

Boosting Recommendation in Unexplored Categories by User Price Preference

JIA CHEN, Shanghai Jiaotong University
 QIN JIN, Renmin University of China
 SHIWAN ZHAO, IBM CRL
 SHENGHUA BAO, IBM Waston Group
 LI ZHANG and ZHONG SU, IBM CRL
 YONG YU, Shanghai Jiaotong University

State-of-the-art methods for product recommendation encounter a 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 article, we investigate the challenging problem of product recommendation in unexplored categories and discover that the price, a factor comparable across categories, can improve the recommendation performance significantly. We introduce the price utility concept to characterize users' sense of price and propose three different utility functions. We show that user price preference in a category is a distribution and we mine typical user price preference patterns based on three different types of distance between distributions. We fuse user price preference through regularization and joint factorization to boost recommendation performance in both browsing and buying shopping orientations. Experimental results show that fusing user price preference improves performance in a series of recommendation tasks: unexplored category recommendation, product recommendation under a given unexplored category, and product recommendation under generic unexplored categories.

CCS Concepts: • **Information systems** → **Collaborative filtering**;

Additional Key Words and Phrases: Product recommendation, unexplored category, price

ACM Reference Format:

Jia Chen, Qin Jin, Shiwang Zhao, Shenghua Bao, Li Zhang, Zhong Su, and Yong Yu. 2016. Boosting recommendation in unexplored categories by user price preference. *ACM Trans. Inf. Syst.* 35, 2, Article 12 (October 2016), 27 pages.

DOI: <http://dx.doi.org/10.1145/2978579>

1. INTRODUCTION

At present, a mature E-commerce website has abundant product categories that offer rich shopping choices for users. However, users are normally used to browsing and purchasing in only a small portion of categories. Current recommendation systems

This work was partially supported by the Beijing Natural Science Foundation (No. 4142029), the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China (No. 14XNLQ01).

Authors' addresses: J. Chen and Y. Yu, Computer Science Department, 800 Dongchuan Road, Minhang District, Shanghai, China, 200240; emails: {chenjia, yuy}@apex.sjtu.edu.cn; Q. Jin (corresponding author), School of Information, Renmin University of China, No. 59 Zhongguancun Street, Haidian District, Beijing, P.R. China, 100872; email: qjin@ruc.edu.cn; S. Zhao, L. Zhang, and Z. Su, IBM CRL, Building 19, Zhongguancun Software Park, 8 Dongbeiwang West Road, Haidian District, Beijing, P.R. China, 100193; emails: {zhaosw, zhangli, suzhong}@cn.ibm.com; S. Bao, IBM Waston Group, 650 Harry Road, San Jose, CA, 95120; email: baoshhua@us.ibm.com.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2016 ACM 1046-8188/2016/10-ART12 \$15.00

DOI: <http://dx.doi.org/10.1145/2978579>

tend to recommend more correct products in categories with user purchase history rather than in categories without. We discover that the recommendation performance drops in unexplored categories for the state-of-the-art algorithm [Chen et al. 2014]. This makes products in categories without the user's purchase history less exposed to the user. Some users may not be aware of these categories.

Behind this phenomenon, 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 of users made purchases in fewer than four categories while the number of categories on this website is 761. Recommending products in unexplored categories not only can meet users' potential shopping 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 the setting of “unexplored categories” 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 it is hard to compare the user preference on products across different categories. After investigating several factors, we discover that price is a factor that is comparable 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.

We make the following contributions by using the price factor to boost recommendation performance.

- (1) We propose three different ways to transform the concrete price value information to “price utility,” an abstract description of a user's sense of the product price.
- (2) We describe user price preference under a category by a distribution over price utility. We propose different distance types to measure the difference between user price preference in categories with both sufficient and insufficient shopping records.
- (3) We design two matrix representations of user price preference and fuse them by regularization and factorization to boost the recommendation performance.

We evaluate our method in three unexplored-category-related recommendation tasks from two shopping orientations: browsing and buying [Brown et al. 2003]. In the browsing orientation, users prefer to gather more information. Therefore, 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 to users for making a quick purchase decision.

In both shopping orientations, we compare our method systematically to six state-of-the-art baselines, which could be categorized to three families: popular item recommendation family (POPULAR [Grbovic et al. 2015] and POPULAR-CR), user behavior data only family (weighted matrix factorization (WRMF) [Hu et al. 2008] and Bayesian personalized recommendation (BPR) [Rendle et al. 2009]), and price preference modeling family (UPPBoost.SIMPLE and UPPBoost.SOFT [Fang and Si 2011]). The experimental results show that our proposed method does improve the recommendation performance significantly on all three unexplored-category-related recommendation tasks by fusing user price preference.

The rest of the article is organized as follows. Section 2 presents related work. Section 3 introduces the dataset and reports the observation of recommendation performance drop under unexplored categories in this dataset. Section 4 analyzes the subtleness of user price preference across categories and its correlation to user behavior data. Section 5 proposes several strategies to characterize price utility, an abstract description of a user's sense of the product price, and represents user price preference in a matrix form based on typical price preference patterns. Section 6 presents a formal definition of boosting product recommendation by user price preference in a series of tasks and proposes to fuse the user price preference through two different methods. Section 7 reports extensive experimental results. Section 8 investigates cases and discusses the merits of regularization versus those of joint factorization in utilizing price preference. Section 9 draws conclusions.

2. RELATED WORK AND TECHNICAL BACKGROUND

2.1. 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 [Fleder and Hosanagar 2009; Ziegler et al. 2005; Jansen 2007; Onuma et al. 2009] and pursues the balance between novelty and accuracy, that is, improving the novelty while maintaining the accuracy as high 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 main research challenge in our problem is to reduce the performance drop in unexplored categories. Therefore, we only focus on the accuracy-related metrics in this work.

Many factors affect user purchase behavior and can be leveraged to improve recommendation performance. Time factor is considered in Zhao et al. [2012] and Wang and Zhang [2013]. Chen [2012] 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 tasks. To the best of our knowledge, the most related one is the exploration of price's marginal net utility role in E-commerce recommendation in Wang and Zhang [2011]. Their approach focuses on the balance between re-purchase product recommendation and un-purchased product recommendation. It does not address the unexplored category problem. Differently, we directly tackle the unexplored category problem in E-commerce recommendation and model user price preference as distribution. We fuse it through regularization and factorization to boost performance in three different recommendation tasks under an unexplored-category setting.

There are abundant algorithms on leveraging side information to boost the recommendation performance [Porteous et al. 2010; Chen et al. 2013; Ahmed et al. 2013; Paterek 2007]. Porteous et al. [2010] proposes a Bayesian matrix factorization model with embedded side information. Chen et al. [2013] constructs feature functions in matrix factorization process. Paterek [2007] introduces regularization to embed additional information. The above-mentioned algorithms focus on the explicit feedback [Paterek 2007; Koren 2008], while E-commerce-related applications usually have to be centered on implicit feedback [Mnih 2012; Yang et al. 2012].

There are also several implicit feedback models that handle side information. Singh et al. propose joint matrix factorization [Singh and Gordon 2008] to add additional information. In this model, the user item matrix shares user latent factor or item latent factor with the side information matrices. In Ma et al. [2011], Jamali and Ester [2010], and Park and Chu [2009], the authors propose to add information through regularization. It requires that the user latent factor of friends in the social network

is similar. Yuan et al. [2011] compares regularization and joint factorization in fusing social network relation. In our work, we systematically investigate and compare these two approaches of fusing price preference in a series of recommendation tasks in unexplored categories. Furthermore, our conclusion about the property of these two approaches generalizes to other types of side information.

Item recommendation in unexplored categories has received increasing attention from the research community in recent years [Ahmed et al. 2013; McAuley et al. 2015; Grbovic et al. 2015]. The work most related to ours is Ahmed et al. [2013], in which they propose a hybrid model that adds and smooths user preference across categories. They adopt a two-step approach. First, they smooth user preference on attributes over the category-taxonomy to combat data sparsity. Then they embed the smoothed user price preference in collaborative filtering. That is, they “guess” a user’s preference in unexplored categories without leveraging the power from collaborative filtering. Differently, in our approach, we represent user price preference at category level for explored categories, which is not sparse compared to item level representation. Then we leverage collaborative filtering to borrow information from similar users to help recommendation in unexplored categories. Our approach unifies price preference smoothing and recommendation of unexplored categories in one step. McAuley et al. [2015] focuses on non-personalized item link prediction in general categories. Grbovic et al. [2015] exploits various side information, including user gender and item text description, to improve large-scale item recommendation covering general categories. Both works are related to item recommendation in general categories, including unexplored categories for certain users. In contrast, we are more focused on unexplored categories and explore the role of price information in this setting.

2.2. Implicit Feedback Problem

As pointed out in Hu et al. [2008], the purchase history of a user is implicit feedback rather than explicit feedback, so our problem formalization is built on the implicit feedback formalization. When handling implicit data, a confidence level C_{ui} is associated with each user-item by $C_{ui}^* = 1 + \theta r_{ui}$, where r is a variable associated with shopping records such as the amount of products. Elements in preference matrix P is derived by binarizing r_{ui} ,

$$P_{ui} = \begin{cases} 1 & r_{ui} > 0 \\ 0 & r_{ui} = 0 \end{cases}. \quad (1)$$

Combining the confidence level with the classical matrix factorization, we get the following problem formalization:

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

where $\|\cdot\|_F$ denotes the Frobenius norm, \odot is the elementwise product operation, \mathbf{X} is the user latent factor matrix, and \mathbf{Y} is the item 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 item 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: the user latent factor and the item latent factor.

The solution of Equation (2) can be calculated by use of the alternating least-square (ALS) algorithm. That is, we switch between estimating \mathbf{X} using \mathbf{Y} and estimating \mathbf{Y} using \mathbf{X} . In each, we have an analytical estimation by derivation:

$$\begin{aligned} \mathbf{X}_u &= (\mathbf{Y} \text{diag}(\mathbf{C}_u^\top) \mathbf{Y}^\top + \lambda \mathbf{I})^{-1} (\mathbf{Y}^\top \text{diag}(\mathbf{C}_u^\top) \mathbf{P}_u^\top) \\ \mathbf{Y}_i &= (\mathbf{X} \text{diag}(\mathbf{C}_i) \mathbf{X}^\top)^{-1} (\mathbf{X} \text{diag}(\mathbf{C}_i) \mathbf{P}_i), \end{aligned} \quad (3)$$

Table I. Example Categories

wedding dress	pants	winter coat
investment book	child book	history book
vacuum cleaner	electric cooker	hifi audio
cabinet	sofa	lighting
cleanser	wine	canned food

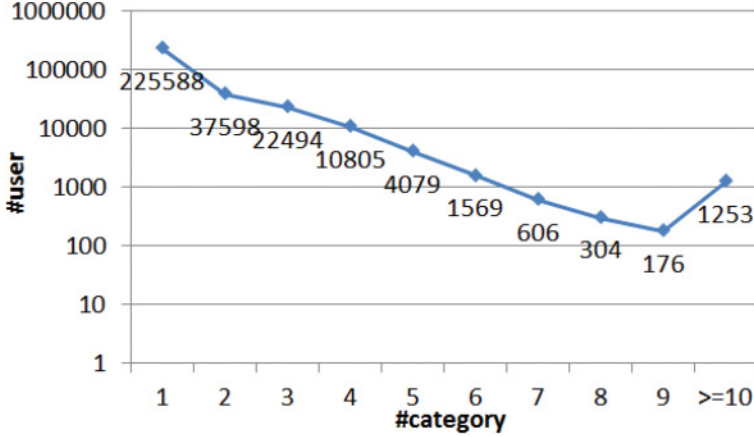


Fig. 1. Distribution of the number of users' explored categories.

where $diag(\mathbf{v})$ means constructing a diagonal matrix with vector \mathbf{v} on its diagonal. To affect the final user latent factor, we will add additional user price preference related term in the objective function of Equation (2). We will explain the corresponding change in the solution.

3. DATASET COLLECTING AND UNEXPLORED CATEGORY SETTING

3.1. Dataset Collecting

To investigate the problem of recommendation in unexplored categories, it requires a dataset to contain E-commerce transaction records. We have not noticed any public transaction dataset that contains price and category information. Therefore, we collected a dataset from one of the leading E-commerce websites in China to fulfill such a requirement. There are in total 761 product categories on this E-commerce website. Some example leaf categories are listed in Table I.

Figure 1 plots the distribution of the number of users' explored categories. We can see that most users only shop under one category, and when 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.

We remove 52 sub-categories of *prepaid game card* and *mobile recharge card* as the item price in these categories are proportional to the card usage time and do not reflect purchasing power. For example, a mobile recharge card at \$10 covers 500 minutes call while a mobile recharge card at \$20 covers 1,000 minutes call. The price per unit time is same for both cards though their prices differ. We draw a subsample from the dataset such that every user in the subsample has purchased at least 15 items and each item has at least five users. In the end, we get the subsample containing 11,893

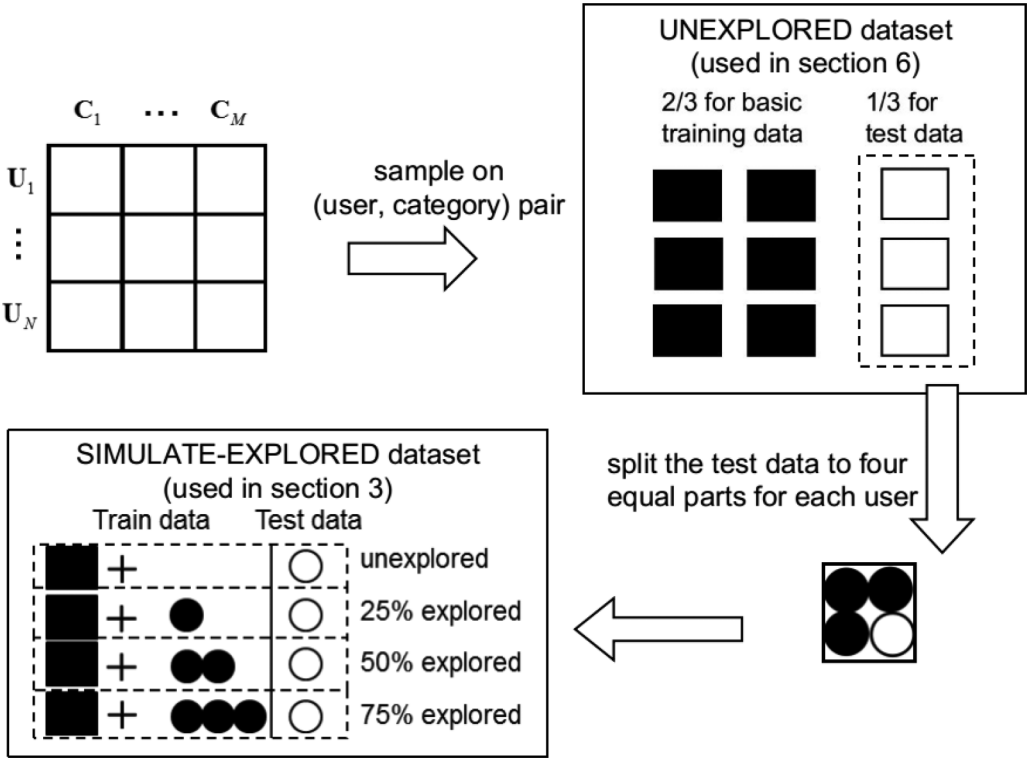


Fig. 2. Generating the UNEXPLORED and SIMULATE-EXPLORED datasets.

users, 22,227 items, and 361,584 shopping records under 344 categories. We follow two steps to create the unexplored category setting:

- (1) We group user-item pairs to user-category groups.
- (2) We split the dataset on user-category groups rather than on user-item pairs.

It would be ideal to split the dataset by time, but in the simulated unexplored category setting, we do not have enough data to split along time on the user-category group. We choose to randomly split the user-category groups to three equal parts for threefold cross validation, which is a common data-split strategy in a cold-start recommendation [Gantner et al. 2010]. Furthermore, the time span of the dataset covers only 3 months, a relatively short period, which makes random split reasonable. The first row in Figure 2 illustrates the procedure to generate the UNEXPLORED dataset, which is used to evaluate our method in Section 7. In each fold, the training data and test data of each user have no intersection on category to guarantee that it is an unexplored-category setting.

To compare performance between unexplored situation and explored situation, we further create a SIMULATE-EXPLORED dataset, which is only used in this section.

- (1) From the UNEXPLORED dataset, we split the test data into four equal parts *for each user*.
- (2) We randomly select one of the four parts as simulated test data, leaving the other three parts as non-test simulation data.

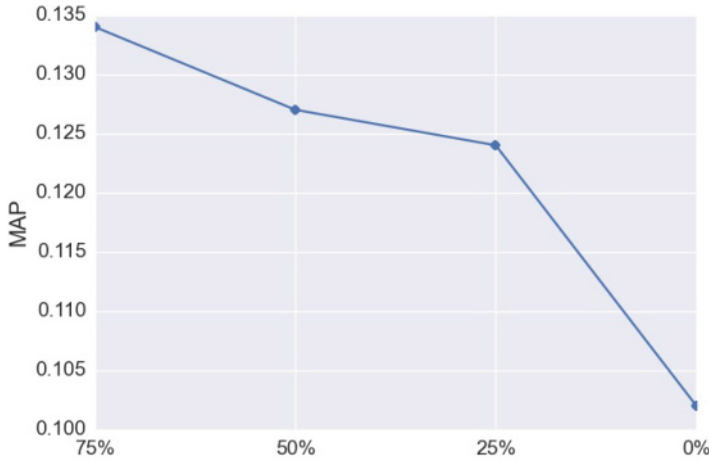


Fig. 3. Recommendation performance comparison under the unexplored- and explored-category conditions.

As illustrated in the second row of Figure 2, we conduct experiments under four training conditions with different training data setup: (1) 0% explored (totally 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; and (4) 75% explored, the basic training data plus all three parts from the non-test simulation data. In the $X\%$ explored setting, for each user-category pair (u, c) in the test data, the training data consists of $X\%$ shopping records of user u in category c . Larger X means that the user has explored more in the test category.

3.2. Performance Drop in Unexplored Category

We check how the state-of-the-art recommendation system [Hu et al. 2008] performs on the SIMULATE-EXPLORED dataset. In this dataset, the four settings share the same test set, but the percentage of the shopping records in the test category increases in the training set from 0% to 75%. The latent factor number is set to 40. We illustrate the performance under each setting in Figure 3. We see that the performance drops with less training data, which is not surprising. However, the performance drop between 0% (totally unexplored) setting and 25% (partially explored) setting is much larger than other neighboring settings by an order of magnitude, which indicates that traditional state-of-the-art system suffers much more significant performance drop under unexplored categories.

3.3. Price Preprocess

When we calculate price interval for each category, we use the 10% quantile as a lower bound l_c and the 90% quantile as the upper bound u_c to remove noises from extreme prices. For example, a laptop at \$100 is for a price-oriented search engine optimization (SEO)¹ or a piece of furniture at \$10,000 is just a high-priced example. Furthermore, we truncate every item's price p to $\min(\max(p, l_c), r_c)$ in each category.

¹The extremely cheap price is compensated by the extremely expensive transportation fee.

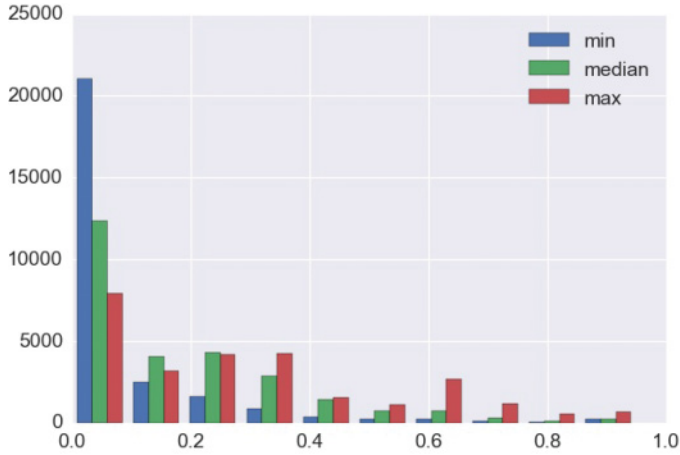


Fig. 4. Coherence of price factor across categories for users: x-axis represents the distance value and y-axis represents the number of users.

4. ANALYSIS OF USER PRICE PREFERENCE

4.1. Coherence of User Price Preference Across Category

We study whether a user will prefer low price products in an unexplored category if he/she prefers low-price products in the explored categories. We call this a 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.

To make user price preference comparable across categories, we normlize price within each category to $[0, 1]$. We will explore different price utility functions reflecting user's sense of a single item price in Section 5.1. For each user, we calculate the MEAN+CHI distance of price preference between all pairs of explored categories. The details of MEAN+CHI distance will be covered in Section 5.3. It is a distance that not only reflects the difference between the mean value of two distributions but also the difference between the shape of two distributions. For each user, we record the minimum, medium, and maximum of all the pairwise MEAN+CHI distances. We plot the histogram of these three values on all users in Figure 4.

Majority users have very small value on minimum distance (70% users have the minimum distance less than 0.1). This shows that, for majority users, there are at least two categories exhibiting very coherent price preference pattern. More than 20% users have maximum distance greater than 0.5. This shows that for these users, there exists two categories exhibiting very different price preference pattern. On medium distance, the number of users gradually decreases as the distance increases. This shows that for different users, the difference of price preference pattern between categories varies on users. Putting all these three pieces together, it shows that user price preference across categories is subtle and it needs to be carefully modeled.

4.2. Correlation between User Price Preference and User Shopping Behavior

We first do a thought experiment about the correlation between user price preference and user shopping behavior. Then we show that different price transform strategies and different distances for price preference affect the correlation. See Figure 5, for an example; we consider target user u and sort other users based on the similarity of their



Fig. 5. Distinguish users of different preference.

shopping behaviors to target user u 's. Based on the similarity value, we categorize users into three types:

1. The users on the leftmost side are users with very similar shopping behavior to target user u . For these users, collaborative filtering with shopping record data alone is enough to achieve decent recommendation performance for target user u .
2. The users in the middle part of the axis have subtle contribution in collaborative filtering. They have similar overlap on shopping records with the target user but their real preference may vary a lot, for example, some users who actually share a similar preference to target user u but their shopping records do not overlap a lot due to the large amount of products. Other users that actually do not share similar preference to target user u but their shopping records overlap to some extent on some popular products. To distinguish between these two groups of users, price provides additional information that is likely to help. Thus, if the correlation between user price preference and user shopping behavior is medium, then it suggests that price provides additional information that may be useful to distinguish these two groups of users.
3. The users on the rightmost side are users with no shopping behavior intersection with target user u . These users may have some overlap with the target user on price preference and could be leveraged in collaborative filtering to some extent.

To estimate the correlation between user price preference and user shopping behavior, we randomly sample 1,000 users and calculate pairwise similarity. On each user pair, we calculate shopping behavior similarity as the cosine value between rows in user-item matrix. On each user pair, we also calculate user price preference similarity as the cosine value between the rows in user pattern matrix, which is introduced in Section 5.4. To be specific, we introduce three price utility strategies and three distance types, resulting in nine combinations. We remove pairs whose similarity values are 0, as such pairs dominate the dataset and make the correlation calculation biased. Then we calculate the Pearson correlation between the two similarity values on pairs of the sampled users. On user pairs, the Pearson correlation between user behavior and price preference ranges from 0.599 to 0.677, as shown in Table II. It suggests two points:

Table II. Correlation between user Price Preference and user Shopping Behavior for Different Price Utility Strategies and Distance Types

	LT	WCN	WCN+LT
MEAN	0.623	0.677	0.623
CHI	0.599	0.660	0.640
MEAN+CHI	0.603	0.621	0.606

1. User price preference is correlated to user shopping behavior to some extent.
2. Different price utility strategies and distance types do affect its correlation with user shopping behavior. We will explore and compare these strategies and distance types in this article.

5. USER PRICE PREFERENCE MODELING

In this section, we characterize user price preference by four steps:

- (1) We measure the user's sense of a single item/product price by different utility functions. After this step, we get price utility for each product. It handles the gap between concrete price value and user's sense of price.
- (2) We group price utility values of products in his/her shopping history by category. We use the distribution of such values to model user preference on price under each category. After this step, we get a histogram on price utility under each category for a user. It describes user price preference at category-level, a higher level than product level.
- (3) We apply clustering to find out typical price preference patterns for users. In the clustering process, we design three types of distance to measure the difference between user preference. Based on typical price preference patterns, we encode a user's price preference in a category by one integer id using vector quantization. The output of this step is an integer id for each user's price preference under a category. It is the most compact representation we can get.
- (4) Based on the one integer encoding, we encode all user's price preference in two matrices: user-pattern matrix and user-user matrix. In user-pattern matrix, each row corresponds to a price preference profile of a user while user-user matrix encodes user similarity on price preference.

5.1. Price Utility

We use *price utility* l to represent the user's sense of the price of a product, which is the base of price preference. We note that price utility is not simply equal to the absolute price value p due to certain predictable irrationality [Ariely 2008]. For example, we are more concerned with a \$10 difference when we are considering a product at the price of \$50 than a product at the price of \$1,000 (First observation). However, when we are comparing products from different categories, we think that a laptop at the price of \$200 is a bargain and a piece of dessert at \$200 is usually a luxury, though they are sold at the same price (Second observation). The first observation indicates that our sensitivity to price is proportional to the price value. The second observation suggests that the price utility of a product is measured by its position within the category.

Given these two observations, we propose the following three strategies to transform absolute price value to price utility—the user's sense of price in the user's mind.

Log transform (LT): This strategy addresses the first observation. It applies log transform to the price value. After the log transform, the user's sensitivity to price is stretched to constant,

$$l = \log(p). \quad (4)$$

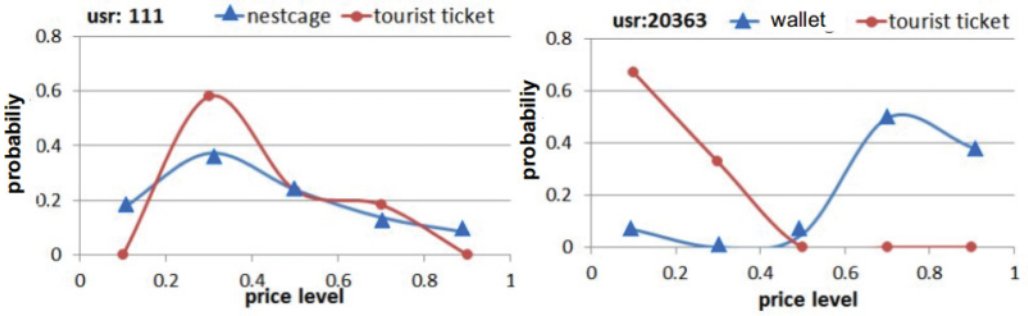


Fig. 6. User's price utility distribution under different categories based on shopping records: axis x is price utility and axis y is the probability at each utility.

Within category normalization (WCN): This strategy addresses the second observation. It normalizes the price within each category's price interval $[p_l, p_u]$. After normalization with regard to their position in the category, products from different categories can be directly compared,

$$l = \frac{p - p_l}{p_u - p_l}. \quad (5)$$

Within category normalization by log transform (WCN+LT): This strategy addresses both observations. It normalizes the price within each category's price interval and then applies log transform to the normalized value,

$$l = \frac{\log(p) - \log(p_l)}{\log(p_u) - \log(p_l)}. \quad (6)$$

5.2. Histogram of User Price Preference in Category

After transforming the absolute price value to price utility, we can get a distribution of the price utility for a user's explored category given the shopping records. This distribution characterizes the user's price preference under a category. We could then form the user's purchase behavior as sampling from the underlying price preference distribution. As shown in Figure 6, each plot shows a specific user's price utility histogram under certain categories. It provides much richer information than single purchase's price utility:

- (1) This shows the most probable price utility value that a user will purchase in a category. Consider user 111 and category nestcase; he has shopping records on both price utility value 0.3 and 0.7. From distribution, we can say that user 111 is most likely to buy a nestcase at price utility value 0.3 though he may also buy a nestcase at price utility value 0.7.
- (2) This shows the change of the most probable price utility value across categories for a user. Consider user 111 again; he has shopping records at price utility 0.3 in both category nestcase and tourist ticket. We cannot tell the difference between user price preference on these two categories from a single purchase. However, from distribution, we can tell that user 111 is more likely to buy a tourist ticket at price utility value 0.3 rather than a nestcase.
- (3) This shows the change of the most probable price utility value across users under the same category. Both user 111 and user 20363 have shopping records at price utility value 0.3 in tourist ticket. From distribution, we can say that user 111 is more likely to make a purchase at price utility value 0.3.

We can see that the distribution varies in skewness, kurtosis, and number of peaks from user to user and from category to category. Thus, we use a data-driven approach to extract typical price preference patterns without introducing any assumption about the distribution. To be specific, we cluster the distributions to find out typical price preference patterns.

We generate a histogram for the distribution by binning the price utility. A larger number of bins reduces information loss while a smaller number of bins enhances statistical strength under each bin. Thus, we need to tune the bin number to achieve a balance between information loss and statistical strength.

5.3. Typical Price Preference Pattern

We design three distance measurements to compute the difference between histograms.

mean distance (MEAN). We use the distance between mean value of histograms.

Mean value is a single point estimation of the histogram. This distance is a good approximation when there are not sufficient shopping records under an explored category for a user, that is, we cannot get a reliable histogram.

χ^2 distance (CHI). We use χ^2 distance [Pele and Werman 2010] to measure the weighted difference on each bin between two histograms,

$$\chi^2 = \sqrt{\frac{1}{2} \sum_{b \in \text{bins}} \frac{(x_b - y_b)^2}{x_b + y_b}}. \quad (7)$$

It is accurate on all details of the distribution when there are sufficient shopping records under an explored category for a user, that is, we have a reliable histogram approximation of the distribution. We do not use KL-divergence because its numerical value is ill conditioned when one of the histograms has value 0 on a particular bin. Such a situation is very common in user price distribution data. Even after smoothing, the KL-divergence value is still much less stable compared to the χ^2 distance.

combined distance (MEAN+CHI). This is designed to handle both sufficient and insufficient shopping record situations through a convex combination of mean distance and χ^2 distance. The weight can be dynamically adjusted based on the number of shopping records. For simplicity, we set the weight to 0.5 in our implementation.

We randomly sample 10,000 distributions from all the categories and construct a similarity graph by assigning each edge with $\exp(-\frac{d^2}{\sigma^2})$, where d is one of the three distances and σ is set to be the median of all d s. We use spectral clustering [Shi and Malik 2000] to get typical price preference patterns. Nine typical price preference patterns are shown in Figure 7. Patterns 2, 3, 6, and 9 do not fit very well by single normal distribution. It is possible to use a mixture of Gaussians to fit these patterns, but we need to tune the hyperparameter of the component number carefully. Instead, we choose to use a histogram, a data-driven approach, which is simple and effective.

After we get the typical price preference patterns, we then use vector quantization technique [Farvardin 1994] to assign the user's price preference on each explored category to the nearest typical price preference pattern $\hat{p}^{(t)}$.

5.4. Matrix Representation for User Price Preference

To model user's price preference in different categories, we need category information to be preserved in the final representation. However, the typical price preference pattern is mined across categories, and its ID contains no category information. We append

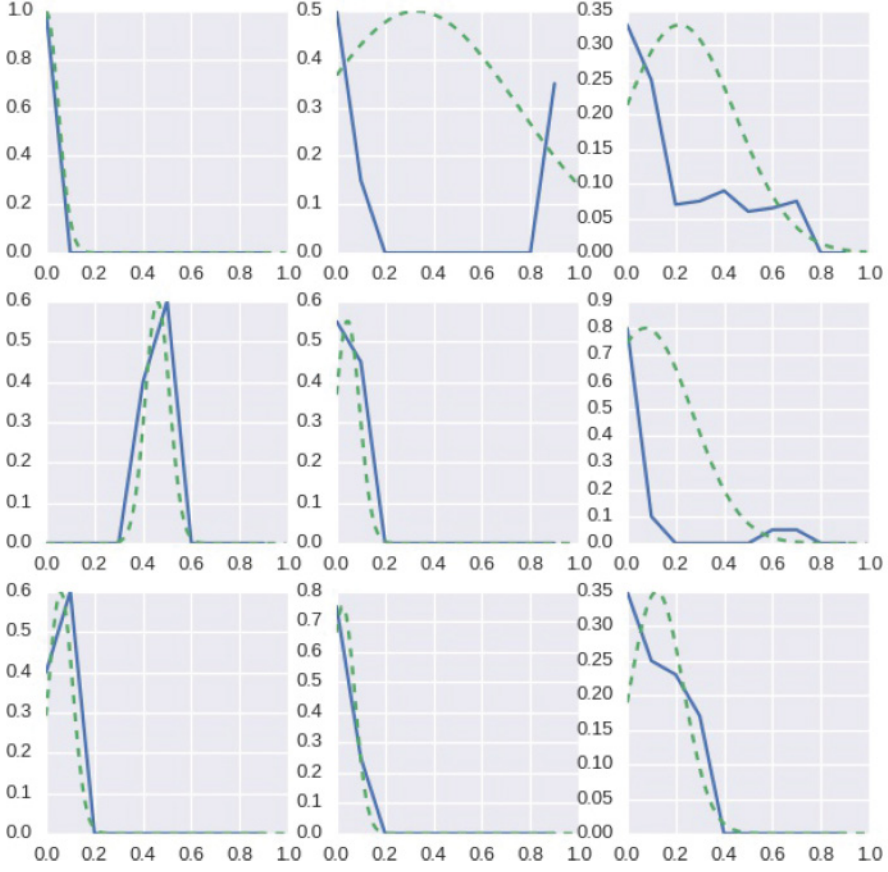


Fig. 7. Example of typical price preference patterns: Axis x represents price utility value, and axis y represents distribution value. The blue curve is the histogram and the green dashed curve is the normal distribution fitting.

category c to the assignment in the above step to recover category information, which results in $(c, \hat{p}^{(t)})$. Based on $(c, \hat{p}^{(t)})$, we first construct a user price preference profile in the user-pattern matrix. Then, based on the user-pattern matrix, we construct a user-user matrix that encodes similarity between two users on their price preference.

user-pattern matrix \mathbf{P}^* : Each row corresponds to the price preference profile of a user. Each column corresponds to a unique $(c, \hat{p}^{(t)})$. Columns are grouped by category c . For example, for two categories c_1, c_2 and two typical price preference patterns $\hat{p}^{(1)}, \hat{p}^{(2)}$, we have four columns: $(c_1, \hat{p}^{(1)})$, $(c_1, \hat{p}^{(2)})$, $(c_2, \hat{p}^{(1)})$, $(c_2, \hat{p}^{(2)})$. Generally, for m categories and n typical price preference patterns, we have $m \times n$ columns. In each row, the elements corresponding to the price preference pattern assignment $(c, \hat{p}^{(t)})$ is set to 1 while other elements are set to 0. That is, there is one and only one non-zero element in each explored category for a user in the matrix. An illustration is given in Figure 8.

user-user matrix \mathbf{S} : Each element S_{ij} represents the similarity of price preference between user i and user j . The similarity is calculated by the jaccard similarity coefficient between row i and row j in user-pattern matrix \mathbf{P}^* . The intersect part in the jaccard similarity corresponds to a shared typical price preference pattern of categories between two users. The union part in the jaccard similarity corresponds to the total categories in the two users' shopping records.

	C_1 $(C_1, \hat{p})^{(1)} \dots (C_1, \hat{p})^{(5)}$				C_3 $(C_3, \hat{p})^{(1)} \dots (C_3, \hat{p})^{(3)}$		
U_1	1	...	0	...	0	...	1
\vdots	\vdots			\ddots	\vdots		
U_3	0	...	0	...	1	...	0

Fig. 8. Illustration of matrix representation of user price preference, the user-pattern matrix \mathbf{P}^* .

6. BOOSTING RECOMMENDATIONS IN UNEXPLORED CATEGORIES BY USER PRICE PREFERENCE

In this section, we systematically study using users' price preference to boost recommendations under a series of unexplored-category conditions. According to Brown et al. [2003], we design three unexplored-category recommendation tasks from two shopping orientations: buying and browsing [Brown et al. 2003; Li et al. 2010].

In the browsing orientation, we recommend unexplored categories to users (called CR task later) to extend the coverage of their browsing activities. Once the user chooses to browse a recommended unexplored category, products under the selected unexplored category will then be recommended to users (called CIR task later). In the buying orientation, we recommend products in unexplored categories directly (called UCIR task later), without any need of category designation to generate revenue.

Figure 9 illustrates the CR, CIR, and UCIR tasks in the browsing and buying orientations. The CR and CIR tasks can be considered a two-stage decomposition of a UCIR task, but they aim at different shopping orientations of users. Thus, each of them is worth studying separately.

We answer the following three questions within each task in our study:

CR (category recommendation). Does the price preference pattern help to boost the performance of recommending unexplored categories to users?

CIR (item recommendation in the given unexplored category). Does the price preference pattern help to boost the performance of recommending items to the user when he/she is browsing an unexplored category?

UCIR (item recommendation in all unexplored categories). Does the price preference pattern help to boost the performance of recommending items in unexplored categories to the potential buyers?

We propose to fuse user price preference to improve the user latent factor of matrix factorization in two ways. The first one is regularization, which encourages user pairs with similar user price preferences to have similar user latent factors by adding constraints. The second one is joint factorization, which shares the user latent factor between the user-item matrix and the user-pattern matrix. To describe the formal definitions of these problems, we first introduce notations in Table III. The three tasks share the same mathematical formulation with different interpretations on item. The following actions are executed at the test stage in the three tasks:

CR. Given user u , we recommend unexplored category \hat{c} by $\hat{c} = \arg \max_c \mathbf{X}_u \mathbf{Y}_c$ as long as c is an unexplored category.

CIR. Given user u and unexplored category c , we recommend item \hat{i} by $\hat{i} = \arg \max_i \mathbf{X}_u \mathbf{Y}_i$ as long as i is a product in the unexplored category c .

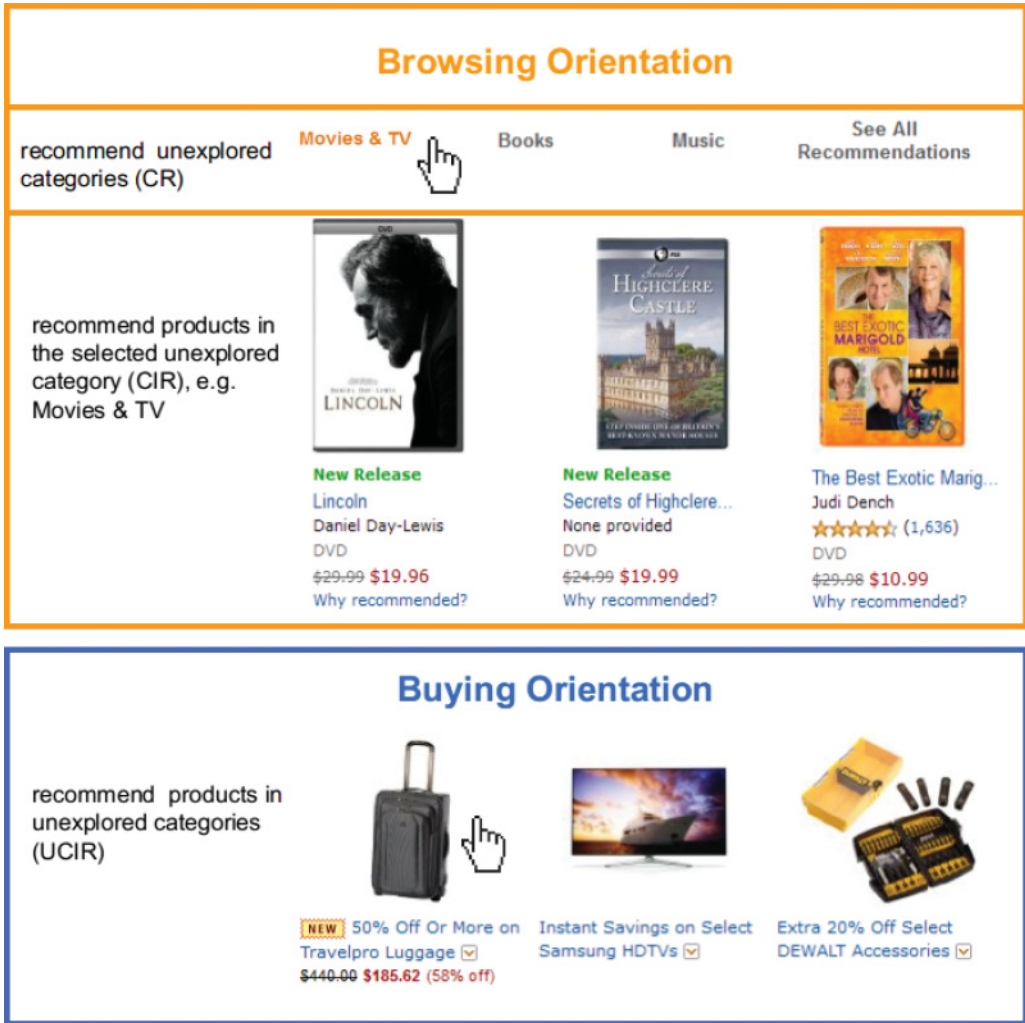


Fig. 9. Illustration of CR, CIR, UCIR tasks in browsing and buying orientations.

Table III. Summary of Notations Used in the Following Problem Formulation (an Item Can Be a Category in the CR Task and a Product in the CIR and UCIR Tasks)

u	the index of a user
i	the index of an item. The meaning of item changes in different tasks.
t	id of a typical price preference pattern
\mathbf{P}	user-item matrix. The meaning of item changes in different tasks.
\mathbf{C}	the confidence matrix corresponding to matrix \mathbf{P}
\mathbf{P}^*	user-pattern matrix which contains users' price preference pattern
\mathbf{C}^*	the confidence matrix corresponding to matrix \mathbf{P}^*
\mathbf{S}	user-user matrix which contains similarity of price preference between users
$\mathbf{X}, \mathbf{Y}, \mathbf{M}$	user-latent factor, item-latent factor, price preference latent factor. The meaning of item changes in different tasks.

UCIR. Given user u , we recommend item \hat{i} in any unexplored category by $\hat{i} = \arg \max_i \mathbf{X}_u \mathbf{Y}_i$ as long as i is a product in any unexplored category.

6.1. Fusing User Price Preference by Regularization

We learn the user latent factor \mathbf{X} from user-item matrix \mathbf{P} with the similarity constraint induced from user-user matrix \mathbf{S} . To be specific, for each user u , we first extract set $S(u)$ of 100 users that have the most similar price preference to him/her from row u in matrix \mathbf{S} . We add the constraint that the user latent factor corresponding to user u should be close to the weighted average of latent factors corresponding to $S(u)$. The weight is the similarity value in matrix \mathbf{S} . The formulas of the constraint for each user are as follows:

$$w_{uu'} = \frac{\mathbf{S}_{uu'}}{\sum_{u'' \in S(u)} \mathbf{S}_{uu''}} \quad (8)$$

$$\left\| \mathbf{X}_u - \sum_{u' \in S(u)} w_{uu'} \mathbf{X}_{u'} \right\|_F \leq \epsilon.$$

We then add the constraint as a penalty term in the objective function with penalty weight λ_p :

$$\min_{\mathbf{X}, \mathbf{Y}} \|\mathbf{C} \odot (\mathbf{P} - \mathbf{X}^\top \mathbf{Y})\|_F^2 + \lambda (\|\mathbf{X}\|_F^2 + \|\mathbf{Y}\|_F^2) + \lambda_p \left\| \mathbf{X}_u - \sum_{u' \in S(u)} w_{uu'} \mathbf{X}_{u'} \right\|_F^2, \quad (9)$$

where the parameter λ_p is used to adjust the influence of constraints.

6.2. Fusing User Price Preference by Joint Factorization

We learn the user latent factor \mathbf{X} from user-item matrix \mathbf{P} and user-pattern matrix \mathbf{P}^* jointly. Here an item is a category in the CR task and a product in the CIR/UCIR task,

$$\min_{\mathbf{X}, \mathbf{Y}, \mathbf{M}} (1 - \alpha) \|\mathbf{C} \odot (\mathbf{P} - \mathbf{X}^\top \mathbf{Y})\|_F^2 + \alpha \|\mathbf{C}^* \odot (\mathbf{P}^* - \mathbf{X}^\top \mathbf{M})\|_F^2 \quad (10)$$

$$+ \lambda (\|\mathbf{X}\|_F^2 + (1 - \alpha) \|\mathbf{Y}\|_F^2 + \alpha \|\mathbf{M}\|_F^2),$$

where the parameter α is used to adjust the influence of user-item matrix \mathbf{P} and user-pattern matrix \mathbf{P}^* on the shared user latent factor \mathbf{X} .

6.3. Solution

We perform optimization in the CR, CIR, and UCIR tasks by extending the ALS algorithm in Hu et al. [2008]. The key idea is to optimize the objective function with respect to one latent factor, while fixing the other two.

In fusing by regularization, by fixing item latent factor \mathbf{Y} , we get the formula of user latent factor \mathbf{X} as shown in Equation (11). Compared to the user latent factor without regularization in Equation (3), we see that \mathbf{X}_u is affected by not only the item latent factor \mathbf{Y} but also user latent factors from users that are similar on price preference ($u' \in S(u)$). By fixing user latent factor \mathbf{X} , we get the formula for the item latent factor, which is identical to that in Equation (3),

$$\mathbf{X}_u = (\mathbf{Y} \text{diag}(\mathbf{C}_u^\top) \mathbf{Y}^\top + (\lambda + \lambda_p) \mathbf{I})^{-1} \left(\mathbf{Y}^\top \text{diag}(\mathbf{C}_u^\top) \mathbf{P}_u^\top + \lambda_p \sum_{u' \in S(u)} w_{uu'} \mathbf{X}_{u'} \right) \quad (11)$$

$$\mathbf{Y}_i = (\mathbf{X} \text{diag}(\mathbf{C}_i) \mathbf{X}^\top)^{-1} (\mathbf{X} \text{diag}(\mathbf{C}_i) \mathbf{P}_i).$$

In fusing by use of joint factorization, by fixing item latent factor \mathbf{Y} , we get the formula for user latent factor \mathbf{X} as shown in Equation (12). Compared to the user latent factor without joint factorization in Equation (3), we see that \mathbf{X}_u is affected by not only the item latent factor \mathbf{Y} but also by the price preference latent factor \mathbf{M} . By fixing user latent factor \mathbf{X} , we calculate both item latent factor \mathbf{Y} and price preference latent factor \mathbf{M} ,

$$\begin{aligned}\mathbf{X}_u &= ((1 - \alpha)\mathbf{Y} \text{diag}(\mathbf{C}_u^\top) \mathbf{Y}^\top + \alpha \mathbf{M} \text{diag}(\mathbf{C}_u^{*\top}) \mathbf{M}^\top + \lambda \mathbf{I})^{-1} \\ &\quad ((1 - \alpha)\mathbf{Y} \text{diag}(\mathbf{C}_u^\top) \mathbf{P}_u^\top + \alpha \mathbf{M} \text{diag}(\mathbf{C}_u^{*\top}) \mathbf{P}_u^{*\top}) \\ \mathbf{Y}_i &= ((1 - \alpha)\mathbf{X} \text{diag}(\mathbf{C}_i) \mathbf{X}^\top + \lambda \mathbf{I})^{-1} ((1 - \alpha)\mathbf{X} \text{diag}(\mathbf{C}_i) \mathbf{P}_i) \\ \mathbf{M}_j &= (\alpha \mathbf{X} \text{diag}(\mathbf{C}_j^*) \mathbf{X}^\top + \lambda \mathbf{I})^{-1} (\alpha \mathbf{X} \text{diag}(\mathbf{C}_j^*) \mathbf{P}_j^*).\end{aligned}\tag{12}$$

7. EXPERIMENT

In this section, we present experiments on a series of product recommendation in unexplored category setting. Section 7.1 and 7.2 introduce the experiment settings and baselines. Sections 7.3, 7.4, and 7.5 study the performance improvement of fusing user price preference by regularization and joint factorization in three unexplored-category tasks.

7.1. Implementation Details

For the histogram of user price preference, we set the bin number to 10 as a compromise between information loss and statistical sufficiency. We tune the number of typical price preference patterns among 10, 20, 50 and find that 10 gives the best performance. We set the latent factor number to 20 for the CR task and 40 for the CIR and UCIR tasks.

7.2. Evaluation Metric and Baselines

For the category recommendation, considering that the number of ground-truth categories for each user in the test set is small, we use R-precision (RP) [Manning et al. 2008], which is the precision at the position with recall value at 1. We also measure mean average precision (MAP) [Manning et al. 2008] at top-five candidates.

For item recommendation, considering that the number of ground-truth item in the test set is much larger, we measure the overall performance by area under curve (AUC) [Gantner et al. 2010] on the top 100 predicted products,

$$\text{auc}(u) = \frac{1}{|B||X \setminus B|} \sum_{x \in B, y \in X \setminus B} \delta(r(x) < r(y)),\tag{13}$$

where B is the ground-truth items of user u . In real applications, the positions available for item recommendation is usually less than 5. For the topK item recommendation in real applications, we measure it by precision at topK, where $K = 1, 3, 5$.

In our method, we propose three strategies to transform absolute price value to price utility and three types of distances to get typical user price preference patterns. We use notation “{.}” to distinguish among the nine combinations, where the first “{” corresponds to the price utility strategy (LT, WCN, WCN+LT, introduced in Section 5.1) and the second “{” corresponds to the distance type (MEAN, CHI, MEAN+CHI, introduced in Section 5.3). We compare two different ways of adding user price preference to the basic WRMF models: regularization (UPPBoost.REG) and joint matrix factorization (UPPBoost.JOINT).

We compare our method to six baselines: most popular item recommendation (POPULAR), enhancing most popular item recommendation by the best result from category

recommendation (POPULAR-CR), collaborative filtering for implicit feedback (WRMF) [Hu et al. 2008], BPR [Rendle et al. 2009], boosting with soft assignment of user patterns (UPPBoost.SOFT), and side information embedding model UPPBoost.SIMPLE [Fang and Si 2011].

POPULAR: When in an absence of any prior purchase data in an unexplored category, recommending most popular items is a good alternative and simple in a recommender system [Grbovic et al. 2015].

POPULAR-CR: It enhances the POPULAR baseline by leveraging the best result from category recommendation (CR problem). To be specific, it recommends most popular items within the top five categories recommended by our algorithm that achieve the best performance in CR problem.

WRMF: WRMF is the state-of-the-art in product recommendation. It does matrix factorization for implicit feedback but does not consider any side information in the process.

BPR: BPR [Rendle et al. 2009] is another state-of-the-art in product recommendation. It models the ranking of user preference by sampling positive and negative pairs from the implicit feedback.

UPPBoost.SIMPLE: It extends WRMF to incorporate side information. It is similar to our UPPBoost.JOINT except that it considers the auxiliary matrix as explicit feedback. To be specific, we construct an auxiliary matrix, in which rows correspond to users and columns correspond to categories. The value of each element is the mean price of user's shopping records in the category. For unexplored categories of a user, we set the corresponding elements to the average price level on shopping records from all users under that category.

UPPBoost.SOFT: The model is exactly the same as UPPBoost.SIMPLE. The difference is that we use the user pattern matrix as auxiliary matrix. Differing from the user pattern matrix in UPPBoost.JOINT, the user pattern matrix is real valued using soft assignment. To be specific, we assign the weight by $\exp(-\frac{d^2}{\sigma^2})$, where d and σ is introduced in Section 5.2.

We categorize the above baselines to three families as follows:

Popular item recommendation family (POPULAR and POPULAR-CR): Baselines in this family are simple but effective in real-world applications [Grbovic et al. 2015]. Comparing our model to this family shows that an unexplored category needs a personalized recommendation.

User behavior data only family (WRMF and BPR): Baselines in this family are state-of-the-art algorithms that exploit only user behavior data. Comparing our model to this family shows how much price factor helps on product recommendation in unexplored categories.

Price modeling family (UPPBoost.SIMPLE and UPPBoost.SOFT): Baselines in this family focus on some different price modeling methods that differ from the ones proposed by ours. Comparing our model to this family shows the effectiveness of our price modeling.

Among these baselines, POPULAR, POPULAR-CR, UPPBoost.SIMPLE, and UPPBoost.SOFT are only meaningful in item recommendation tasks and we only compare our methods to them in CIR and UCIR problems. Furthermore, POPULAR-CR degenerates to POPULAR in CIR problem as the category is given.

7.3. Boosting Category Recommendation

In this experiment, we recommend unexplored categories to users (the CR task). The evaluation is done on MAP and RP. Both the training data and the test data are user-category pairs. As shown in Table IV, adding user price preference through joint

Table IV. Boosting Category Recommendation: Considering That the Number of Ground-Truth Categories for Each User in the Test Set Is Small, We Use the MAP and RP Metrics

methods	MAP	RP
WRMF	0.219	0.196
BPR	0.330	0.322
UPPBoost.REG	0.147	0.136
UPPBoost.JOINT	0.358	0.337

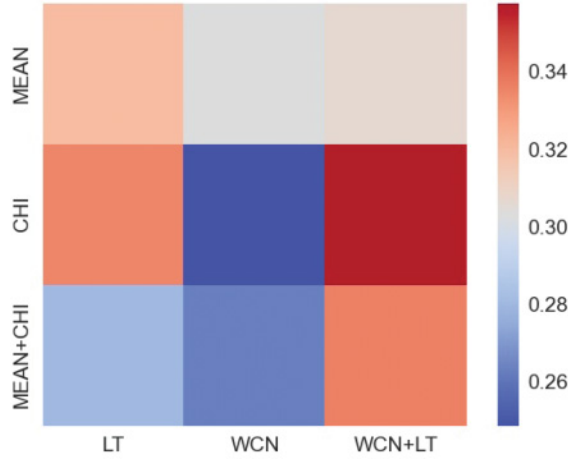


Fig. 10. Details of price utility strategy and distance type combination on performance.

matrix factorization (UPPBoost.JOINT) performs best with 0.358 on that MAP metric and 0.337 on the RP metric. It improves the best baseline (BPR) by 11.2% relatively on the MAP metric and 7.1% relatively on the RP metric. Compared to its base model WRMF, the improvement ratio is much more significant, 67.6% relatively on the MAP metric and 76.0% relatively on the RP metric.

The best price utility and distance type combination for UPPBoost.JOINT is {WCN+LT}.{CHI}. We compare the performance of nine combinations of price utility and distance type in Figure 10. Combinations {WCN+LT}.{MEAN+CHI} and {LT}.{CHI} perform slightly worse than {WCN+LT}.{CHI}. Generally comparing the columns in the heatmap, the WCN+LT price utility strategy performs best, and WCN performs worst. This indicates that considering both the relative price influence of customer psychology (LT) and the price range difference across category (WCN) is helpful to boost category recommendation. There is no general trend by comparing rows.

UPPBoost.JOINT achieves its best performance when $\alpha = 0.9$ in formula (12). Larger α means larger impact from the user price preference matrix on the performance. We further plot the impact of different α values on the performance in Figure 11. As we can see, there is a significant performance rise when α changes from 0.8 to 0.9. After analysis, we find that in the user-category matrix, the (u, c) pair only encodes a subset of information that is encoded in the user-pattern matrix element $(u, (c, p^{(t)}))$ pair. To be specific, each user-category pair in the user-category matrix is refined to user-category-pattern pair in the user-pattern matrix.

On the contrary, adding user price preference through regularization (UPPBoost.REG) actually deteriorates the performance. It performs worse than the baseline model WRMF. Based on the analysis of UPPBoost.JOINT, we know that its success

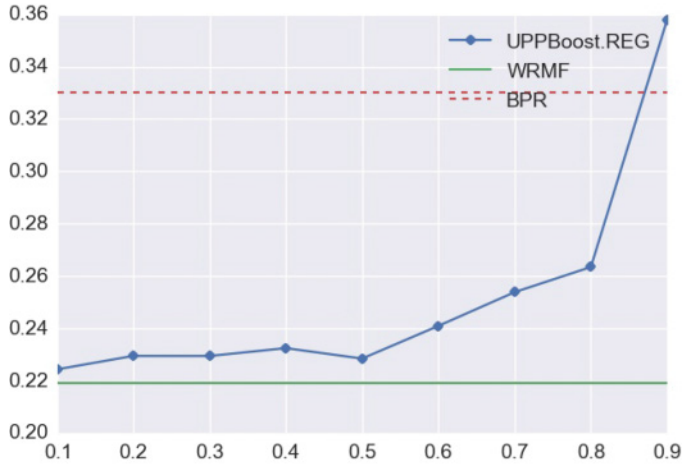


Fig. 11. Impact of user price preference on the CR task performance.

Table V. Boosting Product Recommendation under Given Unexplored Category: POPULAR-CR Degenerates to POPULAR in This Problem

methods	AUC	P@1	P@3	P@5
POPULAR	0.182	0.147	0.093	0.073
WRMF	0.311	0.224	0.146	0.118
BPR	0.304	0.184	0.123	0.098
UPPBoost.SIMPLE	0.313	0.239	0.15	0.117
UPPBoost.SOFT	0.317	0.243	0.151	0.117
UPPBoost.REG	0.328	0.231	0.150	0.120
UPPBoost.JOINT	0.329	0.227	0.147	0.119

largely attributes to the user price preference matrix, which contains far more rich information than the user-category matrix. That is, the problem with user-category matrix is that it is difficult to distinguish different user groups purely based on the shopping category history. In contrast, the regularization in UPPBoost.REG makes user latent factors more similar based on price information rather than distinguishes between different user groups. This causes the performance drop.

7.4. Boosting Product Recommendation under a Given Unexplored Category

In this experiment, we recommend products in the given unexplored category (the CIR task). Both the training data and the test data are user-product pairs. As shown in Table V, UPPBoost.JOINT, performs best on the overall item ranking (metric AUC) and UPPBoost.REG performs best on the top item ranking (metric p@5). The improvement over all baselines is significant according to a t -test at confidence level $p = 0.001$. Both UPPBoost.JOINT and UPPBoost.REG improve the baseline on both top five predictions and overall 100 predictions.

Baselines in the popular item recommendation family (POPULAR and POPULAR-CR) perform worst, which indicates that personalized recommendation is necessary for unexplored categories. For baselines in the user behavior data only family (WRMF and BPR), WRMF performs better. The baselines in the price preference modeling family (UPPBoost.SIMPLE and UPPBoost.SOFT) performs best on p@1 but not as well on AUC.

For UPPBoost.JOINT, the best strategy combination is {WCN}.{MEAN+CHI}. For UPPBoost.REG, the best strategy combination is {LT}.{MEAN+CHI}. On the top five predictions, the improvement by UPPBoost.REG is more significant than UPPBoost.JOINT. On the overall 100 predictions, the improvement by these two methods are comparable. This differs from the situation in category recommendation (the CR task), where UPPBoost.JOINT performs much better than UPPBoost.REG. After analysis, we find that it is difficult to group similar users together from user-item matrix since they rarely buy exact same items. UPPBoost.REG helps smooth similarity between user latent factors through constraints. UPPBoost.JOINT also improves user latent factor by the auxiliary matrix.

We further study the influence of parameters on the performance. For UPPBoost.JOINT, we plot the performance change with parameter α in Figure 12(a) on AUC. We see that UPPBoost.JOINT reaches its best performance on AUC with $\alpha = 0.4$ and is consistently better than baselines in the range $[0.1, 0.9]$. This shows that the improvement of UPPBoost.JOINT on AUC is robust to parameter α . For UPPBoost.REG, we plot the performance change with parameter λ in Figure 12(b) on P@1. We see that UPPBoost.REG reaches its best performance on P@1 with $\lambda = 1.0$ and is consistently better than baselines in the range $[1.0, 500.0]$. This shows that the improvement of UPPBoost.REG on P@1 is robust to parameter λ .

7.5. Boosting Product Recommendation in Unexplored Categories

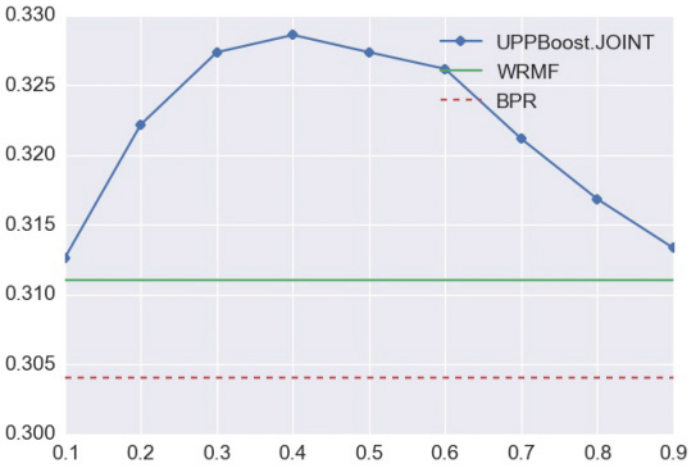
In this experiment, we recommend items in generic unexplored categories (the UCIR task). Both the training data and test data are user-product pairs. Compared to the CIR problem, this problem is more difficult since it has to determine both the category and the product for recommendation simultaneously. As shown in Table VI, UPPBoost.REG performs best on both overall ranking performance (metric AUC) and topK predictions (P@1, P@3, P@5). All improvements are significant according to t -test at confidence level $p = 0.001$.

The performance of baselines show a similar trend as that of product recommendation under a given unexplored category, except that the baselines in the price preference modeling family (UPPBoost.SIMPLE and UPPBoost.SOFT) improve the behavior data only baseline WRMF slightly, which shows the effectiveness of our price modeling. Compared to all the baselines, UPPBoost.JOINT improves the performance on overall ranking. UPPBoost.REG performs best on the all the metrics. This again shows that directly smoothing similarity between user latent factors through constraints induced by price information is beneficial for the user-item matrix.

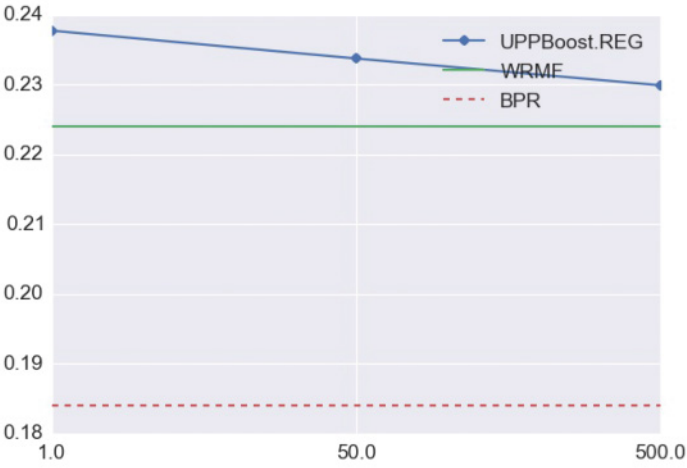
For UPPBoost.REG, the influence of price utility strategy and distance type combination is neglectable as the difference of best and worst combination is less than 0.001 on AUC. We study the influence of parameter λ on the performance. As shown in Figure 13(a), we see that it reaches the best performance when $\lambda = 50.0$, and it improves the baselines in range $[1.0, 500.0]$. This shows that the improvement of UPPBoost.REG on P@1 is robust to parameter λ . A similar conclusion also holds on the AUC metric as shown in Figure 13(b).

8. DISCUSSION

In the experiment section, we have shown quantitative improvement of recommendation by exploiting user price preference in three unexplored category-related tasks. In this section, we do a detailed case study to understand the contribution from user price preference in the recommendation improvement and discuss the merits of regularization versus the merits of joint factorization in utilizing price preference.



(a) α in UPPBoost.JOINT on AUC



(b) λ in UPPBoost.REG on P@1

Fig. 12. Impact of parameters on the CIR task performance.

Table VI. Boosting Product Recommendation under Unexplored Categories

methods	AUC	P@1	P@3	P@5
POPULAR	0.01	0.01	0.003	0.002
POPULAR-CR	0.041	0.01	0.003	0.004
WRMF	0.222	0.121	0.095	0.087
BPR	0.094	0.035	0.023	0.018
UPPBoost.SIMPLE	0.222	0.119	0.092	0.080
UPPBoost.SOFT	0.229	0.121	0.091	0.079
UPPBoost.REG	0.251	0.135	0.103	0.089
UPPBoost.JOINT	0.239	0.117	0.091	0.080

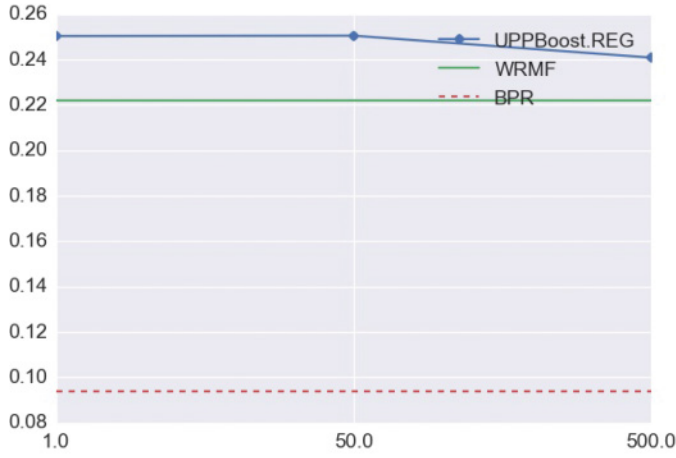
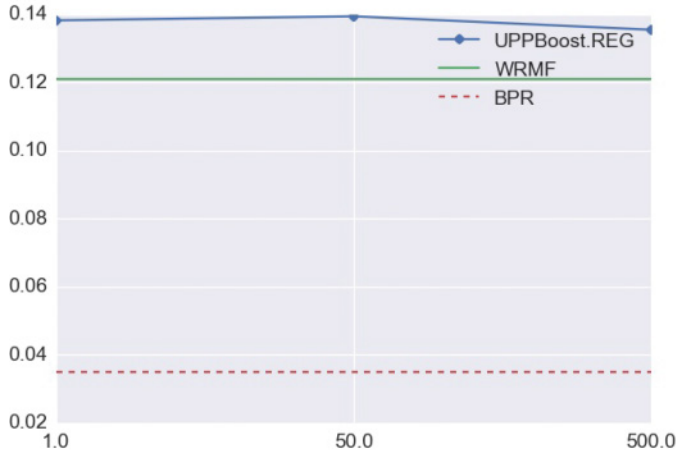
(a) λ in UPPBoost.REG on AUC(b) λ in UPPBoost.REG on P@1

Fig. 13. Impact of parameters on the UCIR task performance.

8.1. Case Study

We conduct a case study on the performance improvement with respect to the distance of user price preference from the unexplored category to explored categories. If the combined distance is smaller than 0.3, then we categorize such test data as the coherent type, and other test data are categorized as the incoherent type. Figure 14 shows the recommendation performance for two typical users, 111 and 20363. These two users are previously studied in Figure 6 in Section 5. The graph area shows the ground-truth price distribution of products that the user purchased in this category. For simplicity, we only list the best baseline WRMF and best UPPBoost method (UPPBoost.REG) in the table below the graph. It shows the MAP value and the top five predicted items.

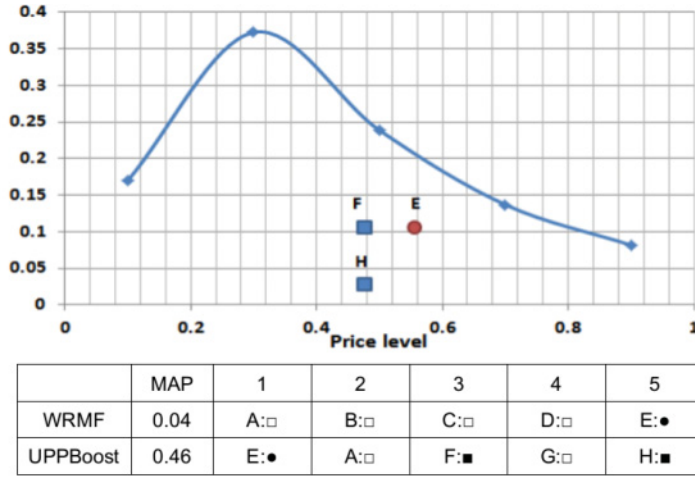
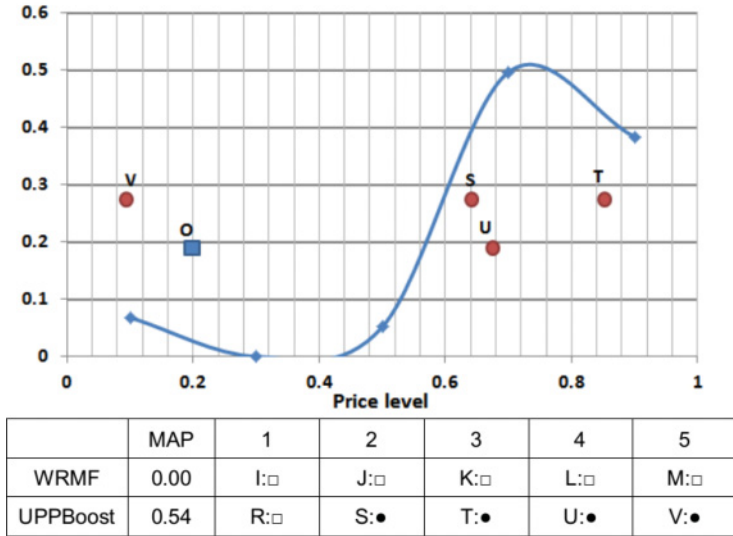
(a) coherent type: user 111 in the test category *nestcage*(b) incoherent type: user 20363 in the test category *wallet*

Fig. 14. Case study of performance on coherent and incoherent test data. The graph area shows the ground-truth price distribution and the table shows top five items predicted by two 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 utility in the graph area. □ items are not shown since they are not in the ground-truth category.

In the coherent type of test case study, we observe that although both methods have at least one good prediction (right item or right category) in the top five, UPPBoost achieves better predictions than WRMF 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 five predictions from WRMF are totally wrong (neither the right category nor the right item). UPPBoost gets 4 of five predictions right. Items *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 UPPBoost are more likely to be in the right category and at the right price utility.

8.2. Regularization vs. Joint Factorization

In Section 7, we systematically compare boosting performance by regularization and joint factorization. Here, we discuss the property and functionality of these two fusing methods. Furthermore, we think the discussion is widely applicable to user preference on any auxiliary information.

Review the experiments across the three tasks (CR, CIR, and UCIR) and two fusing methods (regularization and joint factorization), there exists two seemingly "contradiction":

Regularization *deteriorates* the performance in *the CR task* while it *improves* the performance in *the CIR and UCIR tasks*. We have given analysis and explanation for each part of the contradiction in Section 7 but have not given an overall deep analysis on the entire contradiction.

Joint matrix factorization performs *better* on the *CR task* but is *beaten* by regularization on the *CIR and UCIR tasks*. Note that the latter performs even worse than baseline on the CR task.

Here we give a deep analysis to find the rationale through by four steps:

- (1) We start from the density of three matrices: the user-category matrix (1.22%), the user-product matrix (0.14%), and the user-pattern matrix (0.43%). We see that the density of the user-pattern matrix is between that of the user-category matrix and user-product matrix. As for the user-category matrix, it is too dense to distinguish two different users just by category. However, from Equation (8), we see that the added regularizations are actually grouping users rather than distinguishing users. The density of the user-product matrix is sparser than the user-pattern matrix. Thus, adding regularization to group users helps.
- (2) A follow-up question is this: Can we add regularization to distinguish users of different preference? The hard part of the answer does not lie in *how to write down the regularization mathematically* but in *how to set the scope of users to distinguish*. Again we use Figure 5 in Section 3 for thought experiment. For example, we consider user u and sort other users based on the similarity of their price preference to user u 's. The leftmost users are users with the most similar price preference to user u . Such users definitely do not belong to the scope of users to distinguish. The rightmost users are users with no price preference intersection with user u . These users belong to the scope of users to distinguish, but they are easy cases that could probably be handled by the user-category matrix alone. The rest of the users in the middle part of the axis may belong to the scope of users to distinguish. Setting the scope by thresholding on price preference similarity is subtle, and we do not have a good criterion beforehand.
- (3) Why does fusing by joint matrix factorization work on both the user-category matrix and the user-product matrix? In joint matrix factorization, it requires the user latent factor to explain both the user-category/user-product matrix and the user-pattern matrix. Since the user-pattern matrix is sparser than the user-category matrix, it will affect the user latent factor to distinguish between different users that cannot be distinguished by the user-category matrix. The situation of the user-pattern matrix and the user-product matrix is the opposite. In a summary, the influence of the user-pattern matrix depends on its relative sparseness to the user-category matrix and the user-product matrix. That is, fusing by joint factorization enjoys more elasticity than regularization.

- (4) Finally, given the above analysis, why is joint factorization beaten by regularization on the CIR and UCIR tasks? This is because that regularization pushes the user latent factor in the correct direction, grouping rather than distinguishing, in the CIR and UCIR tasks. When both methods push the user latent factor in the correct direction, hard constraints in regularization has a more significant impact.

9. CONCLUSION AND FUTURE WORK

In this article, 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 unexplored 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 the unexplored category causes a significant performance drop for the state-of-the-art recommendation systems. Price can be a factor comparable across categories and therefore can be helpful for recommendation in unexplored categories. We propose the price utility concept to represent the sensitivity of a single product's price in the user's mind. We then further propose three different strategies to transform absolute price value to price utility. Given price utility, user price preference is a distribution, and we mine typical patterns of user price preference based on three different types of distances. We fuse user price preference through regularization and joint factorization to boost recommendation in a series of real-world settings: recommending unexplored category, recommending products under the given unexplored category, and recommending products under generic unexplored categories. Experiment results show that our approach of fusing price preference helps improve both top item recommendation and overall item ranking in all settings. Our analysis on pros and cons of fusing through regularization and joint factorization can be generalized to other properties. In the future, we will explore fusing more product properties, such as brand, to boost recommendation in unexplored categories.

REFERENCES

- Amr Ahmed, Bhargav Kanagal, Sandeep Pandey, Vanja Josifovski, Lluís Garcia Pueyo, and Jeffrey Yuan. 2013. Latent factor models with additive and hierarchically-smoothed user preferences. In *WSDM*. 385–394.
- Dan Ariely. 2008. *Predictably Irrational*. HarperCollins.
- Mark Brown, Nigel Pope, and Kevin Voges. 2003. Buying or browsing?: An exploration of shopping orientations and online purchase intention. *Eur. J. Mark.* 37, 11/12 (2003), 1666–1684.
- Hui Chen. 2012. The impact of comments and recommendation system on online shopper buying behaviour. *JNW* 7, 2 (2012), 345–350.
- Jia Chen, Qin Jin, Shiwang Zhao, Shenghua Bao, Li Zhang, Zhong Su, and Yong Yu. 2014. Does product recommendation meet its waterloo in unexplored categories?: No, price comes to help. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'14)*. 667–676.
- Tianqi Chen, Hang Li, Qiang Yang, and Yong Yu. 2013. General functional matrix factorization using gradient boosting. In *ICML*. 436–444.
- Yi Fang and Luo Si. 2011. Matrix co-factorization for recommendation with rich side information and implicit feedback. In *Proceedings of the 2Nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec'11)*. 65–69.
- Nariman Farvardin. 1994. Review of 'vector quantization and signal compression' (Gersho, A., and Gray, R.M.; 1992). *IEEE Trans. Inform. Theor.* 40, 1 (1994), 287.
- Daniel M. Fleder and Kartik Hosanagar. 2009. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Manag. Sci.* 55, 5 (2009), 697–712.
- Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. 2010. Learning attribute-to-feature mappings for cold-start recommendations. In *Proceedings of the 10th IEEE International Conference on Data Mining*. 176–185.

- Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp. 2015. E-commerce in your inbox: Product recommendations at scale. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10–13, 2015*. 1809–1818.
- Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *ICDM*. 263–272.
- Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the 2010 ACM Conference on Recommender Systems (RecSys'10)*. 135–142.
- Bernard J. Jansen. 2007. 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 (2007), 1147–1148.
- Yehuda Koren. 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *KDD*. 426–434.
- Wei Li, Xuerui Wang, Ruofei Zhang, Ying Cui, Jianchang Mao, and Rong Jin. 2010. Exploitation and exploration in a performance based contextual advertising system. In *KDD*. 27–36.
- Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM'11)*. 287–296.
- Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, United Kingdom.
- Julian J. McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 785–794.
- Andriy Mnih. 2012. Taxonomy-informed latent factor models for implicit feedback. *J. Mach. Learn. Res. Proc. Track* 18 (2012), 169–181.
- Kensuke Onuma, Hanghang Tong, and Christos Faloutsos. 2009. TANGENT: A novel, “surprise me,” recommendation algorithm. In *KDD*. 657–666.
- Seung-Taek Park and Wei Chu. 2009. Pairwise preference regression for cold-start recommendation. In *RecSys*. 21–28.
- Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of the 13th ACM International Conference on Knowledge Discovery and Data Mining*. 39–42.
- Ofir Pele and Michael Werman. 2010. The quadratic-chi histogram distance family. In *Proceedings of the 11th European Conference on Computer Vision (ECCV'10), Part II*. 749–762.
- Ian Porteous, Arthur U. Asuncion, and Max Welling. 2010. Bayesian matrix factorization with side information and dirichlet process mixtures. In *AAAI*.
- Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. 452–461.
- Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 8 (2000), 888–905.
- Ajit Paul Singh and Geoffrey J. Gordon. 2008. Relational learning via collective matrix factorization. In *KDD*. 650–658.
- Jian Wang and Yi Zhang. 2011. Utilizing marginal net utility for recommendation in e-commerce. In *SIGIR*. 1003–1012.
- Jian Wang and Yi Zhang. 2013. Opportunity model for e-commerce recommendation: Right product; right time. In *SIGIR*. 303–312.
- Diya Yang, Tianqi Chen, Weinan Zhang, Qiuxia Lu, and Yong Yu. 2012. Local implicit feedback mining for music recommendation. In *RecSys*. 91–98.
- Quan Yuan, Li Chen, and Shiwang Zhao. 2011. Factorization vs. regularization: Fusing heterogeneous social relationships in top-n recommendation. In *RecSys*. 245–252.
- Gang Zhao, Mong-Li Lee, Wynne Hsu, and Wei Chen. 2012. Increasing temporal diversity with purchase intervals. In *SIGIR*. 165–174.
- Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *WWW*. 22–32.

Received December 2015; revised May 2016; accepted July 2016