Matrix Factorization for Package Recommendations

Agung Toto Wibowo
Computing Science / Informatics Engineering
University of Aberdeen / Telkom University
wibowo.agung@abdn.ac.uk /
agungtoto@telkomuniversity.ac.id

Advaith Siddharthan
Knowledge Media Institute
The Open University
advaith.siddharthan@open.ac.uk

Chenghua Lin
Computing Science
University of Aberdeen
chenghua.lin@abdn.ac.uk

Judith Masthoff
Computing Science
University of Aberdeen
j.masthoff@abdn.ac.uk

ABSTRACT

Research in recommendation systems has to date focused on recommending individual items to users. However there are contexts in which combinations of items need to be recommended, and there has been less research to date on how collaborative methods such as matrix factorization can be applied to such tasks. The research contributions of this paper are threefold. First, we formalize the collaborative package recommendation task as an extension of the standard collaborative recommendation task. Second, we describe and make available a novel package recommendation dataset in the clothes domain, where a combination of a "top" (e.g. a shirt, t-shirt or top) and "bottom" (e.g. trousers, shorts or skirts) needs to be recommended. Finally, we describe several extensions of matrix factorization to predict user ratings on packages, and report RMSE improvements over the standard matrix factorization approach for recommending combinations of tops and bottoms.

KEYWORDS

Package Recommendation, Matrix Factorization, Clothes Domain, Collaborative Filtering

1 INTRODUCTION

Recent research into recommendation systems has focused on methods for Collaborative Filtering (CF) [5, 20] for tasks such as recommending individual or top-N items to users [8] and for making cross-domain recommendations [3, 12, 18].

There has been less research into package recommendations, where a combination of items needs to be recommended together. Travel is one domain that is mentioned in the literature [6, 13], where a travel package could consist of a set of destinations and is often recommended to a group of users. For example, in a travel planning task, a user (or group) is recommended a package of places of interest (POI) which satisfy some constraints such as budget or time [22, 23]. Such travel recommender systems need to be able to handle constraints, e.g. "no more than 3 museums" or "travel distance is less than 10 km". Another task is to provide alternatives for restaurants, transportation and hotels as POI [1].

ComplexRec 2017, Como, Italy.

Outside of travel/tourism there are several other domains, such as food (e.g. recommending a starter and main course), furniture [25] and clothing (e.g. recommending a shirt and trousers), which offer good opportunities for recommending packages.

In the clothes domain, there are some package recommendation approaches based on image features [7, 17]. These approaches collected images (each image containing both top and bottom) from fashion websites [17] or fashion magazines [7] to create a package reference database. Using image processing techniques, they automatically separated top and bottom. Miura et al. [17] extracted image features (such as RGB histogram and scale invariant features transform [SIFT] [14] values) for both top and bottom. To provide package recommendations, they required the user to provide a query (top or bottom) image. This image was then compared with packages in the reference database, and the closest package reference returned as a recommendation. Similar to Miura's work, Iwata et al. [7] extracted visual features (such as colour, texture and SIFT as a bag-of-features, and derived a topic model over these using Latent Dirichlet Allocation (LDA). When a user provided a query image (top/bottom), Iwata et al. recommended the other part by searching the topic model in their package reference database.

Shen et al. [19] developed a clothes package recommendation system based on user context. First, they stored clothing items and combinations of items in a user wardrobe database. They also annotated its contents using English words. To generate recommendations, their system asked the user about their goals ("destinations" and "want to look like") and mapped them to possible characteristic of clothes in the user wardrobe.

With respect to recommendations, all these methods have the following drawbacks: (a) they work from a fixed reference database, with no flexibility for recommending combinations not in the database; (b) the recommendations provided from the database are not tailored to user preferences (though Shen et al. allow the user to specify some aspects of the style); and (c) The methods are highly tailored to the clothes domain and cannot be readily applied to package recommendations in other domains.

To overcome these drawbacks, we formalize package recommendations as a collaborative filtering task and argue that collaborative package recommendation is an interesting task for three reasons. First, collaborative package recommendation is more challenging than item recommendation since people might dislike a package, even if they like the individual items. Such preferences can reflect

Table 1: Mathematical Notations

Notations	Descriptions
m	number of users
o	number of "top" items
p	number of "bottom" items
u_x	a user u with id x
i_y^c	an item i with id y from category c
(i_x^t, i_y^b)	a package of items consist of a top i_x^t, and a bottom i_y^b
$U = \{u_1, u_2, \cdots, u_m\}$	a set of users
$I^t = \{i_1^t, i_2^t, \cdots, i_o^t\}$	a set of "top" items
$I^b = \{i_1^b, i_2^b, \cdots, i_p^b\}$	a set of "bottom" items
$I^a = I^t \cup I^b$	a set of all individual items
$(u, i, r_{u,i})$	individual rating in triplet format from user u to item i
$\hat{r}_{u,i}$	predicted rating from user u to item i
$(u, i^t, i^b, r_{u,(i^t,i^b)})$	a package rating in quadruple format from user u to a combination of (i^t, i^b)
V^t	matrix of individual "top" ratings, where $\|rows\| = m$ (no. of users); $\|columns\| = o$ (no. of tops)
V^b	matrix of individual "bottom" ratings, where $\|rows\| = m$; $\|columns\| = p$ (no. of bottoms)
V^a	matrix of individual ratings for top and bottom, where $\|rows\| = m$; $\|columns\| = o + p$,
V^p	matrix of package ratings, where $\|rows\| = m$; $\|columns\| = o \times p$

individual taste and style, and recommendations therefore need to be personalized to users. Second, package recommendations face greater data sparsity issues compared to the collaborative item recommendation task. The number of possible combinations is large and for the same number of user ratings, the package recommendation matrix is much sparser than for item recommendation. Third, unlike the previous work described above, we can easily extend our package recommendation approach to other domains (such as food, etc.) by formulating package recommendation as collaborative filtering.

The remainder of this paper is organized as follows. Section 2 defines the package recommendation task and the notation used, describes how the dataset was generated, and formulates several Matrix Factorization approaches for package recommendation. Section 3 details our experimental settings, Section 4 reports our experiment results, and Section 5 provides a discussion and suggests directions for future work.

2 PACKAGE RECOMMENDATIONS

2.1 Definition

The traditional collaborative filtering (CF) [5, 20] task is defined as predicting the ratings given by users to items, based on a set of previous ratings by any user to any item. In this paper, we introduce a collaborative filtering task for package recommendations, where we need to predict ratings given by a user to combinations of items, based on a set of previous ratings by any user to any item or combination of items.

In this paper, we discuss package recommendation for the clothes domain. Consider a set of clothes $I^a = \{i_1^t, i_2^t, \ldots, i_o^t, i_1^b, i_2^b, \ldots, i_p^b\}$, consisting of two disjoint complementary sets: a set of o top items $I^t = \{i_1^t, i_2^t, \ldots, i_o^t\}$ and a set of p bottom items $I^b = \{i_1^b, i_2^b, \ldots, i_p^b\}$, where $I^t \cup I^b = I^a; o + p = n$. Some of these items and their combinations (a package) have received ratings from one or more of m possible users $U = \{u_1, u_2, ..., u_m\}$. In our notation, individual ratings are denoted as a triplet $(u, i, r_{u,i})$, where $u \in U$, $i \in I^a$ and $r_{u,i}$ is the rating given by user u to item i. Package ratings are denoted as a quadruple $(u, i^t, i^b, r_{u,(i^t,i^b)})$, where $u \in U$, $i^t \in I^t$, $i^b \in I^b$, and $r_{u,(i^t,i^b)}$ is the rating provided by user u to the package (i^t, i^b). Our task is to predict the unobserved package ratings for a user from an observed set of ratings for individual items and packages by this and other users. This definition is easily extended to other domains and to tasks which might involve more than two items within a package.

2.2 Dataset Generation

A bottleneck to research on package recommendations is the lack of open datasets suited for this task. To overcome this, we generated a dataset by randomly selecting 1,400 "top" and 600 "bottom" images from Amazon product data [15, 16] and obtaining 30 ratings each from 200 participants recruited from Amazon Mechanical Turk for individual tops and bottoms and packages combining them.

For each participant, we first asked them whether they wear clothes for men or women, and then provided 30 screens where each screen showed images of one top and one bottom filtered for their chosen gender preference. We also asked participants to rate on a scale of 1 to 5 how much:

(1) they would like to wear the top,
(2) they would like to wear the trousers,
(3) they would like to wear the top and trousers together.

An example can be seen in Figure 1. From our participants, we obtained 12,000 individual ratings and 6,000 package ratings. We have made this dataset freely downloadable from the *PackageRec-Dataset* Github repository[1].

The distribution of ratings for our data set are shown in Table 2. Note that the percentage of highly rated packages is much lower than that of either tops or bottoms, which makes package recommendation a more challenging task.

2.3 Matrix Factorization Methods

2.3.1 Matrix Factorization for Item Recommendations. In collaborative filtering, there are many approaches to provide rating predictions. Some approaches calculate similarity between users or items [5, 20], while other approaches use matrix factorization techniques to decompose the rating matrix into two (or more) matrices. The first winner [9] of the Netflix prize reported that matrix

[1] https://github.com/atwRecsys/PackageRecDataset

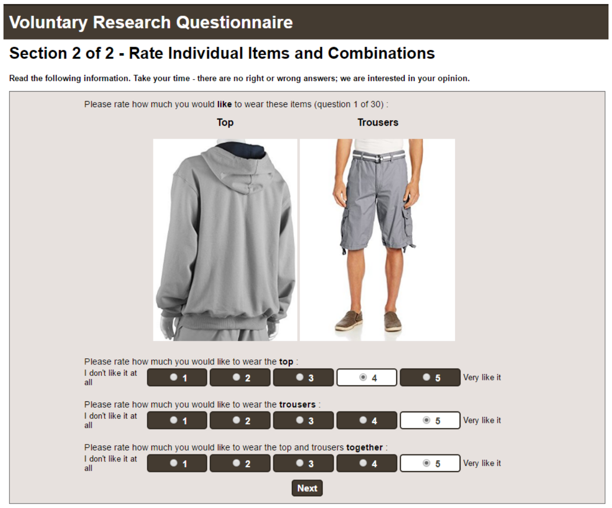

Figure 1: Example of Clothes Questionnaire

Table 2: Rating Distributions

	Rating Frequency				
	1	2	3	4	5
Tops	1,710	1,080	1,169	1,167	874
Bottoms	1,574	958	1,185	1,282	1,001
Packages	2564	1216	1060	760	400

factorization has many benefits for overcoming common problems in recommender systems such as data sparsity and cold start [24].

Matrix factorization (MF) can be defined as producing two factor matrices, say $W = [w_{ij}] \in \mathbb{R}^{m \times k}$ and $H = [h_{ij}] \in \mathbb{R}^{k \times n}$ from one known matrix $V = [v_{ij}] \in \mathbb{R}^{m \times n}$, so the product of W and H are (approximately) equal to V:

$$V \approx WH \,, \tag{1}$$

where each cell is computed as:

$$\hat{v}_{xy} \approx \sum_{i=1}^{k} w_{xi} h_{iy} \tag{2}$$

There are many algorithms for MF, such as Multiplicative [10, 11], Gradient descent [2, 4], Alternating Least Square [2, 4], and more. These algorithms aim to minimize the difference between the known values in matrix V and the corresponding values in its multiplicative form WH (the cost function) through an iterative process. When the factors W and H are computed in this manner, it has been found that the product WH provides values for missing cells in V, and that these turn out to be good estimates of these missing ratings.

2.3.2 Extending MF for Package Recommendations. From the definitions in Table 1, there are four different matrices that can be input to matrix factorization methods (V^t, V^b, V^a, V^p). In this paper, we utilized these inputs in seven different scenarios:

(1) In our first scenario (*Average Predictor*), we used the average value of each package in V^p (the matrix of user $[m]$ and packages $[o \times p]$) as prediction. We used this scenario as a baseline.

(2) In our second scenario (*MF-Package*), we used V^p as input and ran MF over this package rating matrix. Using this scenario, we obtained two latent matrices W^p and H^p, which when multiplied together provided ratings for missing cells. This is our second baseline.

(3) In our third scenario (*MF-Pseudo*), to address the matrix sparsity issue in the baseline above, we first populated V^p by adding some pseudo-ratings ($r'_{u,(i^t,i^b)}$) into $V^{p'}$, before then applying MF to the matrix. Starting from a rating by a user for a package, we identified similar packages involving a new item (either top or bottom) where the cosine similarity between the new and known item was more than specified threshold.

Consider a known package rating $(u, i^t_x, i^b_y, r_{u,(i^t_x,i^b_y)})$. For each top item i^t_z in the matrix, we added a package pseudorating $r'_{u,(i^t_z,i^b_y)}$ where $r'_{u,(i^t_z,i^b_y)} = r_{u,(i^t_x,i^b_y)}$ if cossim(i^t_x, i^t_z) $\geq \theta$. Likewise, for each bottom item i^b_s we added a package pseudorating $(u, i^t_x, i^b_s, r'_{u,(i^t_x,i^b_s)})$, where $r'_{u,(i^t_x,i^b_s)} = r_{u,(i^t_x,i^b_y)}$ if cossim(i^b_y, i^b_s) $\geq \theta$. After we added these pseudoratings, we ran MF and obtained $W^{p'}$ and $H^{p'}$. The package rating predictions are generated by multiplying these matrices.

(4,5) In our fourth and fifth scenarios (*MF-Min-Cat* and *MF-Mul-Cat*), we ran MF individually over the user–top (V^t) and user–bottom (V^b) matrices. From V^t we obtained W^t and H^t, and from V^b we obtained W^b and H^b. The package rating predictions $\hat{r}_{u,(i^t,i^b)}$ were obtained in two ways: (a) *MF-Min-Cat* predicted package ratings using the minimum value of $\hat{r}_{u,i^t}$ and $\hat{r}_{u,i^b}$; (b) *MF-Mul-Cat* predicted package ratings using the harmonic mean of $\hat{r}_{u,i^t}$ and $\hat{r}_{u,i^b}$ (Equation 3).

$$harmonic\_mean(a, b) = \frac{a \times b}{\frac{1}{2}(a + b)} \tag{3}$$

(6,7) In our sixth and seventh scenarios (*MF-Min-All* and *MF-Mul-All*), we ran MF over all individual rating matrix (V^a). From this process, we obtained W^a and H^a. The package rating predictions $\hat{r}_{u,(i^t,i^b)}$ were obtained in two ways: (a) *MF-Min-All* predicted package rating using the minimum value of $\hat{r}_{u,i^t}$ and $\hat{r}_{u,i^b}$; (b) *MF-Mul-All* predicted ratings using the harmonic mean of $\hat{r}_{u,i^t}$ and $\hat{r}_{u,i^b}$ (Equation 3).

To summarise, scenaro 1 applies *Average Predictor* baseline over V^p, which takes the average rating for each item; scenario 2 applies MF over the user–package matrix; and scenarios 3–7 apply MF to the user–item matrices and combine predicted ratings of items in the package using either a minimum or a harmonic mean function.

3 EXPERIMENTAL SETTINGS

3.1 Crossvalidation Method

Given r_{u,i^t} as a "top" rating, r_{u,i^b} as a "bottom" rating and $r_{u,(i^t,i^b)}$ as "package" rating, there are six possibilities:

(1) We do not know $r_{u,(i^t,i^b)}$, but we know r_{u,i^t} or r_{u,i^b};
(2) We do not know $r_{u,(i^t,i^b)}$, but we know r_{u,i^t} and r_{u,i^b};
(3) We know $r_{u,(i^t,i^b)}$, but we only know one of r_{u,i^t} and r_{u,i^b};
(4) We know $r_{u,(i^t,i^b)}$, but do not know either r_{u,i^t} or r_{u,i^b};
(5) We know r_{u,i^t}, r_{u,i^b}, and $r_{u,(i^t,i^b)}$.
(6) We do not know r_{u,i^t}, r_{u,i^b}, or $r_{u,(i^t,i^b)}$.

Our dataset collected ratings for "top", "bottom" and "package" together; thus for any user only (5–6) are possible. However (1–4) are realistic scenarios for a package recommendation system. To cover all these possibilities, we adopted the following methodology. First, we used 4-fold crossvalidation by randomly splitting the individual ratings into four parts. We rotated and used 3 parts as the training set and one for testing. Then in each fold we used only 25% of package ratings $r_{u,(i^t,i^b)}$ as the training set, and the remaining 75% package ratings $r_{u,(i^t,i^b)}$ as the test set. These mechanisms for holding back data to make package predictions ensure that all possibilities are covered in a realistic manner.

3.2 Experimental Settings

We used matrix factorization with gradient descent [21], with 100 iterations. In this experiment we varied k, the number of latent dimensions in MF (k = 5, 10, 15, 20). We also varied the threshold value for similarity when adding pseudo-transactions in *MF-Pseudo* using values (θ = 0.5, 0.7, 0.9).

We report the average RMSE performance over the test sets in each fold, as defined in Section 3.1. In addition, we also report the average RMSE performance for each known rating. As we can see in Table 2, users tended to give low ratings for package recommendations, and such low-rated packages dominate the dataset. However, for a recommendation task, we are primarily concerned with accurately predicting the highly rated packages. The table thus allows for comparison of algorithms on the more realistic task.

In real world situations, we are usually interested in providing top-N packages to users. This sort of evaluation is unfortunately not possible for our dataset, as mechanical turkers were given random combinations to rate, and were not allowed to choose items or packages they liked. Though out of scope for this paper, we would in the future like to evaluate package recommendations using a rank performance metric in a real domain.

4 RESULTS

Table 3 shows the average RMSE for the testing set for different scenarios. The "All" column represent the overall RMSE, while the 1, 2, 3, 4, and 5 columns represent the average RMSE over the known package ratings. For example, column "1" represent average RMSE to $r_{u,(i^t,i^b)} = 1$. The yellow cells in this table show our baseline RMSE over the package testing set.

All our adaptations outperform the *MF-Package* baseline (lower RMSE values in the "All" column), and many of them outperform the *Average Predictor*. *MF-Min-Cat* has the best overall performance (the green cell in the "All" column). In our dataset, people overall gave lower rating for packages than for individual items. Estimating the package rating as the minimum of the individual item rating predictions therefore gives better results overall, but increased errors for highly rated packages that we would want to recommend.

Table 3: Average Testing set RMSE Performance

		Average Package Rating RMSE				
Scenario	All	1	2	3	4	5
Average Predictor	1.234	1.146	0.200	0.770	1.715	2.670
MF-Package(Base)	1.435	1.396	0.949	0.974	1.607	2.372
MF-Pseudo(θ = 0.5)	1.296	1.137	0.825	1.060	1.755	2.429
MF-Pseudo(θ = 0.7)	1.330	1.166	0.841	1.076	1.762	2.481
MF-Pseudo(θ = 0.9)	1.395	1.319	0.922	1.009	1.637	2.403
MF-Min-Cat	1.154	1.242	0.715	0.734	1.338	1.893
MF-Mul-Cat	1.218	1.485	0.891	0.601	1.067	1.591
MF-Min-All	1.166	1.327	0.776	0.686	1.233	1.766
MF-Mul-All	1.237	1.537	0.940	0.599	1.013	1.515

The pseudo-ratings approach reducing sparsity in the package matrix and the minimum function for combining item ratings performed better at predicting low ratings. *MF-Pseudo* and *MF-Min-Cat* outperform other scenarios for low ratings (marked as green cells in the "1" and "2" columns). *MF-Pseudo* increases matrix density by populated the package rating matrix with some pseudo-ratings based on similarity to known packages. Since low package ratings are frequent, *MF-Pseudo* might get a stronger signal to predict low ratings rather than higher ones.

For highly rated packages, on the other hand,the multiplicative methods for combining individual ratings performed better. *MF-Mul-All* outperformed other approaches for the high ratings (marked as green cells in the "3", "4" and "5" columns). This is not unexpected, as when we combine two ratings for "top" and "bottom", the harmonic mean (*MF-Mul-All*) will by definition give a slightly higher estimate than the minimum (*MF-Min-All*).

5 CONCLUSIONS AND FUTURE WORK

We have defined a new task of collaborative package recommendation and made available the first public dataset for this task. We have also suggested several adaptations of the standard matrix factorization approach to item recommendation. All the adaptations outperform the standard MF baseline, and different adaptations demonstrated strengths in different situations.

Our work can be extended in a couple of ways. One is take into account item attributes (such as colour, dresscode, etc.), and user attributes (such as gender and age, etc) within a tensor factorization framework. We would also like to extend our clothes package recommendations by adding other categories (such as accessories) and also investigate the package recommendation methods in other domains, such as food, where we might additionally consider constraints such as allergens and nutrition.

ACKNOWLEDGMENTS

We would like to thank Lembaga Pengelola Dana Pendidikan (LPDP), Departemen Keuangan Indonesia for awarding a scholarship to support the studies of the lead author.

REFERENCES

[1] Sihem Amer-Yahia, Francesco Bonchi, Carlos Castillo, Esteban Feuerstein, Isabel Mendez-Diaz, and Paula Zabala. 2014. Composite retrieval of diverse and complementary bundles. *IEEE Transactions on Knowledge and Data Engineering* 26, 11 (2014), 2662–2675.

[2] Michael W Berry, Murray Browne, Amy N Langville, V Paul Pauca, and Robert J Plemmons. 2007. Algorithms and applications for approximate nonnegative matrix factorization. *Computational statistics & data analysis* 52, 1 (2007), 155–173.

[3] Iván Cantador, Ignacio Fernández-Tobías, Shlomo Berkovsky, and Paolo Cremonesi. 2015. Cross-domain recommender systems. In *Recommender Systems Handbook*. Springer, 919–959.

[4] Moody Chu, Fasma Diele, Robert Plemmons, and Stefania Ragni. 2004. Optimality, computation, and interpretation of nonnegative matrix factorizations. In *SIAM Journal on Matrix Analysis*. Citeseer.

[5] Michael D Ekstrand, John T Riedl, and Joseph A Konstan. 2011. Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction* 4, 2 (2011), 81–173.

[6] A Felfernig, S Gordea, D Jannach, E Teppan, and M Zanker. 2007. A short survey of recommendation technologies in travel and tourism. *OEGAI journal* 25, 7 (2007), 17–22.

[7] Tomoharu Iwata, Shinji Wanatabe, and Hiroshi Sawada. 2011. Fashion coordinates recommender system using photographs from fashion magazines. In *IJCAI*, Vol. 22. Citeseer, 2262.

[8] George Karypis. 2001. Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the tenth international conference on Information and knowledge management*. ACM, 247–254.

[9] Yehuda Koren. 2009. The Bellkor Solution to the Netflix Grand Prize. *Netflix prize documentation* 81 (2009).

[10] D. D. Lee and H. S. Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 6755 (1999), 788–791.

[11] D. D. Lee and H. S. Seung. 2001. Algorithms for non-negative matrix factorization.

[12] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction.. In *IJCAI*, Vol. 9. 2052–2057.

[13] Qi Liu, Enhong Chen, Hui Xiong, Yong Ge, Zhongmou Li, and Xiang Wu. 2014. A cocktail approach for travel package recommendation. *IEEE Transactions on Knowledge and Data Engineering* 26, 2 (2014), 278–293.

[14] David G Lowe. 1999. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, Vol. 2. Ieee, 1150–1157.

[15] Julian 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*. ACM, 785–794.

[16] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 43–52.

[17] Shinya Miura, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2013. SNAPPER: fashion coordinate image retrieval system. In *Signal-Image Technology & Internet-Based Systems (SITIS), 2013 International Conference on*. IEEE, 784–789.

[18] Shaghayegh Sahebi and Peter Brusilovsky. 2013. Cross-domain collaborative recommendation in a cold-start context: The impact of user profile size on the quality of recommendation. In *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 289–295.

[19] Edward Shen, Henry Lieberman, and Francis Lam. 2007. What am I gonna wear?: scenario-oriented recommendation. In *Proceedings of the 12th international conference on Intelligent user interfaces*. ACM, 365–368.

[20] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009), 4.

[21] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. 2008. Matrix Factorization and Neighbor Based Algorithms for the Netflix Prize Problem. In *Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys '08)*. ACM, New York, NY, USA, 267–274. DOI: https://doi.org/10.1145/1454008.1454049

[22] Min Xie, Laks VS Lakshmanan, and Peter T Wood. 2010. Breaking out of the box of recommendations: from items to packages. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 151–158.

[23] Min Xie, Laks VS Lakshmanan, and Peter T Wood. 2011. Comprec-trip: A composite recommendation system for travel planning. In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*. IEEE, 1352–1355.

[24] Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. 2011. Functional matrix factorizations for cold-start recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 315–324.

[25] Tao Zhu, Patrick Harrington, Junjun Li, and Lei Tang. 2014. Bundle recommendation in ecommerce. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 657–666.