Learning Intermediary Category Labels for Personal Recommendation

WenLi Yu[1], Li Li[1,✉⋆],Jingyuan Wang[1], Dengbao Wang[1], Yong Wang[1], Zhanbo Yang[1], Min Huang[1]
[1] Faculty of Computer and Information Science, Southwest University,
Chongqing 400715, China
m13101332539@163.com,{ wjykim,wy8654,perphyoung }@email.swu.edu.cn,
wangdengbao0620@gmail.com,hmin@swu.edu.cn

Abstract. In many recommender systems, category information has been used as additional features for recommender for quite some time, whose application has tended to be understand relationships between products in order to surface recommendations that are relevant to a given context. Nevertheless, the categories as intermediary are labels for not only attributes of products but also preference characteristics of people, is ignored. Here we propose a framework to learn the intermediary role of categories acting as a bridge between users and items. The framework includes two parts. Firstly, we collect the intermediary factors that category labels affect attributes of items and user preferences respectively. Secondly, we integrate the category medium of assemble item attributes and user preferences to online recommender systems to help users discover similar or complementary products. We evaluate our framework on the Amazon product catalog and demonstrate hierarchy categories can capture characteristics of users and items simultaneously.

Keywords: Collaborative filtering; Category Labels; Latent Factors;

1 Introduction

Recommender Systems(RSs) help users to navigate a huge selection of items with unprecedented opportunities to meet a variety of special needs and user tastes. RSs based Matrix Factorization(MF) suffer from cold-start issues due to the lack of observations to estimate the latent factors of new users and items. Making use of side-signals on top of MF approaches can provide auxiliary information in cold-start settings while still maintaining MFs strengths. Such signals include the content of the items themselves, ranging from the timbre and rhythm for music [1], textual reviews that encode dimensions of opinions [2–4], or social relations [5, 6]. Moreover, knowing which items are 'similar', substitutable or complementary, is key to building systems that can understand user's context, recommend alternative items from the same style [7], or generate bundles of items that are compatible [8–10].

[⋆] Corresponding author. E-mail address: lily@swu.edu.cn.

There has been some effort to investigate taxonomy-aware recommendation models, including earlier works extending neighborhood-based methods [11]and more recent endeavors to extend MF using either explicit [12] or implicit [13–15] taxonomies. McAuley et al. [16] models substitutable and complementary product graphs with topics associated with category tree and leaf categories. He et al. [17] improves the sparse hierarchical embeddings, where the items from different parts of a category hierarchy may vary considerably.

Nevertheless, the fact that not only items have category information but also users show complementary preferences for categories in hierarchical structure, is ignored. The category labels, which users are interested in, are the latent information offering more personalized recommendations for users in RSs. For items, diverse categories label the different aspects of items contents, which are important latent factors to acquire multiple comprehensive relatedness among items to generate diverse set of recommendations. Users will choose their favorite combination among diversity categories firstly. Despite the important intermediary role of category acting between item and user, when we explore intermediary information of category labels, there are several interesting questions : How could we establish the role of category labels as the medium associating properties of products with preference characteristics of users? How would we utility the intermediary information to capture the variance across diverse categories in order to offer users functionally complementary or visually compatible products.

In the paper, a framework named Bayesian Probabilistic Matrix Factorization with Category(Category-BPMF) is proposed to explore the intermediary role of categories jointing users and items. The framework consists of two parts. Firstly, we introduce MF to factor item-category information and category-user information matrix to extract the intermediary features of category labels unifying the hierarchical structures of users' preferences and classifying principles of items correspondingly, which provide priors for user behaviors and item characteristics simultaneously. Secondly, we incorporate user-category and item-category factors as the priors of user and items inherently features matrices correspondingly, in order to produce recommendations that not only meet our needing, but also collect multiple category factors complementary for user preferences.

The paper is organized as follows: Section 2 is the problem formulation, model learning and inferring is detailed. The experiments are presented in Section 3. Section 4 is the conclusion.

2 Category BPMF

In this paper, we have three observed matrices: the rating matrix $R \in R^{M \times N}$ the latent information of categories and items $C \in R^{P \times M}$ matrix and users' tastes for categories $RC \in R^{N \times P}$ matrix. And some other notation used in the paper described in Table.1 .

Traditional methods such as Matrix Factorization [18] are usually based on a low-rank assumption. They project users and items to a low-rank latent space (D-dimensional) such that the coordinates of each user within the space capture

Table 1: Notations.

Symbol	Description
N, M , P	N users, M items, P categories
I_{ij},D	whether user i rated item j,the dimension of the latent factors space
R_{ij},C_{pj}	rating of item j by user i, item j labeled by category p
U_i, V_j	user i feature vector, item j feature vector
RC_{ip}	probability of user i preferences of category p
VP_j,UP_i	item-category feature for item j(1 × D),user-category feature for user i(1 × D)
CC_p	category feature for category p(1 × D)
Θ_U,Θ_V	the hyperparameters for user features,the hyperparameters for item features

the preferences towards these D latent dimensions. The affinity $\hat{\mathrm{R}}_{ui}$ between user u and item i is then estimated by the inner product of the vector representations of u and i:

$$\hat{\mathrm{R}}_{ui} = \langle U_u, V_i \rangle \tag{1}$$

Categories bind together the hierarchical structure of items and hierarchical structures of users' preferences. In order to recommend alternative items that are relevant to a given context from the same style, or generate bundles of items that are compatible, our objective in this section is to detect user needs across-categories and offer substitutable and complementary products fitting the user mostly. Although theoretically latent factors in Eq.1 are able to uncover any relevant dimensions, one major problem it suffers from is the existence of cold items in the system, about which there are too few associated observations to estimate their latent dimensions. Using category features extracted from category structures can alleviate this problem by providing an auxiliary signal in such situations and generate bundle recommendations of items that are compatible.

2.1 Weight Matrix of Category

Our goal is to learn the category labels furnishing features for items and users so as to recommend substitutable or complementary products for users. However, categories are organized as a hierarchical structure, seen in Fig.2, and the category of each product is a node in the tree. Here, there is a question:How would we build matrix C and RC in the framework?

We build matrix C using the following scheme: First, each product is represented by a path, or more simply a set of nodes in the category tree. For products belonging to multiple categories, we will take the union of those paths. Second, each node of the hierarchy structure has a number. The largest category, as the root node in the tree, is the number.1 in the serial number and nodes on the same layer are assigned a number one by one from high to low. The $c_{pj} = 1$ in matrix C indicate that the path of item j has the p-th node in category tree, otherwise $c_{pj} = 0$. Third, in order to give prominence to their own sub-category factors, we make the higher layer has lower weight and the leaf nodes of the path have maximum weight. For example, if item j has a category label p at the fourth layer, $c_{pj} = 1 \times 4$ and 4 is the weight for fourth layer.

Matrix RC manifests the users' attentions for multiple categories. RC is structured with the program: due to the hierarchical structure of categories and

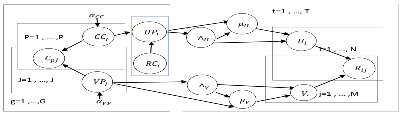

Fig. 1: The graphical model for Category-BPMF

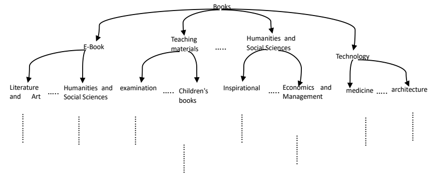

Fig. 2: Part of the product hierarchy for Amazon Books products

purchase history of users, users' preferences for category labels can be protruding by items' paths mentioning above in category tree. In spite of there would have hight level categories in comment, the lower the levels have greater weight as mentioned above. This method can succeed in finding most appropriate relationships between users and categories.

2.2 Extracting the Category Information

In the section, we will present the details of extracting category factors, which are side-signals of MF approaches providing auxiliary information in cold-start settings. As we can see on the Fig.1 left, category weights matrix C is factored to explore category feature CC_p and item-category feature VP_j. And our formulation assumes the following model to predict the importance of a category p toward an item i and user preferences for category structure:

$$\hat{\mathrm{C}}_{pj} = \langle CC_p, VP_j \rangle \ , \ and \ , UP = RC^T \cdot CC \tag{2}$$

where user-category factor UP is the variety of the set of category features CC_p and the matrix RC, which represents the user preferences for categories. The item-category factor VP represents item features connecting directly with categories. ($e_{pj} = C_{pj} - \hat{C}_{pj}$) The minimum optimization of $CC_p$and VP_jcan be approached using gradient descent method as below:

$$\begin{aligned} &\min_{VP,CC} \sum \left(C_{pj} - \hat{C}_{pj} \right)^2 + \lambda_1 \left\| \mathrm{VP} \right\|_F^2 + \lambda_2 \left\| CC \right\|_F^2 \\ &\mathrm{VP}_j = \mathrm{VP}_j + \eta (e_{pj} - \lambda_1 \mathrm{VP}_j), CC_p = CC_p + \eta (e_{pj} - \lambda_1 CC_p) \end{aligned} \tag{3}$$

2.3 BPMF with Category Factors

In the section, we will present the details of the second part of the framework. For the purpose of provide the abundant alternative fitting the heterogeneous needs of users, seen on the Fig.1 right, the core in this paper is that we derive the parameters and hyperparameters ($\Theta_U = \{\mu_U, \Lambda_U\}$,$\Theta_V = \{\mu_V, \Lambda_V\}$), of features matrices (U ,V), automatically from user-category and item-category factors UP ,VP. The conditional distribution over the user hyperparameters $\Theta_U = \{\mu_U, \Lambda_U\}$ conditioned on the user feature matrix U is given by the Gaussian-Wishart distribution of UP and U. The conditional distribution over the user feature vector U_i, conditioned on the item features V_j, observed user rating matrix R, and the values of $\Theta_U = \{\mu_U, \Lambda_U\}$ is:

$$p\left(U_i | R, V, \Theta_U, \alpha\right) = \prod_{j=1}^{M} \left[R_{ij} | \hat{\mathrm{R}}_{ui}, \alpha^{-1} \right]^{I_{ij}} p\left(U_i | \mu_U, \Lambda_U\right) \tag{4}$$

$$p\left(\mu_U, \Lambda_U | U, UP, \Theta_0\right) = N\left(\mu_U | \mu_0^*, \left(\beta_0^* \Lambda_U\right)^{-1}\right) W\left(\Lambda_U | W_0^*, \nu_0^*\right) \tag{5}$$

$$where\ \beta_0^* = \frac{\beta_0 \mu_0 + N\bar{U}}{\beta_0 + N}, \beta_0^* = \beta_0 + N, \nu_0^* = \nu_0 + N, e_U = \left(1 - \gamma^{1-t}\right), e_{UP} = \gamma^{1-t}$$

$$\left[W_0^*\right]^{-1} = W_0^{-1} + N\bar{S} + \frac{\beta_0 N}{\beta_0 + N}\left(\mu_0 - \bar{U}\right)\left(\mu_0 - \bar{U}\right)^T, \gamma > 1$$

$$\bar{U} = \frac{1}{N}\left(e_U \sum_{n=1}^{N} U_i^{t-1} + e_{UP} \sum_{i=1}^{N} UP_i\right), \bar{S} = \frac{1}{N}\left(e_U \sum_{n=1}^{N} (U_i^{t-1})^2 + e_{UP} \sum_{i=1}^{N} UP_i^2\right)$$

t equals to t, if this is the No.t iteration sampling in the Gibbs sampling algorithm. e_U and e_{UP} are the coefficients U_i and UP_i. e_U is a increasing function and e_{UP} is a decreasing function with raising of t. The coefficients for U_i and UP_i originate from two reasons:

- First, when $t = 1$ at the first iteration, $e_U = 0$, $e_{UP} = 1$ ensure the conditional distribution over $\Theta_U = \{\mu_U, \Lambda_U\}$ is given only by the Gaussian-Wishart distribution of UP as: $\bar{U} = \frac{1}{N}\sum_{i=1}^{N} UP_i$ and $\bar{S} = \frac{1}{N}\sum_{i=1}^{N} UP_i^2$.
- Second, when t>=2, $0 < e_U$, $e_{UP} < 1$ ensure the conditional distribution over $\Theta_U = \{\mu_U, \Lambda_U\}$ given by the Gaussian-Wishart distribution of comprise of UP and U^{t-1}. The component of U^{t-1} will become big with raising of t. Using U^{t-1} as prior for U^t during iterations is necessary for the convergence of user feature matrix U to deal with overfitting for regularization parameters.

The conditional distributions over the item feature vectors and $\Theta_V = \{\mu_V, \Lambda_V\}$ have exactly the same form. The samples $\left\{U_i^{(t)}, V_j^{(t)}\right\}$ are generated by running Gibbs sampling whose stationary distribution is the posterior distribution over the model parameters and hyperparameters $\{VP, UP, \Theta_V, \Theta_U\}$. The Gibbs sampling algorithm then takes the following form:

1. Initialize model parameters $\left\{U^0 V^0\right\}$
2. For t=1,...,T
   - Sample the hyperparameters (Eq.5)
     $\Theta_U^t \sim p\left(\Theta_U | \left(e_U U_i + e_{UP} UP_i\right), \Theta_0\right)$, $\Theta_V^t \sim p\left(\Theta_V | \left(e_V V_j + e_{VP} VP_j\right), \Theta_0\right)$
   - For each i = 1, ..., N sample user features and for each j = 1, ..., M sample item featuresin parallel (Eq.4):
     $U_i^{t+1} \sim p\left(U_i | R, \left(e_V V_j + e_{VP} VP_j\right), \Theta_U^t\right)$, $V_j^{t+1} \sim p\left(V_i | R, \left(e_U U_i + e_{UP} UP_i\right), \Theta_V^t\right)$

3 Experimental Results

3.1 Dataset and Evaluation Metric

To fully evaluate the ability of Category-BPMF to handle real-world tasks, we want to experiment on the largest dataset available. In this section, we adopt the dataset from Amazon[1] recently, which includes review texts and time stamps

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

spanning from May 1996 to July 2014 and each top-level category of products on Amazon.com has been constructed as an independent dataset. Statistics are shown in Table.2. Across the entire dataset, such relationships are noisy, sparse, and not always meaningful. To address issues of noise and sparsity to some extent, its sensible to focus on the relationships within the scope of a particular hight-level category. Two popular metrics, the Root Mean Square Error (RMSE) and the Mean Absolute Error(MAE), are chosen to evaluate the prediction performance.

Table 2: Dataset statistics for a selection of categories on Amazon.

Dataset	Subcategories	Items	Users	Rating	Viewed Items
Movies	44	208K	2.11M	6.17M	1.42M
Electronis	56	2498K	4.25M	11.4M	7.3M
Books	187	2.73M	8.2M	25.9M	12.46M
Women's Clothing	116	838K	1.82M	14.5M	6.35M

3.2 Baselines

- MF [18]: It is a low-rank approximation based on minimizing the sum-of-squared-errors and does not employ other information for users and items.
- Category Tree (CT): This method computes a matrix of co-occurrences between subcategories from the training data. Then a pair (x, y) is predicted to be positive if the subcategory of y is one of the top 50% most commonly connected subcategories to the subcategory of x.
- Bayesian Personalized Ranking with Category Tree(BPR-C): Introduced by [19], BPR is the state-of-the-art method for personalized ranking. BPR-C makes use of category tree to extend BPR by associating a bias term to each finegrained category on the hierarchy.
- Item-to-Item Collaborative Filtering (CF) [8]: This baseline identifying items that had been browsed or purchased by similar sets of users, follows the same procedure, actual browsing or purchasing data we consider sets of users who have reviewed each item.

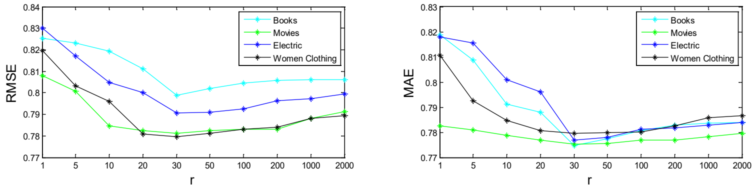

Fig. 3: Impact of parameter γ

3.3 Experimental Results

Impact of parameter analysis the base of exponential function γ: γ is base of exponential function γ^{1-t} regulating proportion of $UP_i$and U_i, VP_j and V_j. Fig.3 shows the impacts of γ on MAE/RMSE with D=20 for four datasets. As γ increases, the two error metrics decrease, but when γ surpasses a value 30, all the metrics increase. We observe that error metrics have the minimum for about $\gamma = 30$ simultaneously. So $\gamma = 30$ can control the strength of Category-BPMF well.

Analysis of Recommendation Performance: All the experiments are conducted using 20 latent factor dimensionality, $\gamma = 30$ and 50 iterations. We can make a few observations to explain and understand our finding as follows:

Dataset	Setting	CT	MF	CF	BPR-C	Category BPMF	% impr. *C-BPMF vs CF*	% impr. *C-BPMF vs BPR-C*
Movies	All items	0.917	0.842	0.854	0.801	**0.771**	8.54	3.74
	cold start	0.945	0.856	0.868	0.812	**0.774**	8.52	4.67
Electronis	All items	0.912	0.839	0.833	0.798	**0.768**	7.8	3.71
	cold start	0.935	0.855	0.857	0.805	**0.764**	10.82	4.67
Women's Clothes	All items	0.908	0.835	0.821	0.792	**0.776**	5.7	2.02
	cold start	0.931	0.840	0.842	0.810	**0.778**	3.8	3.95
Books	All items	0.906	0.832	0.844	0.795	**0.775**	8.17	2.51
	cold start	0.941	0.848	0.838	0.812	**0.774**	7.6	4.67

Table 3: RMSE of the predictions on all items or cold start on four datasets.

1) CT is particularly inaccurate for our task. We also observed relatively high training errors with this method for most experiments. This confirms our conjecture that raw similarity is inappropriate for our task, and that in order to learn the relationships across categories, some sort of expressive transforms are needed for manipulating the raw features. 2) MF performs considerably worse than other methods. This reveals that the predictive information used by the other models goes beyond the features of the products, i.e., that the category based models are learning relationships between finer grained attributes. 3) BPR-C does significantly outperform CT, presumably because the category signals are already encoded by those features. This suggests that improving BPR requires more creative ways to leverage such signals. 4) Category-PMF, BPR-C compared with MF and CF indicate that category information is very important information and can not be ignored. 5) Category-BPMF consistently outperforms baseline methods. These results suggest that hierarchical categories of users and items contain complementary information and capturing them simultaneously can further improve the recommendation performance.

4 Conclusion

In this paper, we exploit the hierarchical category structures of items and users for recommendation when they are not explicitly available and propose a novel recommendation framework Category-BPMF, which captures the hierarchical structures of items and users into a coherent model. Experimental results on real-world dataset demonstrates the importance of the implicit hierarchical categories of both items and those of users in the recommendation systems.

5 Acknowledgments.

This work is supported by NSFC(No.61170192) and National Science and Technology Support Program (No.2015BAK41B01).

References

1. X Wang, Y Wang, D Hsu, and Y Wang. Exploration in interactive personalized music recommendation:a reinforcement learning approach. *TMCCA*.
2. J Mcauley and J Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*, pages 165–172, 2013.
3. YF Zhang, G Lai, and M Zhang. *Explicit factor models for explainable recommendation based on phrase-level sentiment analysis.* ACM, 2014.
4. G Ling, R. Lyu, and I King. Ratings meet reviews, a combined approach to recommend. pages 105–112, 2014.
5. T Zhao, J Mcauley, and I King. Leveraging social connections to improve personalized ranking for collaborative filtering. pages 261–270, 2014.
6. W Pan and L Chen. Gbpr: group preference based bayesian personalized ranking for one-class collaborative filtering. In *IJCAI*, pages 2691–2697, 2013.
7. D Hu, R Hall, and J Attenberg. Style in the long tail: Discovering unique interests with latent variable models in large scale social e-commerce. In *SIGKDD*, pages 1640–1649. ACM, 2014.
8. G Linden, B Smith, and J York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.
9. J McAuley, C Targett, and Q Shi. Image-based recommendations on styles and substitutes. In *SIGIR*, pages 43–52. ACM, 2015.
10. J Zheng, X Wu, and J Niu. Substitutes or complements: another step forward in recommendations. In *ACM*, pages 139–146. ACM, 2009.
11. C Ziegler, G Lausen, and L Schmidt-Thieme. Taxonomy-driven computation of product recommendations. In *ACM*, pages 406–415. ACM, 2004.
12. Andriy Mnih. Taxonomy-informed latent factor models for implicit feedback. In *KDD Cup*, pages 169–181, 2012.
13. Y Zhang, A Ahmed, and V Josifovski. Taxonomy discovery for personalized recommendation. In *ACM*, pages 243–252. ACM, 2014.
14. A Mnih and Y Teh. Learning label trees for probabilistic modelling of implicit feedback. In *ANPIS*, pages 2816–2824, 2012.
15. S Wang, J Tang, Y Wang, and H Liu. Exploring implicit hierarchical structures for recommender systems. In *IJCAI*, pages 1813–1819, 2015.
16. J. J. McAuley, R. Pandey, and J. Leskovec. Inferring networks of substitutable and complementary products. In *KDD*, 2015.
17. R. He, C. Packer, and J. McAuley. Learning compatibility across categories for heterogeneous item recommendation. In *ICDM*, 2016.
18. R Salakhutdinov and A Mnih. Probabilistic matrix factorization. In *Nips*, volume 1, pages 2–1, 2007.
19. S Rendle, C Freudenthaler, and Z Gantner. Bpr: Bayesian personalized ranking from implicit feedback. In *CUAI*, pages 452–461, 2009.