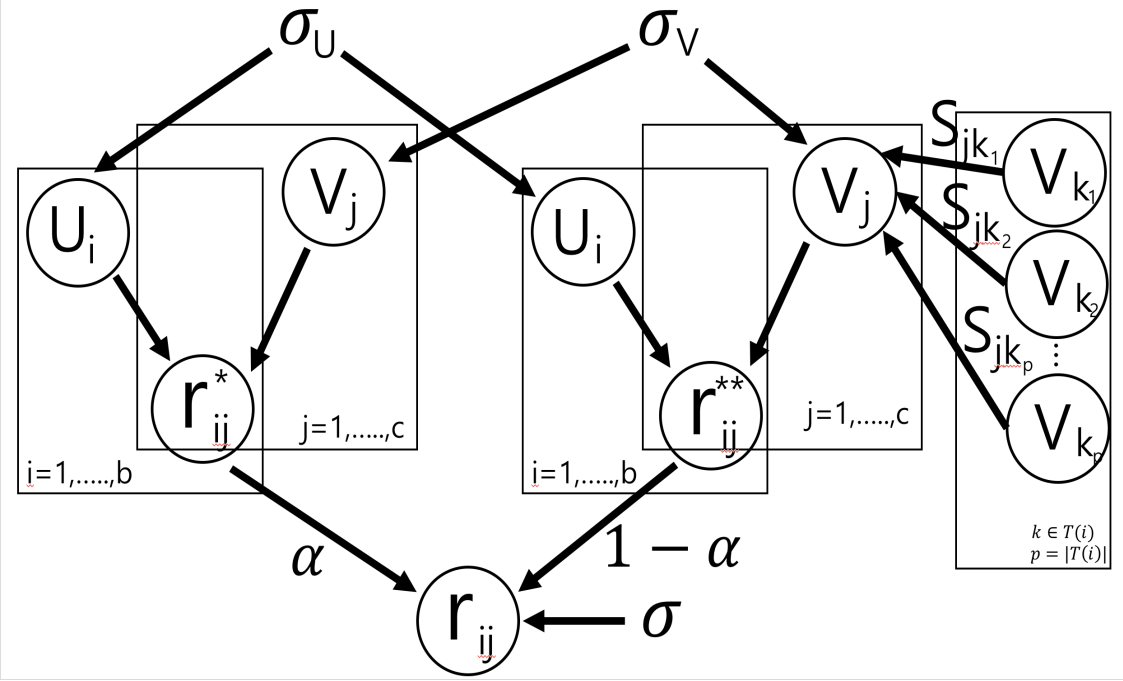


Figure 3.1: Graphical representation of proposed model

Figure 3.1 is the graphical representation of the proposed model. The overall structure is similar to that of the standard PMF from Figure 2.1. The main difference is that the proposed model

takes into account the interaction of categories, as observed in the right side of Figure 3.1. By capturing the consumer's interaction with categories related to the target category, the model can better predict the relative model using the ensemble model of the PMF. In following subsection, I will explain the portion of the model that explains category relations.

3.1.1 Category Interaction

The assumption of category interaction is that although the consumer has his/her own preference for a targeted category, there is a good chance that it can be overshadowed by the consumer having multiple purchases that are relatively cheap in multiple categories. The interaction of related categories can help mitigate the influence of constantly making cheap predictions.

Category Relations

In this section, I will use Amazon product connections to define two types of product relations. The two different types of product relations are defined as follow:

Relations	Amazon tags
Substitutes	Also Viewed, Buy After Viewing
Complements	Also Bought, Bought Together

Table 3.1: Product relations

The categorization of product relations is based on the relationship of competition and accommodation of products. Products in a substitute relationship have a tendency to be in competition with each other whereas complements have a tendency to create synergy when used together. These two types of relationships will create two different product networks, because the nature of the relationships are different.

Suppose product p_a represents all products in target category V_a. Each product p_a may have a product relationship of either substitute or complement with another product p_b. Let T_a be a set that includes all p_b within V_a. Take the number of common products of the related products set,

T_a, and target category V_b and divide by the total number of products in target category V_b. This relation can be expressed as the following equation:

$$S_{V_a,V_b} = \frac{|T_a \bigcap V_b|}{|V_b|} \tag{3.1}$$

The obtained category relationship will be used to compute the ratings of consumer i in category j as follows:

$$\bar{R}_{i,j} = \frac{\sum_{k \in Q(j)} R_{i,k} S_{j,k}}{|Q(j)|} \tag{3.2}$$

$\bar{R}_{i,j}$ is the rating of consumer i in category j and $Q(i)$ represents the categories that are related to the target category category j. The obtained ratings can be interpreted as consumer preference for related categories.

The conditional distribution of observed ratings is defined as follows:

$$p(R|S,U,V,\sigma_R^2) = \prod_{i=1}^{b} \prod_{j=1}^{c} [\mathcal{N}(r_{ij}|S_{ij}U_i^T V_j, \sigma^2)]^{I_{ij}} \tag{3.3}$$

$\mathcal{N}(x|\mu,\sigma^2)$ is the probability density function of the Gaussian distribution with mean μ and variance σ^2. In this model, I also place zero-mean spherical Gaussian priors on consumer and category feature vectors like for PMF.

The log of the posterior distribution over the proposed model is shown as follows:

$$\begin{aligned} lnp(U,V|r,\sigma^2,\sigma_V^2,\sigma_U^2) = &-\frac{1}{2\sigma^2}\sum_{i=1}^{b}\sum_{j=1}^{c} I_{ij}(r_{ij} - g(\alpha U_i^T V_j + (1-\alpha)\sum_{k \in Q(j)} S_{j,k} U_i^T V_k))^2 \\ &-\frac{1}{2\sigma_U^2}\sum_{i=1}^{b} U_i^T U_i - \frac{1}{2\sigma_V^2}\sum_{j=1}^{c} V_j^T V_j - \frac{1}{2}((\sum_{i=1}^{b}\sum_{j=1}^{c} I_{ij}) ln\sigma^2 + bd * ln\sigma_U^2 + cd * ln\sigma_V^2) + C \end{aligned} \tag{3.4}$$

The loss function is obtained that incorporates the category relations as an ensemble model:

$$L(r, S, U, V) = \frac{1}{2}\sum_{i=1}^{b}\sum_{j=1}^{c} I_{ij}(r_{ij} - g(\alpha(U_i^T V_j) + (1-\alpha)\sum_{k\in Q(j)} S_{jk}U_i^T V_k))^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2 \tag{3.5}$$

4. AMAZON PRODUCT DATA ANALYSIS

This section will explain the dataset and data filtering performed for the experiments. This section will be divided into two subsections. I will first introduce the dataset along with the data filtering procedure. Then, I will go over the distribution of the data so that I can provide more context for the challenges faced in my proposed approach. For experiments, I needed a dataset that contains consumers who made purchases as diverse as those shown in Figure 1.3. I located a dataset of electronic devices from Amazon that contained matching reviews [15].

4.1 Details of the Data

The electronics data from Amazon contains reviews crawled from Amazon along with product descriptions. Because Amazon does not provide personal user data for research, I make the assumption that people who leave comments on a product have made a purchase. It is important to note that only users who have purchased a specific product can leave a review. The following subsections will discuss the different features in the dataset.

4.1.1 Review Data

Features	**Description**
reviewerID	ID of the reviewer
asin	ID of the product
reviewerName	name of the reviewer
helpful	helpfulness rating of the review
reviewText	text of the review
overall	rating of the product
summary	summary of the review
unixReviewTime	time of the review (unix time)
reviewTime	time of the review (raw)

Table 4.1: Features in review data

```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "helpful": [2, 3],
  "reviewText": "I bought this for my husband who plays the
piano.  He is having a wonderful time playing these old hymns.
The music  is at times hard to read because we think the book
was published for singing from more than playing from.  Great
purchase though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```


Figure 4.1: Sample review

Review data are used to link consumers with the products they have purchased. Table 4.1 shows the features of a review, and Figure 4.1 illustrates what a typical review looks like. For the purpose of this paper, I will only be using the reviewID, asin, and the unixReviewTime features. Features such as helpful and rating will not be considered because I only need to understand whether a consumer made a purchase.

4.1.2 Product Data

Features	**Description**
asin	ID of the product
title	name of the product
price	price in US dollars
imUrl	url of the product image
related	also bought, also viewed, bought together, buy after viewing
salesRank	sales rank information
brand	brand name
categories	list of categories the product belongs to

Table 4.2: Features in product data

Product data are used to obtain the necessary features to group products into a subcategory. Table 4.2 shows the features of a product, and Figure 4.2 illustrates what a typical product looks

```
{
  "asin": "0000031852",
  "title": "Girls Ballet Tutu Zebra Hot Pink",
  "price": 3.17,
  "imUrl": "http://ecx.images-
amazon.com/images/I/51fAmVkTbyL._SY300_.jpg",
  "related":
  {
    "also_bought": ["B00JHONN1S", "B002BZX8Z6", "B00D2K1M3O",
"0000031909", "B00613WDTQ", "B00D0WDS9A", "B00D0GCI8S",
"0000031895", "B003AVKOP2", "B003AVEU6G", "B003IEDM9Q",
"B002R0FA24", "B00D23MC6W", "B00D2K0PA0", "B00538F5OK",
"B00CEV86I6", "B002R0FABA", "B00D10CLVW", "B003AVNY6I",
"B002GZGI4E", "B001T9NUFS", "B002R0F7FE", "B00E1YRI4C",
"B008UBQZKU", "B00D103F8U", "B007R2RM8W"],
    "also_viewed": ["B002BZX8Z6", "B00JHONN1S", "B008F0SU0Y",
"B00D23MC6W", "B00AFDOPDA", "B00E1YRI4C", "B002GZGI4E",
"B003AVKOP2", "B00D9C1WBM", "B00CEV8366", "B00CEUX0D8",
"B0079ME3KU", "B00CEUWY8K", "B004FOEEHC", "0000031895",
"B00BC4GY9Y", "B003XRKA7A", "B00K18LKX2", "B00EM7KAG6",
"B00AMQ17JA", "B00D9C32NI", "B002C3Y6WG", "B00JLL4L5Y",
"B003AVNY6I", "B008UBQZKU", "B00D0WDS9A", "B00613WDTQ",
"B00538F5OK", "B005C4Y4F6", "B004LHZ1NY", "B00CPHX76U",
"B00CEUWUZC", "B00IJVASUE", "B00GOR07RE", "B00J2GTM0W",
"B00JHNSNSM", "B003IEDM9Q", "B00CYBU84G", "B008VV8NSQ",
"B00CYBULSO", "B00I2UHSZA", "B005F50FXC", "B007LCQI3S",
"B00DP68AVW", "B009RXWNSI", "B003AVEU6G", "B00HSOJB9M",
"B00EHAGZNA", "B0046W9T8C", "B00E79VW6Q", "B00D10CLVW",
"B00B0AVO54", "B00E95LC8Q", "B00GOR92SO", "B007ZN5Y56",
"B00AL2569W", "B00B608000", "B008F0SMUC", "B00BFXLZ8M"],
    "bought_together": ["B002BZX8Z6"]
  },
  "salesRank": {"Toys & Games": 211836},
  "brand": "Coxlures",
  "categories": [["Sports & Outdoors", "Other Sports",
"Dance"]]
}
```


Figure 4.2: Sample product

like. For the purpose of the this paper, I will only be using the Amazon Standard Identification Number (asin), price, related products, and categories features.

4.1.3 Data Filtering

The above two sections discussed features from the dataset. This section will go over logistics and the need for performing preprocessing. Many reviews and product information tend to be noisy, with products not having all features in Table 4.2. The following preprocessing was implemented to reduce the noise within the data.

Price

For the purpose of the thesis, I will only observe products with price features included. The price feature was observed along with the associated product category to obtain the relative price. It is important to note that the price was crawled in a single instance.

Time

The review data contain reviews from 1998. This is not ideal, considering that the prices of products have a tendency to decrease. To reduce the effects of this phenomenon, I only observed the most recent two years of review data that were available by dataset. Therefore, reviews from 2012 to 2014 were observed.

Diverse purchases by consumer

Last, I reduced sparsity by only observing consumers who have made diverse purchases. This means that a consumer needed to have purchased a minimum of eight different categories to be considered for my experiments. Table 4.3 shows the logistics after performing the filtering.

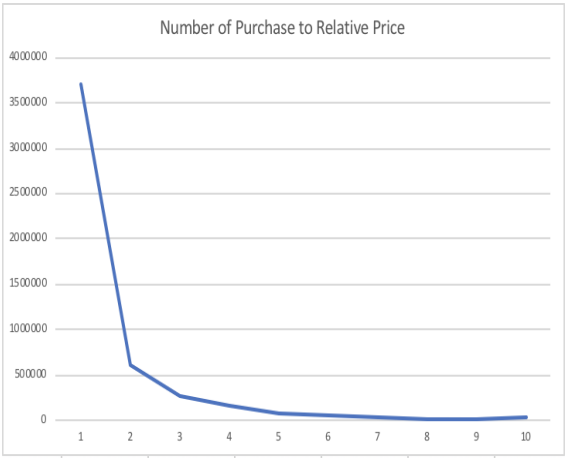
	Before Filtering	After Filtering
Number of Reviews	7,824,482	617,297
Number of Products	498,196	389,693
Number of Consumers	4,824,482	30,629
Number of Categories	772	772

Table 4.3: Logistics before and after filtering

4.2 Distribution

The purpose of this thesis is to see if price and relative price can be used to make recommendations. It is important to address the distribution of the products in terms of actual price and relative price. Figure 4.3 shows the distribution the number of purchases based on actual price and relative price.

(a) Number of Purchases of Products to Actual Price

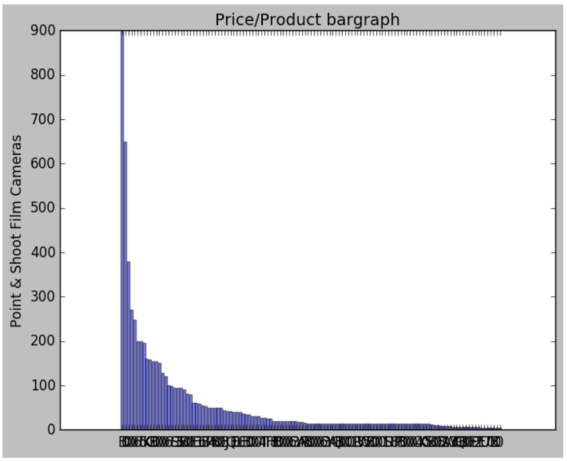


(b) Number of Purchases of Products to Relative Price

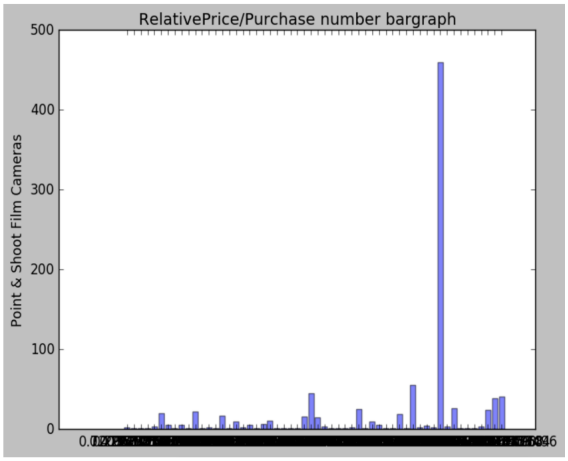
Figure 4.3: Distribution based on actual price and relative price

The distribution for both actual price and relative price graph in Figure 4.3 is known as a long-tail distribution. The majority of purchases occur for a select few product range which makes it difficult for any recommender systems to make predictions for the tail range. Although different approaches were used in movie recommenders, with popular movies had the majority of the views, in this case, the movies were clustered in the tail section to increase the number of views [16]. My problem is slightly different because the rating is skewed. Previous works are unhelpful in this regard.

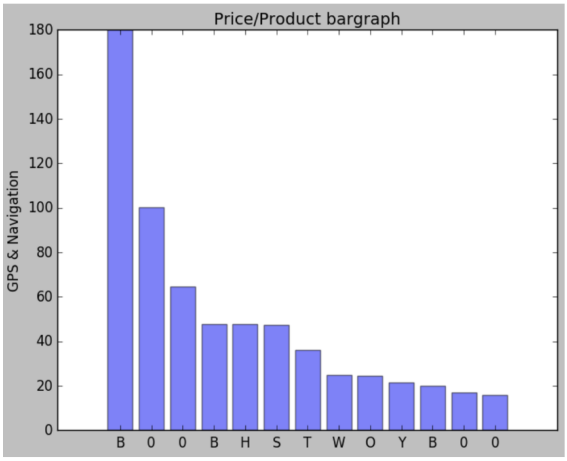
I have shown that the distribution of entire products is skewed towards cheap products, regardless of whether actual price or relative price is shown. However, the distribution of purchases for each category might be different. Figure 4.4 shows sample distributions of two categories. The product distribution graph shows the distribution of products by price. The x-axis represents individual products. For both Point and Shoot Film and GPS Navigation categories, the distribution is long-tail. The most expensive product is less observable than the cheaper products. The graphs of the number of purchases of products show that different categories have different distributions of purchases. For some categories, purchasing the cheapest product is shown to be less desirable for consumers.



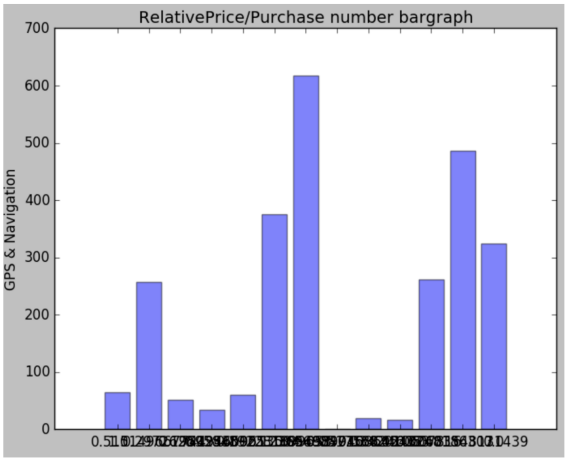
(a) Product distribution for point and shoot film cameras sorted by price



(b) Number of purchases of products for point and shoot film cameras sorted by price



(c) Product distribution for GPS and navigation sorted by price



(d) Number of purchases of products for GPS and navigation sorted by price

Figure 4.4: Sample distribution of products for price and purchases by subcategory

In the previous sections, I have discussed the different models used in this thesis. In the following, I will go into detail for the experiments for each baseline and the proposed model and what turned out to be the best performance. In the final subsection, I will go over the predictions made for the best setup for each model and explore how they fare in comparison with each other.

5.1 Predict Most Popular Price Level

The baseline for predicting the most popular price level does not have a different setting like the other baselines. The following figure is the confusion matrix for making the prediction of the most frequent price level. Rows 0 to 3 show that the overall prediction for rows 0 to 3 is acceptable. Consider that predicting 0 is not too far from predicting 3. However, this most frequent prediction performs poorly in ranges above the lower price level allowing too many price level 0 predictions for the higher price levels.

	0	1	2	3	4	5	6	7	8	9
0	21255	300	167	28	8	8	5	0	1	4
1	3069	925	173	19	5	6	11	0	0	1
2	1325	359	197	69	6	17	0	1	0	3
3	553	152	114	109	2	18	2	0	0	3
4	301	91	38	43	11	22	1	0	1	0
5	217	51	43	32	11	48	0	0	1	0
6	138	22	5	23	5	19	17	0	0	1
7	135	27	9	12	11	8	0	1	1	1
8	93	12	8	13	1	0	0	0	1	0
9	74	125	11	5	4	5	0	1	1	14

Figure 5.1: Confusion matrix for predicting the most popular price level

5.2 User Based Recommender System

In this section, the only parameters that are changed are the k factor that determines the number of neighbors observed in making the predictions. Figure 5.2 shows the RMSE results for varying

k values and Figure 5.3 shows the confusion matrix results for k=10 and k=100.

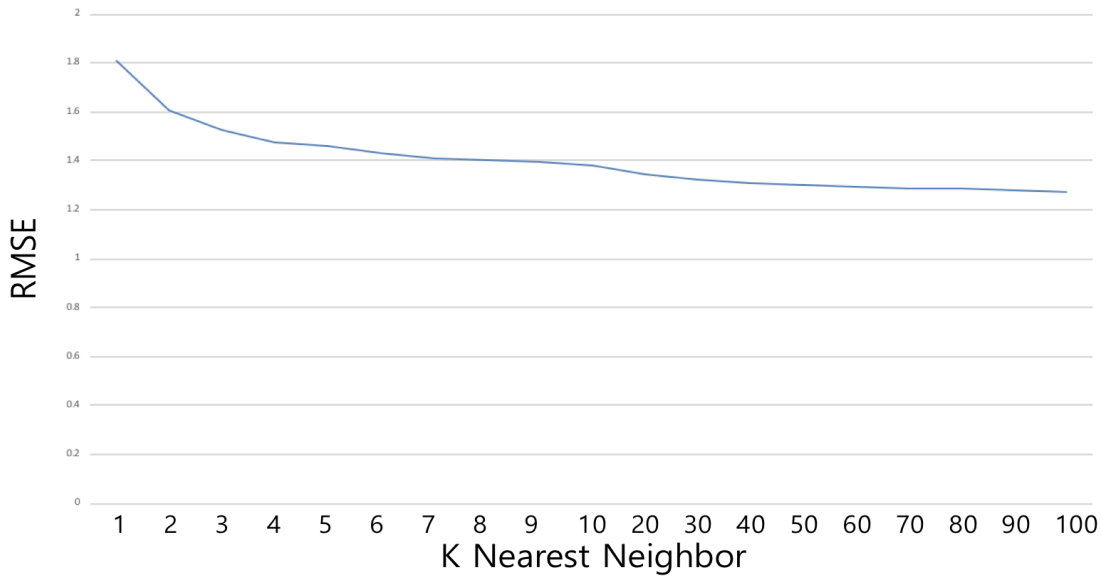


Figure 5.2: RMSE of user based recommender system with varying k

	0	1	2	3	4	5	6	7	8	9
0	16366	3961	842	296	149	79	35	21	14	13
1	1261	1767	706	240	117	53	31	11	8	15
2	368	717	392	202	172	61	22	15	10	18
3	140	195	223	167	124	51	14	14	8	17
4	59	98	104	85	80	47	9	9	7	10
5	30	48	65	81	79	50	18	13	8	11
6	16	23	26	46	56	29	10	5	7	12
7	13	20	34	44	36	26	12	7	4	9
8	8	16	20	25	26	19	6	4	3	1
9	16	16	21	33	35	45	35	19	7	13

(a) User recommender when k=10

	0	1	2	3	4	5	6	7	8	9
0	16762	3927	690	191	135	37	17	10	3	4
1	1216	1931	733	175	83	38	17	11	5	0
2	384	724	438	182	170	30	20	12	11	6
3	116	224	269	128	151	32	12	10	8	3
4	54	114	112	64	104	31	14	5	6	4
5	25	58	64	90	94	35	19	5	8	5
6	13	22	39	35	71	21	12	5	8	4
7	7	30	28	44	57	14	8	8	6	3
8	9	12	25	32	33	5	6	2	4	0
9	14	19	27	31	34	58	35	12	3	7

(b) User recommender when k=100

Figure 5.3: Confusion matrix for k=10 and k=100

Figure 5.2 shows a sharp decrease in RMSE from k=1 to k=3 and a continuous trend of small decreases in RMSE as k becomes greater. This may be caused by higher k values performing better at the lower price levels. Having a larger k value allows improved performance in the lower price level due to the generalization. Considering the skewed distribution, it is easier to improve RMSE by predicting abundant low price level occurrences than by predicting scarce high price level occurrences. Figure 5.3 shows this trend. The user recommender when k=100 shows a higher performance at price levels under 5 whereas the user recommender when k=10 shows a higher performance for upper price levels.

5.3 Probabilistic Matrix Factorization

PMF depends on multiple factors such as learning rate, lambda and others. I have experimented with different features, and I concluded that the parameters in Table 5.1 performed the best for the PMF model. The confusion matrix of PMF is presented in Figure 5.4.

Parameter	Value
feat	12
learning rate	6
lambda	0.0001
epoch	200
batch number	200
batch size	1000

Table 5.1: PMF parameters

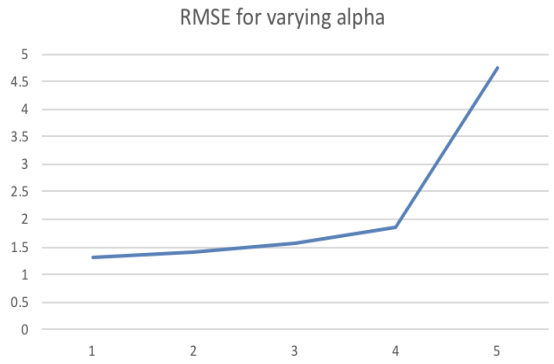
	0	1	2	3	4	5	6	7	8	9
0	16527	4643	370	108	61	28	15	8	10	6
1	1403	2156	444	110	39	24	15	11	5	2
2	434	925	309	150	67	40	26	16	7	3
3	117	394	187	101	57	41	26	12	9	9
4	48	191	109	54	47	28	15	4	7	5
5	28	127	65	62	48	30	24	11	7	1
6	11	51	58	37	25	22	8	11	5	2
7	9	60	45	34	20	9	9	13	2	4
8	3	37	29	19	14	7	11	2	1	5
9	10	42	28	38	30	28	20	19	14	11

Figure 5.4: Confusion matrix of PMF

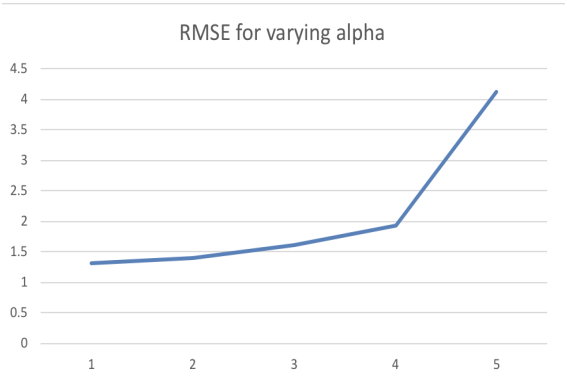
While the high accuracy for price level 0 is notable, it is easily noticeable that the PMF may predict price level of 1 for many of its prediction tasks. This may be due to the probabilistic foundation of PMF. The distribution skewed towards the low price level has quite an influence on false predictions for higher price levels.

5.4 Proposed Model

The proposed model requires fine-tuning of parameters from PMF model and the α variable. The α determines how much the user should influence the relative price, this is also known as the target category interaction. For the proposed model, I tested the performance of the model by varying the α variable value while keeping the other parameters constant as shown in Table 5.1. Figure 5.5 shows the performance of varying alpha values along with the confusion matrix of the best performing alpha.



(a) Performance of proposed model with substitute relations with varying alpha



(b) Performance of proposed model with complement relations with varying alpha

	0	1	2	3	4	5	6	7	8	9
0	10088	10502	667	264	126	66	19	25	18	1
1	783	2451	597	210	92	40	22	11	2	1
2	202	995	348	181	125	58	26	32	9	1
3	61	352	217	100	106	58	33	22	3	1
4	25	166	95	86	58	37	15	12	13	1
5	11	92	68	64	60	32	39	21	12	4
6	3	43	31	42	54	25	10	7	9	6
7	1	45	23	29	33	28	26	15	3	2
8	3	22	20	28	26	18	4	5	2	0
9	3	23	37	22	31	39	40	28	14	3

(c) Substitute relations with alpha at 0.75

	0	1	2	3	4	5	6	7	8	9
0	11535	7490	1291	761	454	178	25	35	7	0
1	1155	2049	482	209	164	110	20	7	10	3
2	397	846	264	196	89	75	27	76	5	2
3	113	376	135	130	63	60	29	46	1	0
4	45	182	67	60	37	36	18	52	9	2
5	20	116	53	61	34	33	37	32	15	2
6	11	37	23	20	36	24	36	30	9	4
7	9	50	17	34	19	28	12	30	3	3
8	6	35	12	8	14	19	10	23	1	0
9	5	30	18	15	21	27	15	16	54	39

(d) Complementary relations with alpha at 0.5

Figure 5.5: Confusion matrix of proposed model

5.5 Experimental Analysis

In this section, I compare the results obtained from the baselines. Two experiments were performed to observe the performance and results of the baseline and proposed model. The first experiment examined the performance of the models by dividing the test set by the price level. This was done to examine how individual models perform in different price level predictions and comparisons. The second experiment examined the performance of the models for a category with purchases from a variety of price levels.

5.5.1 Price Level Observation

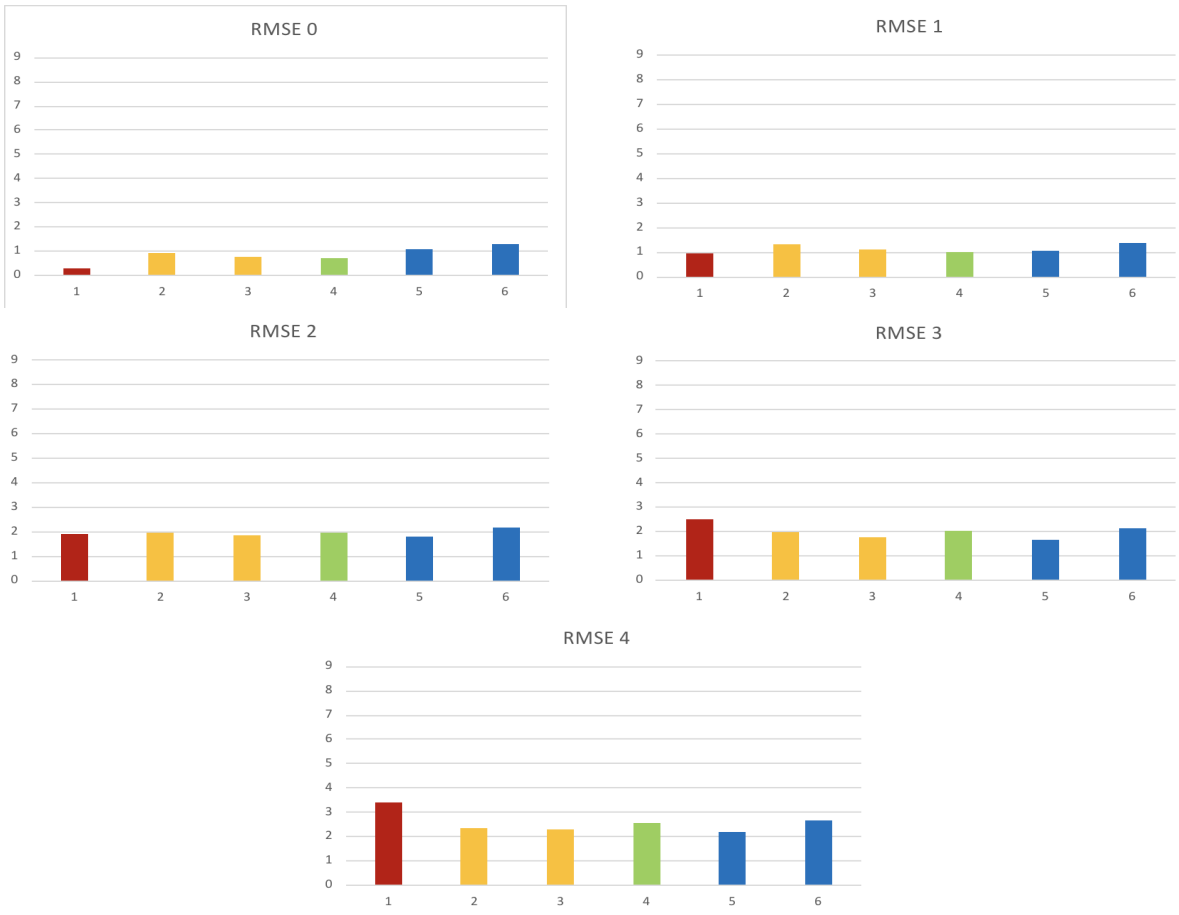


Figure 5.6: Performance of models at lower price levels

In the following section, I examine individual model performance by dividing the test set by its price level. The low price group was made up of purchases with price levels of zero to five, and high price group as five to nine price levels. Figure 5.6 shows individual performance of the models. Each sub-figure contains the performance of each model in RMSE, with low RMSE meaning better performance. Each model performance is color-coded according to its original model. For the lower price levels, the overall performance looks similar. While the naive baseline of predicting the most frequent price level, the red bar, has its performance lead in price level 1 and 0, its performance worsens noticeably as the price level increases. The user recommender systems, the yellow bar, performs well in all price levels with the least amount of impact as the price level increases. PMF, the green bar, performs well, matching the performance of the user recommender systems. The proposed model with substitute relations performed better than the proposed model with complements relations.

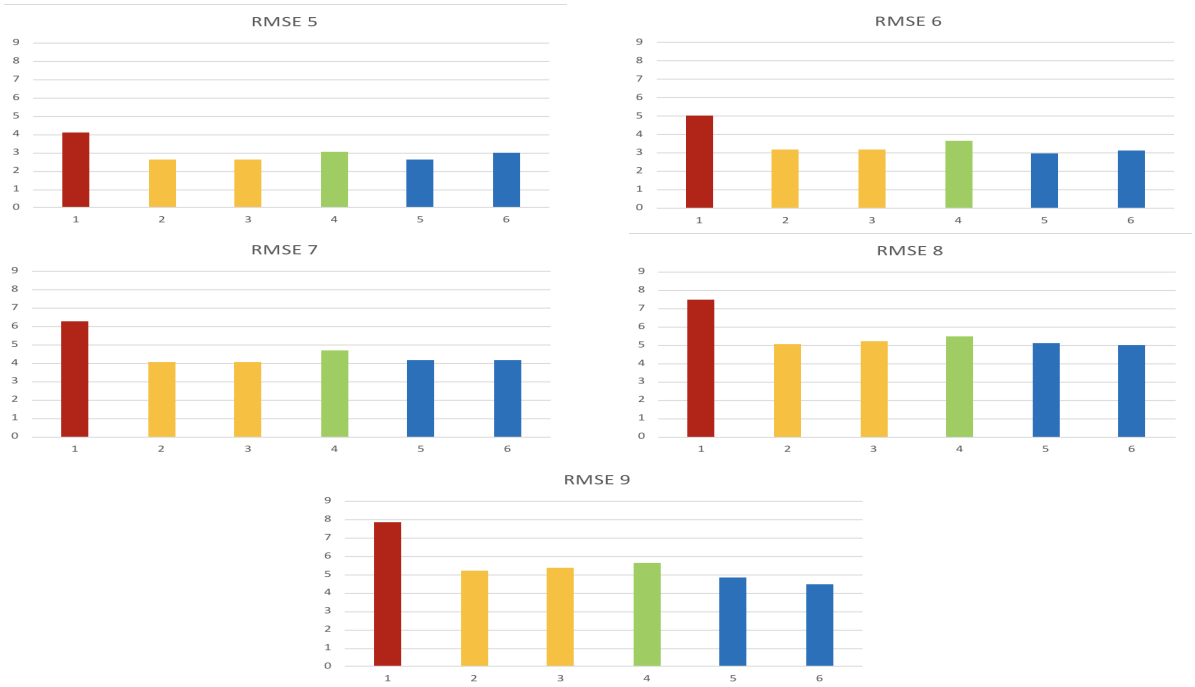
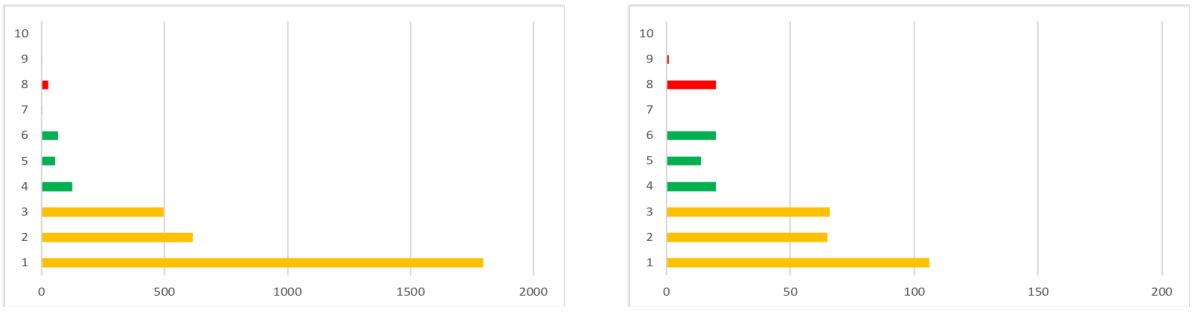


Figure 5.7: Performance of models at higher price levels

Figure 5.7 shows the performance of the models in the price level of five and above. The naive baseline shows very poor performance for the high price levels. The user recommender models fairly well in the higher price levels, outperforming the PMF and proposed model. The user recommender with k=10 starts to outperform k=100 for the higher priced levels. This may be because k=10 generalizes less than that of k=100 allowing k=10 to make more predictions using smaller number of neighbors. The PMF and the proposed model suffered less than the naive baseline but showed sub-par performance for the very high price levels. However, the proposed models have been shown to outperform the PMF model for the higher price levels. Proposed model with substitute relations has been shown to have good performance for the higher price levels, with the proposed model with complement relations outperforming the proposed model with substitute relations for price levels of eight and nine.

5.5.2 Specific Category Observation

In the previous section, I discussed the overall performance of the individual models. However, it is unclear how each of models can be used to make recommendations for each category. In this section, I observe the performance of each model for categories that have transactions spanning low to high price levels. For the purpose of the experiment, I have chosen the category: Graphic Card.



(a) Graphic card train set distribution

(b) Graphic card test set distribution

Figure 5.8: Distribution of test and train set of graphic card

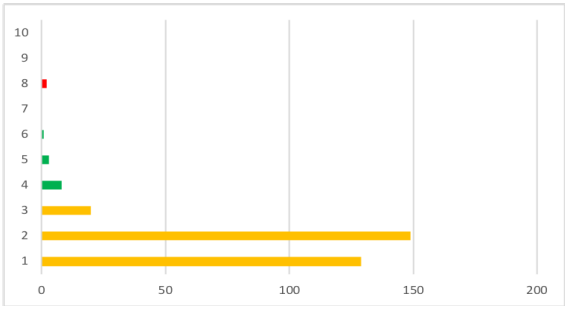
Figure 5.8 shows the distribution of both test and train set. The graphic card category covers all areas of the price level clustered in different price levels. It is interesting to note that there is a good representation of high price levels of 8 existing in the test set.

First I will examine the prediction distribution of the user recommender in Figure 5.9. The user recommender in general makes good predictions throughout all price levels. For k=10, it is noticeable that it makes more predictions in high price levels of 8. The overall spread and frequency matches the distribution of the test set. For k=100, predictions were made in the middle price levels. This is due to the generalization that k=100 makes. Having to take into account more neighbors allows the model to generalize relative prices more.

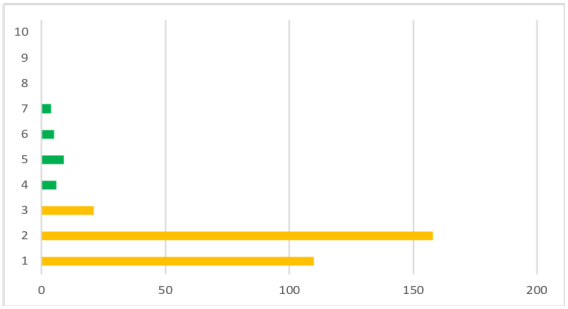
(a) User recommender k=10 prediction distribution (b) User recommender k=100 prediction distribution

Figure 5.9: Prediction distribution for user recommender for graphic card

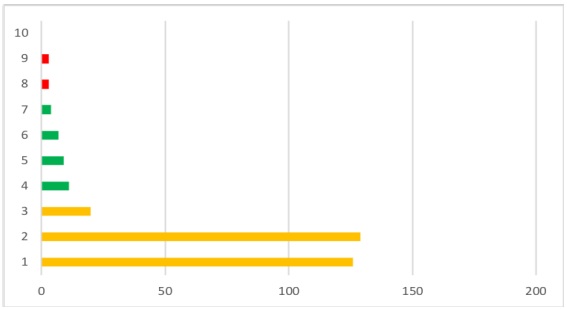
Examine the prediction distribution for PMF and the proposed model. Figure 5.10 shows prediction results for the PMF and the proposed model. For the PMF model, predictions made are skewed towards lower price levels. Even though the test set shows a good representation in the middle and high price levels, the PMF model was not successful in capturing them. For the proposed models, we can observe predictions increasing toward the middle and high levels. For the complement, it is surprising to see good representations in the higher price levels even more than the user recommender had. Interestingly enough, the proposed model with complement relations performed better in the higher price levels than that of the proposed models with substitute rela-



(a) PMF prediction distribution



(b) Proposed model with substitute relations prediction distribution



(c) Proposed model with complement relations prediction distribution

Figure 5.10: Prediction distribution for PMF-based models for graphic card

tions.

5.5.3 Overall Performance

	Most Popular	User Rec k=10	User Rec k=100	PMF	Proposed sub	Proposed comp
RMSE	1.486	1.384	**1.274**	1.311	1.385	1.56
Low PL	**0.91**	1.17	1.02	1.0	1.18	1.39
High PL	5.95	3.90	3.96	4.36	3.86	**3.78**
Graphic Card	x	2.09	**1.61**	2.49	2.55	2.71

Table 5.2: Overall RMSE

Table 5.2 shows the overall RMSE for the different models. The columns represent the indi-

vidual models and the rows represent the different cases under consideration. The items in bold represent the best-performing models for the particular cases divided into rows. The first row represents the RMSE value to observe all the test cases. For this particular case, user recommender outperformed all the models. This is because user recommender had a generally better performance in predicting low and high price levels. However, if the train set is divided into high and low price levels, we see that for low price level predictions the naive baseline performed the best and the proposed model using complement relations performed the best for the high price levels. This suggests that including product relations can improve predictions for niche price levels.

6. CONCLUSION

The purpose of this thesis was to investigate price as a potential feature in making recommendations. Motivated by the fact that money is one of the greatest constraints for consumers to consummate transactions, I have proposed an alternative way to approach this problem using meta-data from Amazon. I propose a new concept of relative price to observe how expensive a product is based on the price of products in the associated category. Using this implicit feature and converting it to a rating form, I explored relative price as a suitable rating to understand consumer purchase behavior using traditional recommender systems.

I observe that the long-tail distribution skewed towards the cheaper relative price, and the traditional models of PMF and user recommendations have performed reasonably well. The user recommender models have surprisingly outperformed the matrix factorization-based PMF model. After I performed the experiment, it was clear that the PMF model has suffered more from the long-tail distribution than the user recommender model has. This shortcoming can be mitigated by using the product relations network based on the economic theory of substitutes and complements. The proposed model with substitute relations has been shown to perform well in predicting low price levels, whereas the proposed model with complement relations has done well in the high price levels.

6.1 Further Study

For further study, it is crucial to understand why substitutes and complement relations make better predictions for different price levels. While the substitute-based models have performed better for the overall prediction price levels, the complement-based models performed better in the difficult challenge of predicting the high price levels. Further, there are other approaches beyond the matrix factorization methods at the center of this thesis that can be augmented with relative price information. How well do these alternative methods perform? There are also other techniques – like clustering for grouping different categories by their price levels – that could give

additional insights into the impact of relative price on recommendation.

REFERENCES

[1] J. Bennett, S. Lanning, *et al.*, "The netflix prize," in *Proceedings of KDD cup and workshop*, vol. 2007, p. 35, New York, NY, USA, 2007.

[2] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, 2009.

[3] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The adaptive web*, pp. 291–324, Springer, 2007.

[4] G. Li and Q. Chen, "Exploiting explicit and implicit feedback for personalized ranking," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[5] D. A. Ackerberg, "Advertising, learning, and consumer choice in experience good markets: an empirical examination," *International Economic Review*, vol. 44, no. 3, pp. 1007–1040, 2003.

[6] P. J. Danaher, M. S. Smith, K. Ranasinghe, and T. S. Danaher, "Where, when, and how long: Factors that influence the redemption of mobile phone coupons," *Journal of Marketing Research*, vol. 52, no. 5, pp. 710–725, 2015.

[7] Y. Jiang and Y. Liu, "Optimization of online promotion: a profit-maximizing model integrating price discount and product recommendation," *International Journal of Information Technology & Decision Making*, vol. 11, no. 05, pp. 961–982, 2012.

[8] M. Wan, D. Wang, M. Goldman, M. Taddy, J. Rao, J. Liu, D. Lymberopoulos, and J. McAuley, "Modeling consumer preferences and price sensitivities from large-scale grocery shopping transaction logs," in *Proceedings of the 26th International Conference on World Wide Web*, pp. 1103–1112, International World Wide Web Conferences Steering Committee, 2017.

[9] P. Kotler and D. Gertner, "Country as brand, product, and beyond: A place marketing and brand management perspective," *Journal of brand management*, vol. 9, no. 4, pp. 249–261, 2002.

[10] W.-C. Kang, C. Fang, Z. Wang, and J. McAuley, "Visually-aware fashion recommendation and design with generative image models," *arXiv preprint arXiv:1711.02231*, 2017.

[11] J. McAuley, R. Pandey, and J. Leskovec, "Inferring networks of substitutable and complementary products," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, ACM, 2015.

[12] G. Adomavicius and A. 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*, vol. 17, no. 6, pp. 734–749, 2005.

[13] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, pp. 1257–1264, 2008.

[14] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 203–210, ACM, 2009.

[15] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *proceedings of the 25th international conference on world wide web*, pp. 507–517, International World Wide Web Conferences Steering Committee, 2016.

[16] Y.-J. Park and A. Tuzhilin, "The long tail of recommender systems and how to leverage it," in *Proceedings of the 2008 ACM conference on Recommender systems*, pp. 11–18, ACM, 2008.