Does Product Recommendation Meet its Waterloo in Unexplored Categories? No, Price Comes to Help

Jia Chen[†], Qin Jin[*], Shiwan Zhao[♮], Shenghua Bao[♮], Li Zhang[♮], Zhong Su[♮], Yong Yu[†]

[†]Dept. of Computer Science & Engineering, Shanghai Jiao Tong University, China
{chenjia, yyu}@apex.sjtu.edu.cn
[*] School of information, Renmin University of China, China
qjin@ruc.edu.cn
[♮]IBM China Research Laboratory, Beijing, China
{zhaosw, baoshhua, lizhang, suzhong}@cn.ibm.com

ABSTRACT

State-of-the-art methods for product recommendation encounter significant performance drop in categories where a user has no purchase history. This problem needs to be addressed since current online retailers are moving beyond single category and attempting to be diversified.

In this paper, we investigate the challenge problem of product recommendation in unexplored categories and discover that the price, a factor transferrable across categories, can improve the recommendation performance significantly. Through our investigation, we address four research questions progressively: 1) what is the impact of unexplored category on recommendation performance? 2) How to represent the price factor from the recommendation point of view? 3) What does price factor across categories mean to recommendation? 4) How to utilize price factor across categories for recommendation in unexplored categories? Based on a series of experiments and analysis conducted on a dataset collected from a leading E-commerce website, we discover valuable findings for the above four questions: first, unexplored categories cause performance drop by 40% relatively for current recommendation systems; second, the price factor can be represented as either a quantity for a product or a distribution for a user to improve performance; third, consumer behavior with respect to price factor across categories is complicated and needs to be carefully modeled; finally and most importantly, we propose a new method which encodes the two perspectives of the price factor. The proposed method significantly improves the recommendation performance in unexplored categories over the state-of-the-art baseline systems and shortens the performance gap by 43% relatively.

*Qin Jin is the corresponding author

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.
Copyright 2014 ACM 978-1-4503-2257-7/14/07 ...$15.00.
http://dx.doi.org/10.1145/2600428.2609608 .

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Retrieval and Search—*Information Filtering*

Keywords

product recommendation; unexplored category; price

1. INTRODUCTION

Nowadays a mature E-commerce website has abundant categories of products that offer rich choices to users. However, users are normally used to browsing and purchasing in only a small portion of categories. Current recommendation systems tend to recommend products in categories with user purchase history rather than in categories without. This makes products in such categories without user purchase history less exposed to the user. Some users may even be unaware of these categories.

Behind this phenomena, there exists a common request, that is "**product recommendation in unexplored categories**". For a user, an "unexplored category" is defined as the category where he/she has never purchased any product before. In our dataset which consists of $92,519$ users and $146,524$ products from the leading E-commerce website in China, we observe that the majority users made purchase in less than 4 categories while the number of categories on this site is 761. Recommending products in unexplored categories not only can meet users' potential need, but also can help retailers to increase their sales on the long tail and expand their business scope. Therefore, addressing the problem of recommendation in unexplored categories benefits both customers and retailers simultaneously.

To the best of our knowledge, recommendation in "unexplored category" setting has not been well studied in previous work although it is very valuable to E-commerce. We observed that the performance of existing recommendation systems in unexplored categories drops significantly compared to that in explored categories. Based on our analysis, the performance drop is mainly caused by the fact that user preference on products is hard to transfer across different categories. After suverying several factors, we find out that price is a factor that is transferrable across categories. We choose price for several other reasons as well: it is one of the most important factors that users consider in making purchase decisions and it is available on almost all E-commerce websites.

Our investigation of recommendation in unexplored categoies in this paper addresses the following four research questions step-by-step:

R1-What is the impact of unexplored categories on recommendation performance? We find out that unexplored categories cause performance drop in current recommendation systems. Furthermore, the performance drop is universal for shopping orientations of both browsing and buying as defined by Mark Brown et al. [2].

R2-How to represent the price factor from the recommendation point of view? In order to achieve the best performance boost, we can consider the price factor from two perspectives: price as an attribute of products and price as user-preference. These two perspectives lead to two different representations of the price factor: a normalized percentage quantity and a distribution under one category.

R3-What does price factor across categories mean to recommendation? Does it mean that if one user prefers low-price products in the explored categories, he may also prefer low price products in an unexplored category? The answer is no. We discover that price-preference is extremely subtle with respect to both users and categories. Therefore the price factor across categories needs to be modeled carefully for recommendation.

R4-How to utilize price factor across categories for recommendation in unexplored categories? Our model embeds both perspectives of the price factor in a unified solution. We incorporate them to the standard matrix factorization formalizations to find out better user and product factors.

We evaluate our method in two shopping orientations: browsing and buying [2]. In the browsing orientation, users prefer to gather more information. Unexplored categories are recommended when a user logs in the E-commerce website. Once the user browses a recommended unexplored category, products in this category will be recommended. In the buying orientation, users prefer to make the decision quickly. Therefore products in unexplored categories are recommended directly for users to make a quick choice.

In both shopping orientations, we compare our method systematically to two state-of-the-art baselines : the implicit matrix factorization algorithm (WRMF) [7] and the regularization algorithm (REG) [9]. The experimental results show that our proposed method does improve the recommendation performance in unexplored categories significantly by embedding either one or both of the perspectives of the price factor. It reduces the performance drop in unexplored categories by 43% relatively.

The rest of the paper is organized as follows. Section 2 presents the related work. The first three research questions are answered through analyzing unexplored categories and the price factor from a recommendation point of view in Section 3. To answer the fourth research question, we first embed the two perspectives of price factor in a unified form in Section 4. In Section 5, we present a formal definition of the R4 question in two different shopping orientations and describe a unified solution by embedding either one or both of the perspectives of the price factor. Experimental results are presented in Section 6. Discussions of connection among the four research questions are described in Section 7. Conclusions are made in Section 8.

2. RELATED WORK

Our work is related to the research topic of novelty in recommendation. Most existing work on novelty is trying to give different definitions of novelty according to the nature of the task [5][20][8][10], and pursues the balance between novelty and accuracy, i.e., improving the novelty while keeping the accuracy as much as possible. Compared to the existing work, our recommendation in unexplored categories naturally increases the novelty of recommendation. Novelty is a natural property derived from the problem setting rather than an objective to pursue. The challenge in our problem is to reduce the performance drop in unexplored categories. Thus, we only focus on the accuracy related metrics.

Many factors affect user purchase behavior and can be leveraged to improve recommendation performance. Time factor is considered in work [19][17]. Chen [3] points out that user comment is another factor that affects user purchase behavior. However, there is little work addressing the price factor in E-commerce related recommendation task. To the best of our knowledge, the only related one is the exploration of price's marginal net utility role in E-commerce recommendation done by Wang and Zhang [16]. Their approach focuses on the balance between re-purchase product recommendation and un-purchased product recommendation. It doesn't address the unexplored category problem. Differently, we directly tackle the unexplored category problem in E-commerce recommendation. We explore and embed the two perspectives of price factor in recommendation, which is an entirely new view of role of price.

There are abundant algorithms on leveraging side information to boost the recommendation performance [13][4][1][12]. Porteous et al. [13] propose a Bayesian matrix factorization model embedding side information. Chen et al. [4] construct feature functions in matrix factorization process. Paterek [12] introduces regularization to embed additional information. The above mentioned algorithms focus on the explicit feedback, while E-commerce related applications usually have to be centered on the implicit feedback. There are also many implicit feedback models by adding side information. Ahmed et al. [1] propose a hybrid model that adds and smooths user preference across categories. Their Bayesian approach can only embed the price factor as a quantity. Our study shows that when price factor is considered as user preference, it turns to be a distribution not belonging to the exponential family. So this perspective of the price factor is not likely to be handled by their framework. Singh et al. propose joint matrix factorization [15] to add additional information. We build our model upon their approach to encode both perspectives of the price factor.

3. ANALYSIS OF UNEXPLORED CATEGORY AND PRICE FACTOR

In this section, we conduct some analysis of unexplored category and price factor to address the first three research questions we mentioned in previous section. To investigate the problem of recommendation in unexplored categories, it requires a dataset to contain E-commerce transaction records. We haven't noticed any public transaction dataset that contains price and category information. Therefore we collected a dataset from one of the largest E-commerce websites in China to fullfill such requirement. On this

lady pants(730)	sports shoes(905)	handbags(2, 636)
laptops(1, 030)	desktops(851)	appliance(1, 150)
outdoor(2, 873)	beauty(49, 701)	kitchen(730)
books(2, 090)	health(1, 783)	watches(1, 585)

Table 1: Example categories: the number in the bracket refers to the number of products in this category

E-commerce website, there are 761 categories. Some example categories are shown in Table 1.

We remove users that purchased products in only one category since we don't have ground truth for such users in evaluation. We also remove users that purchased more than 1, 000 unique products since these users are likely to be retailers rather than normal customers. The filtered dataset contains 1, 064, 865 user-product pairs of 92, 519 users and 146, 524 products spanning 44 common categories.

3.1 Unexplored Category Causes Performance Drop (Answer to R1)

In this subsection, we address the first research question R1: What is the impact of unexplored categories on recommendation performance?

Figure 1 plots the distribution of users' explored category number. We can see that most users only shop under one category and with the number of explored categories increases the number of users drops dramatically. The number of categories explored by each user is far less than the total shopping categories on the E-commerce website (Recall that the number is 761 on this site from which we crawled our dataset). This indicates that recommending products in unexplored categories is very necessary for both customers and retailers.

Next, we check how the state-of-the-art recommendation system [7] performs in unexplored categories. We randomly sample 30% (user, category) pairs and keep all the shopping records under these pairs apart as simulation data. The remaining 70% (user, category) pairs is basic training data. We further split the simulation data into four equal parts for each user. We randomly select one out of the four parts as test data, leaving the other three parts as non-test simulation data. We conduct expriments under four training conditions with different training data setup: 1) *unexplored* - the basic training data; 2) 25%*explored* - the basic training data plus one part from the non-test simulation data; 3) 50%*explored* - the basic training data plus two parts from the non-test simulation data; 4) 75%*explored* - the basic training data plus all three parts from the non-test simulation data; We evaluate the performance on the top 5 predicted products. The evaluation metrics are mean average precision (MAP), R-precision (RP) and precision at K ($p@K$)[1]. These metrics are also used to evaluate our own proposed method later in Section 6. The performance is shown in Figure 2(a). We can see that the performance in unexplored category condition drops significanly by approximately half compared to other explored conditions. The relative drop is 40% on MAP and 42% on $p@1$ between *unexplored* condition and 25%*explored* condition. Different explored conditions have similar performance trend and the performance is slightly improved if more data are provided in the explored categories.

[1] http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-ranked-retrieval-results-1.html

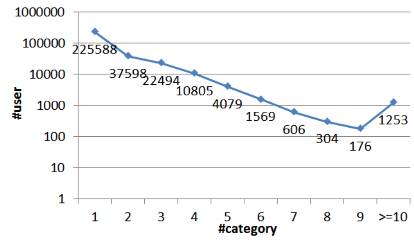


Figure 1: Distribution of users' explored category number

We can further breakdown the performance drop into two parts: drop caused by wrong category prediction (*typeI error*) and drop caused by wrong item prediction in the right category (*typeII error*). We measure the impact of these two types of errors on the recommendation performance separately. For typeI error, we consider all items in the groundtruth category as right answers (which means we only focus on the category prediction error). The result is shown in Figure 2(b). The relative performance drop caused by typeI error is around 21.55%. For typeII error, we only predict items in the groundtruth category (which means we only focus on the item prediction error). As shown in Figure 2(c), the performance in unexplored categories drops by nearly 13.43% relatively compared to other explored situations. Results in these two figures indicate that both typeI error and typeII error have impact on the recommendation performance.

The breakdown setting corresponds to the browsing orientation while non-breakdown setting in the previous paragraph corresponds to the shopping orientation. Based on above results, the answer to R1 is that unexplored categories causes significant performance drop for recommendation.

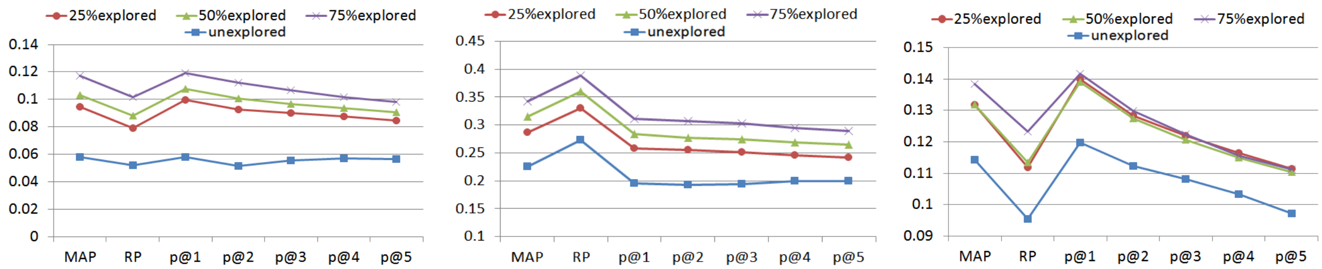
3.2 Price Factor is both a Quantity and a Distribution (Answer to R2)

In this subsection, we address the second research question R2: How to represent the price factor from the recommendation point of view?

We consider that price factor can be both a distribution and a quantity. If we talk about products, price is a quantity. If we talk about user preference, it is a distribution.

For products, price factor is a quantity: In this case, price level rather than price itself is transferrable across categories. It is meaningless to compare products in different categories by their absolute price value. For example, \$500 is a relatively low price under category *laptop* but a relatively high price under category *snack* though it is the same price value. Considering that different categories have different price intervals, one invariant measurement of price across categories is the price level p, which is defined as the percentage in the interval. For example, assuming that the price interval of category *laptop* is $[400, 2000]$ and that of category *snack* is $[0, 100]$, a *laptop* at price \$500 has the same price level $p = 6.25\%$ $(= \frac{500-400}{2000-400})$ as a bag of *snack* at price \$6.25. The value of price level ranges from 0 to 1.

For users, price factor is a distribution: For an explored category, a user usually buys more than one product. We gather the purchased products by their price level and get a histogram for a user in an explored category. As shown in Figure 3(a), each plot shows a specific user's price level histogram under some paticular categories. From the price level historgram, we can say that user 20363 is

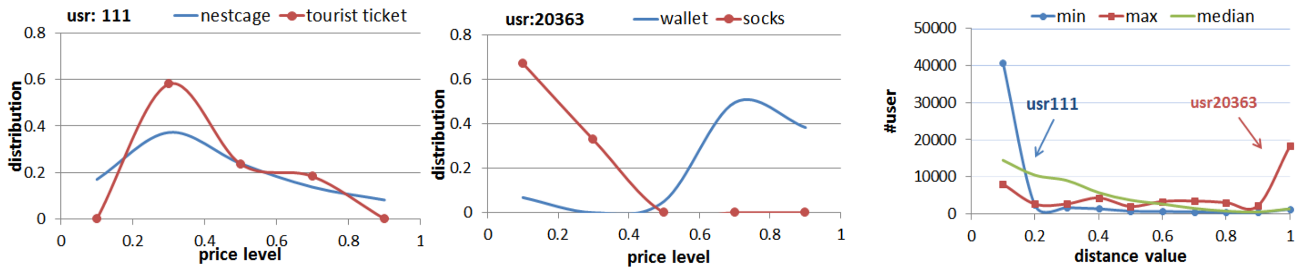


(a) Performance drop in unexplored categories

(b) Study of typeI error(wrong category prediction)

(c) Study of typeII error(wrong item prediction)

Figure 2: Comparison of unexplored and explored category situations



(a) Price distribution of shopping history for users under different categories: x axis is price level and y axis is the probability at each level.

(b) Coherence of price factor across categories for users

Figure 3: Price distribution under a category and between categories

likely to buy a nestcage at price level 0.3 but he may also buy a nestcage at price level 0.5. This means that for users, price is a distribution rather than a quantity. We could form the user's purchase behavior as a sample from an underlying price distribution, which is actually the user's price preference. As shown in Figure 3(a), from user to user and from category to category, the distribution changes in skewness, kurtosis and number of peaks. So it is difficult to describe it by a particular family of distribution such as exponential family.

Based on the above analysis, the answer to R2 is that the price factor should be represented as quantity or distribution or both depending on what perspective we take for recommendation.

3.3 Price Factor across Categories is Subtle (Answer to R3)

In this subsection, we address the third research question R3: what does price factor across categories mean to recommendation?

We first study whether a user will prefer low price products in an unexplored category if he prefers low-price products in the explored categories. We call this coherence issue and measure it by the distance between price distribution under different categories. A user's price preference is coherent if the price distribution under different categories is similar. For each user, we calculate χ^2 distance of price preference between all pairs of explored categories. For each user, we record the mininum, medium and maximum of all the pairwise χ^2 distances. We plot the histogram of these three values on all users in Figure 3(b). Majority users have very small value on minimum distance (80% users have the minimum distance less than 0.1) and very large value on maximum distance (40% users have the maximum distance larger than 0.9). The medium distance has a near uniform distribution on all users. The distribution of these three values indicates that for most users, price preference under different categories is not coherent. It is a subtle factor that needs to be carefully modeled. To make it easier to understand, we also map the two typical user examples we discussed earlier in Figure 3(a) to Figure 3(b). The distances between categories for user 111 and 20363 lie on the two ends of the distance value axis.

Based on above analysis, the answer to R3 is that user bahavior with respect to price factor across categories is really subtle and needs appropriate modeling for recommendation.

4. ENCODING THE TWO PERSPECTIVES OF PRICE FACTOR

As discussed in the last section, the price factor has the potential to help the recommendation system in unexplored category setting. However, the user behavior with respect to price factor across categories is subtle and should be carefully modeled. Before moving on to the modeling part, we need to design suitable representations of price factor from the two different perspectives. Recall that in the last section we show that price factor is a quantity from a product perspective, but it becomes a distribution from a user perspective. In this section we will encode both perspectives into matrix forms, namely product-price matrix **IP** and user-price matrix **UP**. In **IP** matrix, each row corresponds to a product and in **UP** matrix, each row corresponds to a user. Please note that the meaning of columns in these two matrices are totally different, reflecting the two different perspectives of the price factor.

4.1 Price as Quantity: Columns in IP

It is straightforward that price is a quantity attribute for a product. So we focus on how to encode it to columns in **IP**. We discretize the price quantity into bins and encode which bin it appears for each product. For each product, we get

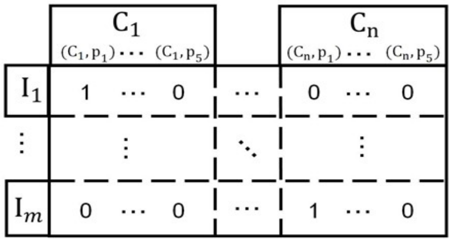


Figure 4: **IP** matrix: for each row, within one category $\mathbf{C}_c$, there is at most one element that is 1 since each product belongs to only one bin of price level in a category

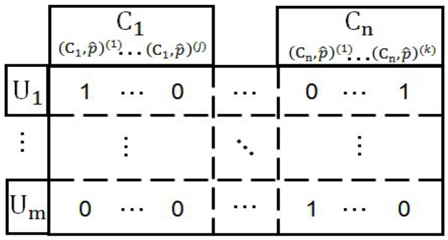


Figure 5: **UP** matrix: for each row, within one category $\mathbf{C}_c$, there is at most one element that is 1 since each user is assigned to only one typical price preference in one category

a binary vector on bins. To be specific, we slice price level equally to 5 bins $p_j, j \in \{1, \cdots, 5\}$ between 0 and 1. Directly aligning these vectors together loses category information. To preserve category information, we align products from the same category together. In the **IP** matrix, from left to right are price levels grouped by categories. In each category, there are five columns corresponding to five bins of price level. Thus each column is described by a pair (c, p_j) of the category c and the bin p_j of price level as shown in Figure 4. Each row has only one nonzero element since the price level falls into one and only one bin and each product belongs to one and only one category.

4.2 Price as Distribution: Columns in UP

We denote the price distribution as $(c, \hat{p})$ under category c of a user. The good property of this perspective is that we can directly compare two users' price preference, e.g. whether they are similar, what is the typical price perference pattern, etc. To use this property, we cluster to find out typical price patterns $(c, \hat{p})^{(t)}$ for each category among all users using affinity propagation [6]. Then we assign each $(c, \hat{p})$ with its nearest typical pattern ID. In the **UP** matrix, each column corresponds to a typical pattern and columns are grouped by categories as shown in Figure 5.

5. EMBEDDING TWO PERSEPECTIVES OF PRICE FACTOR IN TWO SHOPPING ORIENTATIONS

According to work [2], there are two shopping orientations for a user: buying and browsing. We notice that the unexplored category setting exists in both shopping orientations. This is another evidence that justifies the business value of solving the problem of recommendation in unexplored categories. In the browsing orientation, we recommend unexplored categories for users (Category Recommendation - CR problem) to extend the coverage of their browsing activities. In this stage, price as distribution perspective helps. Once the user browses a recommended unexplored category, products under the selected category will then be recommended for users (Category Item Recommendatoin - CIR problem). In this stage, price as quantity perspective helps. In the buying orientation, we recommend products in unexplored categories directly, without any need of category designation to generate revenue (Unexplored Category Item Recommendation - UCIR problem). Both price as distribution and as quantity perspectives come to help in this orientation. The CR problem and the CIR problem can be considered as a two stage decomposition of the UCIR problem, but they aim at different shopping orientations of users.

u	the index of a user
i	the index of a product
c	the index of a category
$\mathbf{C}$	the category vector. $\mathbf{C}_u^{explored}$ is the vector of explored categories for the user u. $\mathbf{C}_u^{unexplored}$ is the vector of unexplored categories for the user u.
$\mathbf{UI}$	the user-product matrix. The element $\mathbf{UI}_{ui}$ is a binary value indicating whether the user u has purchased the product i.
$\mathbf{UC}$	the user-category matrix. The element $\mathbf{UC}_{uc}$ is a binary value indicating whether the user u has purchase history under category c.
$\mathbf{UP}, \mathbf{IP}$	user-price and product-price matrix
$\mathbf{CF}^{(*)}$	The confidence matrix corresponding to matrix $*$, which can be $\mathbf{UC}, \mathbf{UI}, \mathbf{UP}, \mathbf{IP}$.

Table 2: Notations

5.1 Problem Formulation

To describe the formal definitions of these problems, we first introduce notations in Table 2. Please note that we use "product" and "item" interchangeably in this paper. As pointed out in work [7], the purchase history of a user is implicit feedback rather than explicit feedback, so our problem formalization is built upon the implicit feedback formazlization. When handling implicit data, a confidence level $\mathbf{CF}_{ui}^{(UI)}$ is associated with each user-product. Combining the confidence level with the classical matrix factorization, we get the following problem formalization:

$$\min_{\mathbf{X}^*, \mathbf{Y}^*} ||\mathbf{CF}^{(UI)} \odot (\mathbf{UI} - \mathbf{X}^\top \mathbf{Y})||_F^2 + \lambda(||\mathbf{X}||_F^2 + ||\mathbf{Y}||_F^2) \quad (1)$$

where $|| \cdot ||_F$ means Frobenius norm, $\odot$ is the element-wise product operation, $\mathbf{X}$ is user latent factor matrix and $\mathbf{Y}$ is product latent factor matrix. The $\lambda(||\mathbf{X}||_F^2 + ||\mathbf{Y}||_F^2)$ term is necessary for regularizing the model such that it will not overfit the training data. $\mathbf{X}$ and $\mathbf{Y}$ have the same number of rows (latent factor number). When predicting product i for user u, we take inner product of $\mathbf{X}_u$ and $\mathbf{Y}_i$. In this formalization, the purchase behavior is interpreted by two independent latent factors: user and product.

In our investigation, we exploit price factor to tackle the problem of recommendation in unexplored categories. The two different perspectives of price factor introduced in section 3 are encoded in **UP** and **IP** matrices as described in section 4. **UP** matrix describes the user price preference under a certain category while **IP** matrix describes the product's price level.

5.1.1 CR Problem with Price as Distribution

The **CR problem** (recommending unexplored categories): *Given the user-category matrix* $\mathbf{UC}$ *which captures the category purchase history, the system recommends categories from* $\mathbf{C}_u^{unexplored}$ *to user* u.

For the CR problem, we introduce user-price matrix $\mathbf{UP}$ (where price is a distribution) in addition to user-category matrix $\mathbf{UC}$. We interpret $\mathbf{UP}$ by latent user factor $\mathbf{X}$ and latent price factor $\mathbf{M}$. We interpret $\mathbf{UC}$ by latent user factor $\mathbf{X}$ and latent category factor $\mathbf{C}$. The latent user factor $\mathbf{X}$ is shared by $\mathbf{UC}$ and $\mathbf{UP}$. Writing the above formalization in the matrix factorization style, we get:

$$\begin{aligned}\min_{\mathbf{X}^*,\mathbf{Y}^*,\mathbf{M}^*} & (1-\alpha)||\mathbf{CF}^{(UC)} \odot (\mathbf{UC} - \mathbf{X}^\top \mathbf{C})||_F^2 + \\ & \alpha||\mathbf{CF}^{(UP)} \odot (\mathbf{UP} - \mathbf{X}^\top \mathbf{M})||_F^2 + \\ & \lambda(||\mathbf{X}||_F^2 + (1-\alpha)||\mathbf{C}||_F^2 + \alpha||\mathbf{M}||_F^2)\end{aligned} \tag{2}$$

where the parameter α is used to adjust the influence of $\mathbf{UC}$ and $\mathbf{UP}$ on the shared user latent factor $\mathbf{X}$.

5.1.2 CIR Problem with Price as Quantity

CIR problem (recommending products in a given unexplored category): *Given the purchase history* $\mathbf{UI}$*, category* c *from* $\mathbf{C}_u^{unexplored}$*, the system recommends products within category* c.

For the CIR problem, we introduce product-price matrix $\mathbf{IP}$ (where price is a quantity) in addition to user-product matrix $\mathbf{UI}$ under a specific category. We interpret $\mathbf{IP}$ by latent product factor $\mathbf{Y}$ and latent price factor $\mathbf{N}$. We interpret $\mathbf{UI}$ by latent user factor $\mathbf{X}$ and latent product factor $\mathbf{Y}$. The latent product factor $\mathbf{Y}$ is shared by $\mathbf{UI}$ and $\mathbf{IP}$. We form the CIR problem in the matrix factorization style:

$$\begin{aligned}\min_{\mathbf{X}^*,\mathbf{Y}^*,\mathbf{N}^*} & (1-\alpha)||\mathbf{CF}^{(UI)} \odot (\mathbf{UI} - \mathbf{X}^\top \mathbf{Y})||_F^2 + \\ & \alpha||\mathbf{CF}^{(IP)} \odot (\mathbf{IP} - \mathbf{Y}^\top \mathbf{N})||_F^2 + \\ & \lambda(||\mathbf{Y}||_F^2 + (1-\alpha)||\mathbf{X}||_F^2 + \alpha||\mathbf{N}||_F^2)\end{aligned} \tag{3}$$

where the parameter α is used to adjust the influence of $\mathbf{UI}$ and $\mathbf{IP}$ on the shared product latent factor $\mathbf{Y}$.

5.1.3 UCIR Problem with Price as both Distribution and Quantity

UCIR problem (recommending products in unexplored categories): *Given the user-product matrix* $\mathbf{UI}$ *which captures user product purchase history, the system recommends products from unexplored categories* $\mathbf{C}_u^{unexplored}$ *to the user* u.

For the UCIR problem, we introduce both user-price matrix $\mathbf{UP}$ (where price is a distribution) and product-price matrix $\mathbf{IP}$ (where price is a quantity). We interpret $\mathbf{UP}$ by two independent latent factors: user $\mathbf{X}$ and category price $\mathbf{M}$. Similarly, $\mathbf{IP}$ can be interpreted by two independent latent factors: product$\mathbf{Y}$ and product price $\mathbf{N}$. Note that in this interpretation, user latent factor $\mathbf{X}$ is shared by $\mathbf{UI}$ and $\mathbf{UP}$ while product latent factor $\mathbf{Y}$ is shared by $\mathbf{UI}$ and $\mathbf{IP}$. We form the UCIR problem in the matrix factorization style:

$$\begin{aligned}&\min_{\mathbf{X}^*,\mathbf{Y}^*,\mathbf{M}^*,\mathbf{N}^*} (1-\alpha-\beta)||\mathbf{CF}^{(UI)} \odot (\mathbf{UI} - \mathbf{X}^\top \mathbf{Y})||_F^2 + \\ &\alpha||\mathbf{CF}^{(UP)} \odot (\mathbf{UP} - \mathbf{X}^\top \mathbf{M})||_F^2 + \beta||\mathbf{CF}^{(IP)} \odot (\mathbf{IP} - \mathbf{Y}^\top \mathbf{N})||_F^2 \\ &+ \lambda((1-\beta)||\mathbf{X}||_F^2 + (1-\alpha)||\mathbf{Y}||_F^2 + \alpha||\mathbf{M}||_F^2 + \beta||\mathbf{N}||_F^2)\end{aligned} \tag{4}$$

where parameters α and β are used to adjust the influence of $\mathbf{UP}$ and $\mathbf{IP}$ on latent user factor $\mathbf{X}$ and latent product factor $\mathbf{Y}$ respectively.

5.2 Solution

We extend the alternating-least-squares (ALS) algorithm in work [7] to optimize the CR problem, the CIR problem and the UCIR problem. We start from solving the relatively simple CR problem, then the CIR problem and finally the most complex UCIR problem.

5.2.1 CR Problem Solution - Price as Distribution

In the CR problem, we need to solve three latent factors: user latent factor $\mathbf{X}$, category latent factor $\mathbf{C}$ and price latent factor $\mathbf{M}$. Observing that when $\mathbf{X}$ is fixed, the cost function becomes quadratic, so its global minimum can be readily computed and category latent factor $\mathbf{C}$ and price factor $\mathbf{M}$ are fixed. Similarly, when $\mathbf{C}$ and $\mathbf{M}$ are fixed, $\mathbf{X}$ can be readily computed.

$$\begin{aligned}\mathbf{X}_u = & ((1-\alpha)\mathbf{C}\,(\mathcal{D} \circ (\mathbf{CF}^{(UC)\top})_u)\,\mathbf{C}^\top + \\ & \alpha\mathbf{M}\,(\mathcal{D} \circ (\mathbf{CF}^{(UP)\top})_u)\,\mathbf{M}^\top + \lambda\mathbf{I})^{-1} \\ & ((1-\alpha)\mathbf{C}\,(\mathcal{D} \circ (\mathbf{CF}^{(UC)\top})_u)\,(\mathbf{UC}^\top)_u + \\ & \alpha\mathbf{M}\,(\mathcal{D} \circ (\mathbf{CF}^{(UP)\top})_u)\,(\mathbf{UP}^\top)_u)\end{aligned} \tag{5}$$

In the above equation, $\mathcal{D} \circ vec$ is the operation of forming a diagonal matrix with vector vec on its diagonal.

Alternatively, the expressions for category latent factor $\mathbf{C}_c$ and price latent factor $\mathbf{M}_j$ are:

$$\begin{aligned}\mathbf{C}_c &= (\mathbf{X}\,(\mathcal{D} \circ \mathbf{CF}_c^{(UC)})\,\mathbf{X}^\top + \lambda\mathbf{I})^{-1}\mathbf{X}\,\mathbf{CF}_c^{(UC)}\,\mathbf{UC}_c \\ \mathbf{M}_j &= (\mathbf{X}\,(\mathcal{D} \circ \mathbf{CF}_j^{(UP)})\,\mathbf{X}^\top + \lambda\mathbf{I})^{-1}\mathbf{X}\,\mathbf{CF}_j^{(UP)}\,\mathbf{UP}_j\end{aligned} \tag{6}$$

The objective function is optimized by applying Equation (5) and Equation (6) iteratively.

To generate a top-N category list for each user u, we assume that the user's candidate category set (categories that have no purchase records) as $\mathbf{C}_u^{unexplored}$, and for each candidate c in $\mathbf{C}_u^{unexplored}$, we calculate a prediction score $\mathbf{PD}_{uc}$ by equation (7). We then rank categories according to the prediction scores and recommend the top-N categoires to the user:

$$\mathbf{PD}_{uc} = (\mathbf{X}_u)^\top\,\mathbf{C}_c \tag{7}$$

where $\mathbf{X}_u$ and $\mathbf{C}_c$ are user latent factor and category latent factor respectively, which are calculated as described in equation (5) and (6).

5.2.2 CIR Problem Solution - Price as Quantity

In the CIR problem, we need to solve three latent factors: user latent factor $\mathbf{X}$, product latent factor $\mathbf{Y}$ and price latent factor $\mathbf{N}$. Similar to the CR problem, we optimize the objective function iteratively by alternating between (user latent factor $\mathbf{X}$, price latent factor $\mathbf{N}$) pair and product latent factor $\mathbf{Y}$.

$$\begin{aligned}\mathbf{Y}_i = & ((1-\alpha)\mathbf{X}\,(\mathcal{D} \circ \mathbf{CF}_i^{(UI)})\,\mathbf{X}^\top + \\ & \alpha\mathbf{N}\,(\mathcal{D} \circ (\mathbf{CF}^{(IP)\top})_i)\,\mathbf{N}^\top + \lambda\mathbf{I})^{-1} \\ & ((1-\alpha)\mathbf{X}\,(\mathcal{D} \circ \mathbf{CF}_i^{(UI)})\,\mathbf{UI}_i + \alpha\mathbf{N}\,(\mathcal{D} \circ (\mathbf{CF}^{(IP)\top})_i)\,(\mathbf{IP}^\top)_i) \\ \mathbf{X}_u = & (\mathbf{Y}\,(\mathcal{D} \circ (\mathbf{CF}^{(UI)\top})_u)\,\mathbf{Y}^\top + \lambda\mathbf{I})^{-1} \\ & \mathbf{Y}\,(\mathcal{D} \circ (\mathbf{CF}^{(UI)\top})_u)(\mathbf{UI}^\top)_u \\ \mathbf{N}_k = & (\mathbf{Y}\,(\mathcal{D} \circ \mathbf{CF}_k^{(IP)})\,\mathbf{Y}^\top + \lambda\mathbf{I})^{-1}\mathbf{Y}\,(\mathcal{D} \circ \mathbf{CF}_k^{(IP)})\mathbf{IP}_k\end{aligned}$$

To generate a top-N product list for each user u, we assume that the user's candidate product set (products under the given category) as ϕ_u, and for each candidate product i in ϕ_u, we calculate a prediction score $\mathbf{PD}_{ui}$ by equation (8). Then we rank products according to the prediction scores and recommend the top-N products to the user:

$$\mathbf{PD}_{ui} = (\mathbf{X}_u)^\top \mathbf{Y}_i \tag{8}$$

where $\mathbf{X}_u$ and $\mathbf{Y}_i$ are user latent factor and product latent factor respectively, which are calculated as described in above formulas.

5.2.3 UCIR Problem Solution - Price as both Distribution and Quantity

In the UCIR problem, the situation is a little bit more complex: we need to solve four latent factors. They are user latent factor $\mathbf{X}$, product latent factor $\mathbf{Y}$, user price latent factor $\mathbf{M}$ and product price latent factor $\mathbf{N}$. After careful arrangement, we can alternate between $\mathbf{X}$, $\mathbf{N}$ and $\mathbf{Y}$, $\mathbf{M}$. As shown in equation (9),(10), the updates of $\mathbf{X}$, $\mathbf{N}$ rely on the value of $\mathbf{Y}$, $\mathbf{M}$ and vice versa.

$$\begin{aligned}\mathbf{X}_u = {} & \frac{1}{1-\beta}((1-\alpha-\beta)\mathbf{Y}\,(\mathcal{D}\circ(\mathbf{CF}^{(UI)\top})_u)\,\mathbf{Y}^\top + \\ & \alpha\,\mathbf{M}\,(\mathcal{D}\circ(\mathbf{CF}^{(UP)\top})_u)\,\mathbf{M}^\top + \lambda\mathbf{I})^{-1} \\ & ((1-\alpha-\beta)\mathbf{Y}\,(\mathcal{D}\circ(\mathbf{CF}^{(UI)\top})_u)\,(\mathbf{UI}^\top)_u + \\ & \alpha\mathbf{M}\,(\mathcal{D}\circ(\mathbf{CF}^{(UP)\top})_u)\,(\mathbf{UP}^\top)_u) \\ \mathbf{N}_k = {} & (\mathbf{Y}\,(\mathcal{D}\circ\mathbf{CF}_k^{(IP)})\,\mathbf{Y}^\top + \lambda\mathbf{I})^{-1}\mathbf{Y}\,\mathbf{CF}_k^{(IP)}\,\mathbf{IP}_k\end{aligned} \tag{9}$$

$$\begin{aligned}\mathbf{Y}_i = {} & \frac{1}{1-\alpha}((1-\alpha-\beta)\mathbf{X}\,(\mathcal{D}\circ\mathbf{CF}_i^{(UI)})\,\mathbf{X}^\top + \\ & \beta\,\mathbf{N}(\mathcal{D}\circ(\mathbf{CF}^{(IP)\top})_i)\mathbf{N}^\top + \lambda\mathbf{I})^{-1} \\ & ((1-\alpha-\beta)\mathbf{X}\,(\mathcal{D}\circ\mathbf{CF}_i^{(UI)})\,\mathbf{UI}_i + \\ & \alpha\mathbf{N}\,(\mathcal{D}\circ(\mathbf{CF}^{(IP)\top})_i)\,(\mathbf{IP}^\top)_i) \\ \mathbf{M}_j = {} & (\mathbf{X}\,(\mathcal{D}\circ\mathbf{CF}_j^{(UP)})\,\mathbf{X}^\top + \lambda\mathbf{I})^{-1}\mathbf{X}\,\mathbf{CF}_j^{(UP)}\,\mathbf{UP}_j\end{aligned} \tag{10}$$

To generate a top-N product list for each user u, we assume that the user's candidate product set (products under all unexplored categories) is ϕ'_u, and for each candidate product i in ϕ'_u, we calculate a prediction score $\mathbf{PD}_{ui}$ by equation (8).

6. EXPERIMENT (ANSWER TO R4)

In this section, we present experiments on product recommendation in unexplored categories. Section 6.1 and 6.2 introduce the experiment setting and baselines. Section 6.3, 6.4 and 6.5 study the performance improvement by adding price as distribution, price as quantity, and price as both distribution and quantity, respectively in two shopping orientations.

6.1 Experiment Setting

To split data into train set and test set, we group (u, i) pairs by category c into $(u, group_c) = \{(u, i_1), ...(u, i_n)\}_{i_1,...,i_n \in c}$ since our task is product recommendation in unexplored categories. We sample 70% for training and 30% for test on $(u, group_c)$. For user-price matrix $\mathbf{UP}$, we get $2,514$ typical price patterns for all categories, so $\mathbf{UP}$'s perspective dimension (column number) is $2,514$. For product-price matrix $\mathbf{IP}$, we have 5 bins of price level for each category, so $\mathbf{IP}$'s perspective dimension (column number) is $44 \times 5 = 220$.

In our implementation, the lower and upper bounds of price interval for each category are 10% and 90% quantiles of the price distribution respectively. Including products of extremely low or high price into the price interval reduces the meaningful interval of price level, therefore making price level less discriminative. For products with price lower than 10% quantiles, their price level p is set to 0. Similarly, for products with price higher than 90% quantile, their price level p is set to 1.

6.2 Baselines

We compare our method to two baselines: collaborative filtering for implicit feedback (WRMF) [7] and collaborative filtering with regularization (REG) [9].

WRMF: WRMF is the state of the art in product recommendation. It does matrix factorization for implicit feedback but doesn't consider any side information in the process. Comparing our model to WRMF can learn how much price factor helps in product recommendation of unexplored categories.

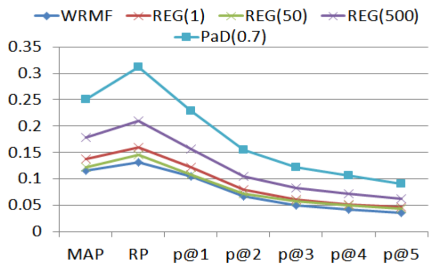
REG: We can also add price information as regularization. We use price to measure user and item similarity. To measure price preference similarity between users, we take the negative of Euclidean distance between two rows in $\mathbf{UP}$. Similarily, we calculate item similarity using $\mathbf{IP}$. Comparing our model to REG can show the superiority of our price encoding method. The regularization parameter λ_N is tuned to its best performance among $1, 50, 500$. We add user similarity in the CR problem and item similarity in the CIR problem. For the UCIR problem, we add both user and item similarity.

There are other models [1] which can embed side information. But it is not likely to add the price as distribution perspective into their bayesian style approach since the distribution does not belong to exponential family as shown in Section 3.2. So we don't consider them here.

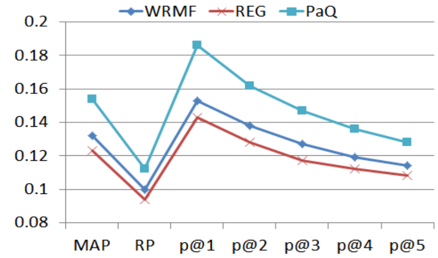
6.3 Category Recommendation with Price as Distribution

In this experiment, we recommend unexplored categories (CR problem). Both the training data and the test data are user-category pairs. Our method, price as distribution (PaD), achieves its best performance when $\alpha = 0.7$. We see that our method achieves better performance in Figure 6(a) on all metrics. On MAP metric, it improves WRMF and REG baselines by 54.78% and 45.90% relatively. The improvement is significant according to t-test with significance level $p = 0.005$ for both WRMF and REG. Compared to WRMF, it shows that price as distribution perspective does help in unexplored category situation. Compared to REG, it shows that the appropriate encoding of price as distribution perspective does play a significant role in the performance boost. Our method achieves 0.228 on p@1 metric, which means that for the top recommended category, users have more than $1/5$ chance to make purchase in it.

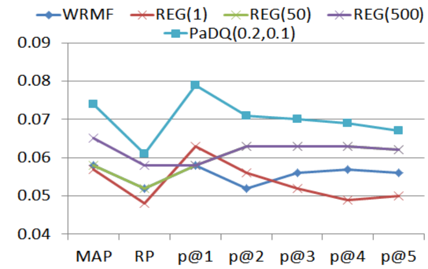
We further study the impact of price as distribution perspective ($\mathbf{UP}$ matrix) in our method on category recommendation. The impact is controlled by the paramter α in Equation (2), which ranges from 0 to 1. When α is 0, the contribution from matrix $\mathbf{UP}$ is zero. This degenerates to WRMF baseline, which uses no price information at all. When α is 1, the contribution from $\mathbf{UC}$ matrix is zero. That is, we only use $\mathbf{UP}$ matrix to train the user latent factor for prediction. As shown in Figure 6(d), our method consistently improves when α increases and reaches the best performance when α is 0.7, which means that the $\mathbf{UP}$ matrix



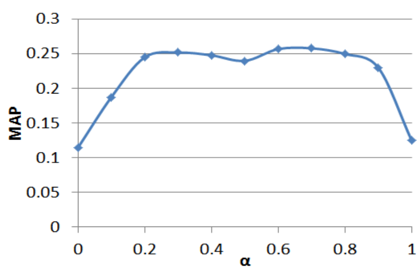
(a) Comparison of WRMF, REG and our PaD for CR problem



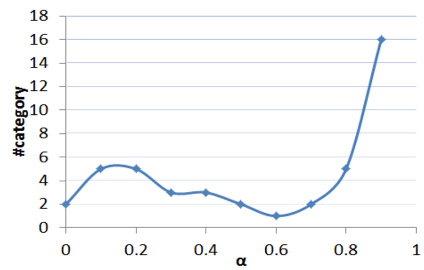
(b) Overall comparison of WRMF, REG and our PaQ for CIR problem



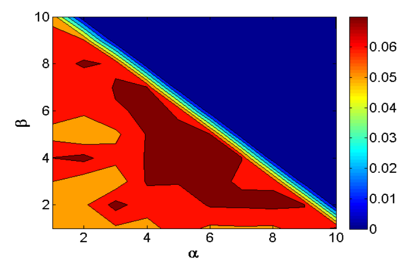
(c) Comparison of WRMF, REG and our PaDQ for UCIR problem



(d) Impact of α on CR problem



(e) Distribution of best tuned α in categories on CIR problem



(f) Impact of α and β on UCIR problem

Figure 6: Comparison of our methods((a)-(c)) to baselines and study on the impact of price ((d)-(f)) in three problems: The number in brackets is the parameter λ of REG, α of PaD, α of PaQ, α, β of PaDQ

plays an improtant role. The consistent improvement also shows that our method is insensitive to parameter α in interval $[0.2, 0.9]$.

From Figure 6(d), we make another comparison between $\alpha = 0$ and $\alpha = 1$. $\alpha = 0$ means that we only use the **UC** matrix while $\alpha = 1$ means that we only use the **UP** matrix. Suprisingly, merely using **UP** matrix outperforms merely using **UC** matrix by 8.9% relatively. We think the reason is that first of all, the **UP** matrix (price as distribution perspective) also contains the category information and is actually a refinement of **UC** matrix. Secondly, merely recording the explored categories of a user is too rough to capture user preference.

6.4 Product Recommendation under Given Category with Price as Quantity

In this experiment, we recommend products in the given unexplored category (CIR problem). Both the training data and the test data are user-product pairs. As shown in Figure 6(b), our method, price as quantity (PaQ), improves WRMF and REG baselines by 15.95% and 26.14% relatively. The improvement is significant according to t-test with significance level $p = 0.005$ for both WRMF and REG. The performance is separately tuned to the best on each category. This is feasible in real applications since the category is fixed in the CIR problem. In our PaQ method, the decomposition of **UI** matrix and **IP** matrix shares the same product latent factor. The improvement shows that price as a quantity perspective does help to achieve better product latent factor in decomposition. The overall result achieves 0.155 on MAP metric and 0.187 on $p@1$ metric, which means that for the top recommended novel product, users have nearly 1/6 chance to buy it.

We also sort categories by their improvement ratio as shown in Figure 7. In category “memory card/U disk”, PaQ achieves 11.13% relative performance improvement. In category “tourist ticket collection”, PaQ achieves 53.08% relative improvement. In category “MP3/MP4/iPod”, PaQ gets a neglectable relative improvement of 0.32%. Analysis shows that the number of items under this category is relatively small (897) and the baseline already reaches a good performance at 0.66 on MAP. There is only one category whose performance drops when price information is added. It is the category “mobile phone accessories” and the performance slightly drops by 2.5% relatively. Further analysis shows that most products in this category are very cheap and people consider factors such as design more than price in purchase. Based on the experimental results, it is verified that price as quantity perspective helps a lot in most categories. We also study the distribution of best tuned α on categories. As shown in Figure 6(e), for most categories α is tuned to 0.9, indicating a large impact from price as quantity perspective.

6.5 Product Recommendation with Price as both Distribution and Quantity

In this experiment, we recommend items in unexplored categories (UCIR problem). Both the training data and test data are user-product pairs. Compared to the CIR problem, this problem is more difficult since it has to determine both the category and the product for recommendation simultaneously. We compare our method, price as both distribution and quantity (PaDQ), to both WRMF and REG baselines. As shown in Figure 6(c), PaDQ achieves better performance than WRMF and REG by 27.59% and 13.85% relatively. The best performance is reached when α and β are set to 0.2 and 0.1 respectively, indicating that both perspectives of price contribute to the performance. The improvement is significant according to t-test with significance level $p = 0.001$ for WRMF and $p = 0.05$ for REG.

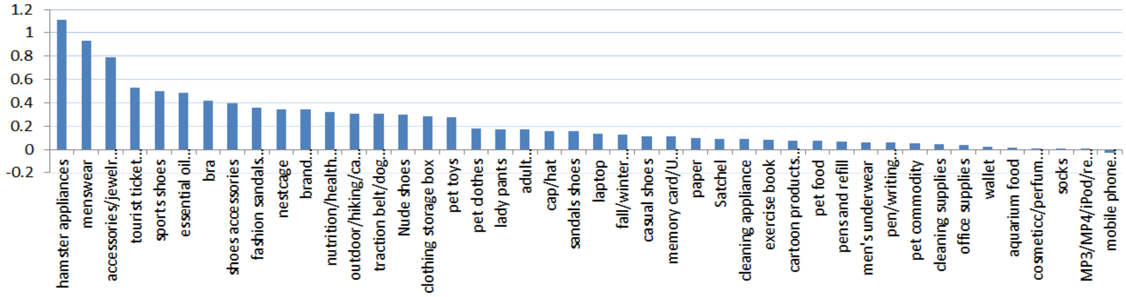


Figure 7: Improvement ratio of PaQ over WRMF is shown by category

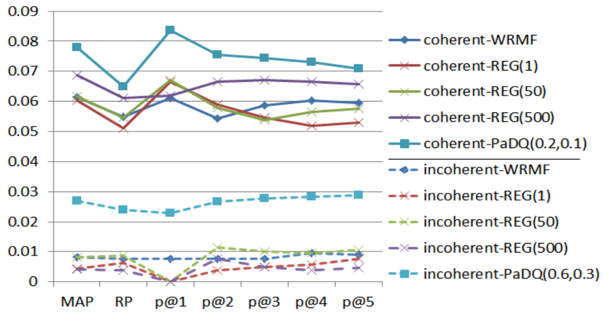


Figure 8: $UCIR$ problem: performance on coherent and incoherent test data

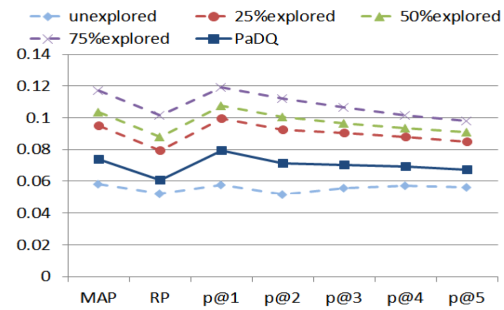


Figure 9: Price comes to help in unexplored category product recommendation

We analyze the impact of α and β on the performance. Figure 6(f) is a contour map of performance on parameter α and β in Equation (4). The elements on the top right side of diagonal is 0 because the sum of α and β shoud be smaller than 1. Performance consistently improves when both α and β increases. Our method is insensitive to these two parameters once they enter the region: $\alpha + \beta \leq 0.9, \alpha \in [0.2, 0.7], \beta \in [0.1, 0.6]$. But if we only increase a single parameter, either α or β, the performance boost is not so significant. This indicates that both "price as distribution" perspective and "price as quantity" perpective need to be considered in order to achieve best performance boost.

7. DISCUSSION

In this section, we discuss the connection of the R4's answer to R1, R2, and R3. As shown in Figure 3(b) in Section 3, the price preference of users varies among different categories. Therefore, we conduct some detailed analysis of prediction performance with respect to the distance of price preference from the unexplored category to explored categories. If the distance is smaller than 0.3, we categorize such test data as coherent type. And other test data are categorized as incoherent type.

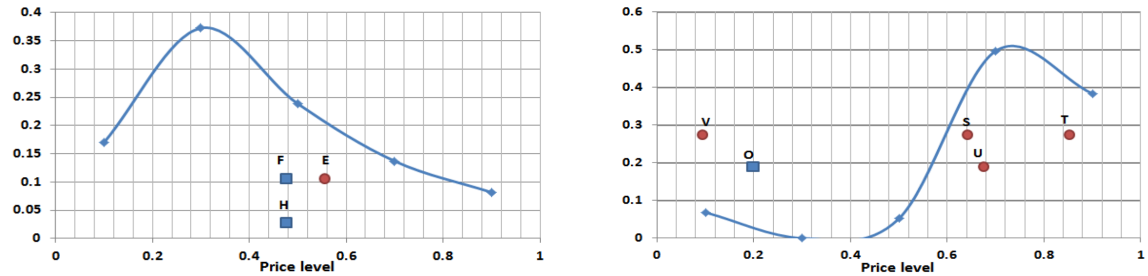
We compare our method (PaDQ) to WRMF and REG on these two types of test separately. As shown in Figure 8, our algorithm improves on both types of test consistently. The performance on coherent type of test is higher than that on incoherent type, because incoherent type of test is more difficult to predict. But the absolute performance boost on the incoherent type of test is greater. The best parameter settings are different for the two types. PaDQ is tuned to $\alpha = 0.2$ and $\beta = 0.1$ on coherent type of test while it is tuned to $\alpha = 0.6$ and $\beta = 0.3$ on incoherent type of test. Larger value for α and β indicates that price does provide greater help on the incoherent (more difficult) type of test.

We also conduct case study on these two types. Figure 10 shows the recommendation performance for two typical users, 111 and 20363. These two users are previously studied in Figure 3 in Section 3. The graph area shows the groundtruth price distribution of products that the user purchased in this category. The table shows the MAP value of three methods and the top 5 predicted items.

In the coherent type of test case study, we observe that although all the three methods have at least 1 good prediction (right item or right category) in top 5, PaDQ achieves more good predictions than the other two and one of the good prediction is right at top1. In the graph area, we observe that item F and item H are not only in the right category, but also in the appropriate price interval based on user 111's price preference. In the incoherent type of test case study, we observe that all the top 5 predicions from WRMF are totally wrong (neither right category nor right item). REG method gets one prediction in right category but none in right item. Futhermore, item O is not in the appropriate price interval based on user 20363's price preference. PaDQ gets 4 out of 5 predictions right. Item S, T, U and V are all in the appropriate price interval based on user 20363's price preference. The above case studies present concrete examples of the performance boost: items predicted by PaDQ are more likely to be in the right category and at the right price level.

Figure 2(a) in section 3 has shown the performance gap between recommendation in explored categories and that in unexplored categories. How well does the proposed PaDQ method shorten this gap? We can see from Figure 9 that it shortens the performance gap between *unexplored* condition and 25%*explored* condition by relatively 43% on MAP and 51% on p@1. Price does come to help!

The unexplored category problem has novelty as its natural property but is not necessarily a cold-start problem [14][11][18]. To be specific, an unexplored category for user A can be an explored category for user B. On a mature website, this is the general case. It becomes a cold-start problem only when the target category has not been explored by any user. This appears only when a brand new category is added to the site.



	MAP	1	2	3	4	5
WRMF	0.04	A: □	B: □	C: □	D:□	E: ●
REG	0.04	A: □	B: □	D: □	C: □	E: ●
PaDQ	0.46	E: ●	A: □	F: ■	G: □	H: ■

	MAP	1	2	3	4	5
WRMF	0.00	I: □	J: □	K: □	L: □	M: □
REG	0.00	N: □	O: ■	P: □	M: □	Q: □
PaDQ	0.54	R: □	S: ●	T: ●	U: ●	V: ●

(a) coherent type: user 111 in the test category *nestcage* (b) incoherent type: user 20363 in the test category *wallet*

Figure 10: Case Study of performance on coherent and incoherent test data. The graph area shows the groundtruth price distribution and the table shows top 5 items predicted by three methods. □ means wrong item in wrong category, ■ means wrong item in the right category, • means right item in the right category. The alphabet before colon is the item ID. We align both ■ and • items to the corresponding price level in the graph area. □ items are not shown since they are not in the ground truth category.

8. CONCLUSION AND FUTURE WORK

In this paper, we investigate the challenge problem of product recommendation in unexplored categories. For a user, an unexplored category is defined as a category that the user has no purchase history. Developing a good recommendation system in unexlored categories is beneficial to both consumers and retailers and can produce great business value. We conduct a series of progressive experiments and analysis on a dataset collected from a leading E-commerce website. We find out that unexplored category causes significant performance drop for state-of-the-art recommendation systems. Price can be a factor tranferrable across categories, and therefore can be helpful for recommendation in unexplored categories. We consider two perspectives of the price factor: price as a quantity for a product and price as a distribution for a user. Based on the observation that user behavior with respect to price factor across categories is subtle and needs to be carefully modeled, we propose a new method encoding both perspectives of price factor for both shopping orientations. Our experimental results show that encoding price as distribution helps to improve unexplored category recommendation precision and encoding price as quantity helps to improve product recommendation precision in a given unexplored category. Our proposed method PaDQ which encodes the two perspectives of price factor significantly boost the recommendation performance in unexplored categories. It shortens the performance gap by 43% relatively. In the future, we plan to investigate more factors that can be used to transfer user preference across categories.

9. REFERENCES

[1] A. Ahmed, B. Kanagal, S. Pandey, V. Josifovski, L. G. Pueyo, and J. Yuan. Latent factor models with additive and hierarchically-smoothed user preferences. In *WSDM*, pages 385–394, 2013.

[2] M. Brown, N. Pope, and K. Voges. Buying or browsing?: An exploration of shopping orientations and online purchase intention. *European Journal of Marketing*, 37(11/12):1666 – 1684, 2003.

[3] H. Chen. The impact of comments and recommendation system on online shopper buying behaviour. *JNW*, 7(2):345–350, 2012.

[4] T. Chen, H. Li, Q. Yang, and Y. Yu. General functional matrix factorization using gradient boosting. In *ICML*, pages 436–444, 2013.

[5] D. M. Fleder and K. Hosanagar. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management Science*, 55(5):697–712, 2009.

[6] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[7] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, 2008.

[8] B. J. Jansen. Chris anderson, the long tail: Why the future of business is selling less or more, hyperion, new york (2006) isbn 1-4013-0237-8 $24.95. *Inf. Process. Manage.*, 43(4):1147–1148, 2007.

[9] 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*, WSDM '11, pages 287–296, 2011.

[10] K. Onuma, H. Tong, and C. Faloutsos. Tangent: a novel, 'surprise me', recommendation algorithm. In *KDD*, pages 657–666, 2009.

[11] S.-T. Park and W. Chu. Pairwise preference regression for cold-start recommendation. In *RecSys*, pages 21–28, 2009.

[12] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proc. KDD Cup Workshop at SIGKDD'07, 13th ACM Int. Conf. on Knowledge Discovery and Data Mining*, pages 39–42, 2007.

[13] I. Porteous, A. U. Asuncion, and M. Welling. Bayesian matrix factorization with side information and dirichlet process mixtures. In *AAAI*, 2010.

[14] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[15] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *KDD*, pages 650–658, 2008.

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

[17] J. Wang and Y. Zhang. Opportunity model for e-commerce recommendation: right product; right time. In *SIGIR*, pages 303–312, 2013.

[18] Q. Yuan, L. Chen, and S. Zhao. Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation. In *RecSys*, pages 245–252, 2011.

[19] G. Zhao, M.-L. Lee, W. Hsu, and W. Chen. Increasing temporal diversity with purchase intervals. In *SIGIR*, pages 165–174, 2012.

[20] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW*, pages 22–32, 2005.