ABCPRec: Adaptively Bridging Consumer and Producer Roles for User-Generated Content Recommendation

Kosetsu Tsukuda, Satoru Fukayama, Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{k.tsukuda,s.fukayama,m.goto}@aist.go.jp

ABSTRACT

In Web services dealing with user-generated content (UGC), a user can have two roles: a role of a consumer and that of a producer. Since most item recommendation models have only considered the role of a user as a consumer, how to leverage the two roles to improve UGC recommendation accuracy has been underexplored. In this paper, based on the state-of-the-art UGC recommendation method called CPRec (consumer and producer based recommendation), we propose ABCPRec (adaptively bridging CPRec). Unlike CPRec, which assumes that the two roles of a user are always related to each other, ABCPRec adaptively bridges the two roles according to the similarity between her nature as a consumer and that as a producer. This enables the model to learn each user's characteristics as both a consumer and a producer and to recommend items to each user more accurately. By using two real-world datasets, we showed that our proposed method significantly outperformed comparative methods in terms of AUC.

1 INTRODUCTION

User-generated content (UGC) is content created or produced by ordinary people rather than by professionals and distributed on the Web. Various types of UGC (videos, photos, discussion forums, etc.) exist on various Web services such as YouTube, Flickr, and Reddit. One characteristic of UGC is its fast content production rate compared to that of non-UGC [3]. Hence, item recommendation in UGC services is critical for helping users find their desired content in a large amount of available content. Another characteristic is the role of a user in UGC services. Unlike with traditional non-UGC services, a user can have two roles: a consumer and a producer. Users not only consume content but are also willing to produce content because, for example, they want to show their content to other users by sharing it on UGC services [10].

In consideration of such characteristics, Kang and McAuley [6] have recently proposed a UGC recommendation method called CPRec (consumer and producer based recommendation), which uses matrix factorization. In CPRec, each user has two vectors corresponding to two roles: consumer and producer. These vectors are created by linearly converting the user's core vector, which represents a kind of personality. The preference of the user toward an item is computed based on the affinity between her and the item and the affinity between her and the producer of the item (see Section 2.2 for more details). Using UGC datasets, it has been shown that CPRec outperforms state-of-the-art methods [6].

Despite the effectiveness of CPRec, the user's consumer vector and producer vector are *always bridged* to the user's core vector; this decreases model's flexibility. For example, in CPRec, users who have same consumer vectors also have same producer vectors. However, in reality, even if two users are similar as consumers, they are not always similar as producers. More details of the problem will be described in Section 2.3.

In this paper, to overcome the limitations of CPRec, we propose ABCPRec (adaptively bridging CPRec). In ABCPRec, the user also has two vectors: a vector for the consumer role and that for the producer role. To make the model flexible, these vectors are not created from the core vector (*i.e.*, the user does not have a core vector). Instead, we put the following constraint between two vectors: if the user's nature as a consumer is similar to that as a producer, the two vectors should be close. The more similar these natures are, the closer the vectors should be. If the user's nature as a consumer is completely different from that as a producer, the two vectors can have values independently of each other. That is, unlike CPRec, which always bridges the user's two vectors via her core vector, in ABCPRec, the two vectors are *adaptively bridged* according to the similarities between the consumer and the producer roles.

We evaluated ABCPRec on two publicly available datasets. The experimental results showed that ABCPRec could improve UGC recommendation performance compared to the state-of-the-art methods. More specifically, we showed that it was effective (1) to increase model's flexibility by eliminating the core vector from CPRec and (2) to adaptively add a constraint according to the similarity between the user's consumer and producer roles, which is computed based on the items produced and consumed by the user.

2 MODEL

In this section, we first introduce CPRec [6], which is the state-of-the-art method for UGC recommendations. We then describe the limitations of CPRec and propose our model to overcome them.

2.1 Problem Description and Notation

Let $\mathcal{U}$ and $\mathcal{I}$ denote sets of users and items, respectively. $\mathcal{I}_u^+$ represents a set of items consumed by user $u \in \mathcal{U}$. In addition, let $C \subseteq \mathcal{U}$ denote a set of consumers who consumed at least one item (*i.e.*, $C = \{u \mid u \in \mathcal{U} \wedge |\mathcal{I}_u^+| > 0\}$). In UGC services, all items are created by users; therefore, a set of producers who have produced at least one item can be defined as $\mathcal{P} \subseteq \mathcal{U}$. Using the above data, our goal is to generate a personalized ranked list of items for each user u from $\mathcal{I} \setminus \mathcal{I}_u^+$, which is a set of items that u has not consumed.

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07...$15.00
https://doi.org/10.1145/3331184.3331335

2.2 CPRec

To achieve this goal, Kang and McAuley [6] proposed a method called CPRec. In CPRec, each user has a K-dimensional core latent vector $\boldsymbol{\gamma_u}$. Each user also has latent vectors $\boldsymbol{\gamma_u^c}$ and $\boldsymbol{\gamma_u^p}$ that represent u's roles as a consumer and a producer, respectively. These vectors are created based on $\boldsymbol{\gamma_u}$ as follows:

$$\boldsymbol{\gamma_u^c} = W^c \boldsymbol{\gamma_u}, \quad \boldsymbol{\gamma_u^p} = W^p \boldsymbol{\gamma_u}, \tag{1}$$

where W^c and W^p are $K \times K$ matrices; they are used to project u's core vector to $\boldsymbol{\gamma_u^c}$ and $\boldsymbol{\gamma_u^p}$. Note that both W^c and W^p are shared among all users. The preference of user u toward item i is then computed as follows.

$$\widehat{x}_{ui} = \alpha + \beta_u + \beta_i + \langle \boldsymbol{\gamma_u^c}, \boldsymbol{\gamma_i} \rangle + \langle \boldsymbol{\gamma_u^c}, \boldsymbol{\gamma_{p_i}^p} \rangle, \tag{2}$$

where α is the global offset, β_u and β_i are the user/item bias terms, $\boldsymbol{\gamma_i}$ is a K-dimensional latent vector of i, and p_i is a producer of i.

Unlike most of the non-UGC recommendation models that consider users' role as consumers only [2], CPRec predicts the preference based on the affinity between the consumer and the producer of the item ($\langle \boldsymbol{\gamma_u^c}, \boldsymbol{\gamma_{p_i}^p} \rangle$) in addition to the affinity between the consumer and the item ($\langle \boldsymbol{\gamma_u^c}, \boldsymbol{\gamma_i} \rangle$). The parameters are learned based on Bayesian Personalized Ranking (BPR) [9], which is a pairwise ranking optimization framework and is designed to deal with users' implicit consumption behaviors such as giving a "like" and posting a comment rather than explicit ones such as rating.

2.3 ABCPRec

In CPRec, due to the assumption that each user has a core vector, $\boldsymbol{\gamma_u^c}$ and $\boldsymbol{\gamma_u^p}$ are *always bridged* to $\boldsymbol{\gamma_u}$ via W^c and W^p; this leads to drawbacks in terms of flexibility. In their model, as the values of W^c and W^p are determined through parameter learning, both matrices have their inverse matrices with overwhelming probability. This means that $\boldsymbol{\gamma_u^c}$ almost always converts linearly to $\boldsymbol{\gamma_u^p}$ (*i.e.*, $\boldsymbol{\gamma_u^p} = W^p (W^c)^{-1} \boldsymbol{\gamma_u^c}$). In addition, because W^c and W^p are shared among all users, those users who have the same consumer vectors also have the same producer vectors. However, it is less reasonable because, in reality, even if two users are similar as consumers, they are not always similar as producers. The same can be said when they have the same producer vectors. Because of such limitations, vectors may not have appropriate values through parameter learning. To overcome such drawbacks, we propose a new preference prediction method called ABCPRec (adaptively bridging CPRec).

2.3.1 Score Prediction. In ABCPRec, user u also has K-dimensional latent vectors $\boldsymbol{\nu_u^c}$ and $\boldsymbol{\nu_u^p}$ that correspond to u's two roles (consumer and producer). Unlike CPRec, u does not have a core vector, and there is no linear relationship between $\boldsymbol{\nu_u^c}$ and $\boldsymbol{\nu_u^p}$; this guarantees flexibility of the model. The preference score $\widehat{x}_{ui}$ is computed as follows:

$$\widehat{x}_{ui} = \alpha + \beta_u + \beta_i + \langle \boldsymbol{\nu_u^c}, \boldsymbol{\gamma_i} \rangle + \langle \boldsymbol{\nu_u^c}, \boldsymbol{\nu_{p_i}^p} \rangle, \tag{3}$$

which is almost identical to Eq. 2 except for the vector representation.

2.3.2 Prameter Learning with Adaptive Bridging. We adopt a BPR framework [9] to learn parameters. In BPR, the training set $\mathcal{D}$ used for optimizing parameters is defined as follows:

$$\mathcal{D} = \{(u, i, j) \mid u \in \mathcal{U} \land i \in \mathcal{I}_u^+ \land j \in \mathcal{I} \setminus \mathcal{I}_u^+\}. \tag{4}$$

That is, a triad (u, i, j) means that user u prefers item i to item j. In BPR, the optimization criterion for $\mathcal{D}$ is given by:

$$\sum_{(u,i,j) \in \mathcal{D}} \ln \sigma(\widehat{x}_{uij}) - \lambda_\Theta \|\Theta\|^2, \tag{5}$$

where σ is the sigmoid function, $\Theta = \{\beta_i, \boldsymbol{\gamma_i}, \boldsymbol{\nu_u^c}, \boldsymbol{\nu_u^p}\}$ represents all model parameters, and λ_Θ is a regularization hyperparameter. The $\widehat{x}_{uij}$ represents the difference between u's preference for i and that for j, which is defined as $\widehat{x}_{uij} = \widehat{x}_{ui} - \widehat{x}_{uj}$. In Eq. 5, there is no constraint between $\boldsymbol{\nu_u^c}$ and $\boldsymbol{\nu_u^p}$. In general, with the increase of the model complexity (*i.e.*, the number of parameters), the risk of model over-fitting also increases. Hence, it may not be effective enough to increase model's complexity just by eliminating core vectors from CPRec. In order for our model to learn parameters more appropriately, we add more realistic constraints. To this end, we assume that if u's nature as a consumer is similar to that as a producer, $\boldsymbol{\nu_u^c}$ and $\boldsymbol{\nu_u^p}$ should also be close. In other words, the more similar u's two natures are, the closer $\boldsymbol{\nu_u^c}$ and $\boldsymbol{\nu_u^p}$ should be. That is, a constraint should be adaptively added between $\boldsymbol{\nu_u^c}$ and $\boldsymbol{\nu_u^p}$ according to the similarity. Based on this idea, we revise Eq. 5 and propose a new optimization criterion as follows:

$$\sum_{(u,i,j) \in \mathcal{D}} \ln \sigma(\widehat{x}_{uij}) - \lambda_\Theta \|\Theta\|^2 - \lambda_s \sum_{u \in \mathcal{U}} sim(u) \|\boldsymbol{\nu_u^c} - \boldsymbol{\nu_u^p}\|^2, \tag{6}$$

where λ_s is a regularization hyperparameter and $sim(u)$ is the similarity function. A large value of $sim(u)$ indicates that the distance between $\boldsymbol{\nu_u^c}$ and $\boldsymbol{\nu_u^p}$ should be smaller; this guarantees the adaptivity of the model.

2.3.3 Similarity Function. To compute $sim(u)$, we propose the following two hypotheses:

Hypothesis 1 (H1): if users who like items consumed by u also consume items produced by u, $\boldsymbol{\nu_u^c}$ and $\boldsymbol{\nu_u^p}$ should be similar.

Hypothesis 2 (H2): if users who like items produced by u also consume items consumed by u, $\boldsymbol{\nu_u^c}$ and $\boldsymbol{\nu_u^p}$ should be similar.

The first hypothesis focuses more on u's consumed items while the second one focuses more on u's produced items. Let $\mathcal{U}_u^c$ and $\mathcal{U}_u^p$ denote the users who consumed at least one item consumed and produced by u, respectively. We assume $\mathcal{U}_u^c$ represents the set of users who like items consumed by u and $\mathcal{U}_u^p$ represents the set of users who like items produced by u. Hence, the similarity function reflecting H1 is given by:

$$sim(u) = |\mathcal{U}_u^c \cap \mathcal{U}_u^p| / |\mathcal{U}_u^c|, \tag{7}$$

while the similarity function based on H2 is given by:

$$sim(u) = |\mathcal{U}_u^c \cap \mathcal{U}_u^p| / |\mathcal{U}_u^p|. \tag{8}$$

3 EXPERIMENTS

In this section, we answer the following three research questions through our experiments:

RQ1 To improve the model, is it appropriate to eliminate users' core vectors in CPRec that cause the loss of model flexibility?

RQ2 Is it effective to adaptively add constraints between $\boldsymbol{\nu_u^c}$ and $\boldsymbol{\nu_u^p}$ according to the similarity between the user's two roles?

RQ3 If RQ2 is true, which hypothesis introduced in Section 2.3.3 is more effective for computing the similarity?

Table 1: Dataset statistics.

	Flickr	Reddit
#users ($\lvert\mathcal{U}\rvert$)	99,060	52,654
#items ($\lvert\mathcal{I}\rvert$)	557,286	336,743
Sum of consumed items ($\sum_u \lvert\mathcal{I}_u^+\rvert$)	11,256,457	1,786,032
Consumer ratio ($\lvert\mathcal{C}\rvert/\lvert\mathcal{U}\rvert$)	99.83%	99.60%
Producer ratio ($\lvert\mathcal{P}\rvert/\lvert\mathcal{U}\rvert$)	40.82%	87.24%
Prosumer ratio ($\lvert\mathcal{C}\cap\mathcal{P}\rvert/\lvert\mathcal{U}\rvert$)	40.65%	86.85%

3.1 Dataset

We used two publicly available datasets of UGC services.

- **Flickr** is a photo sharing service where users can share their personal photos. Users can mark other users' photos as "favorite." We use the dataset released by Cha *et al.* [4], which was crawled in 2006 and 2007. We regard the "favorite" action as consumption. Each photo is produced by a single user.
- **Reddit** is an online community where users can discuss various things using a bulletin board style interface. Users can submit textual content or links to the web content such as news articles. Users can also post comments on submitted content. We use the dataset released by Reddit[1], which consists of all submissions and comments in March 2017. We regard each submission and commenting action as an item and consumption, respectively. Each submission is produced by a single user.

Following Kang and McAuley [6], users who consumed or produced fewer than 10 items and items consumed by fewer than 10 users are discarded. Table 1 shows the dataset statistics. In the table, a "prosumer" means a user who produced and consumed items. As can be seen, although the consumer ratios are almost the same in both datasets, the producer ratio and the prosumer ratio on Flickr are much lower than those on Reddit.

3.2 Comparisons

As is the case with Kang and McAuley [6], we use the following baseline methods[2].

- **PopRec**: this method is not personalized and ranks items according to their popularity.
- **BPR** [9]: this method considers only the consumer-item preference. The preference score is computed by:

$$\widehat{x}_{ui} = \alpha + \beta_u + \beta_i + \langle \boldsymbol{\nu}_u^c, \boldsymbol{\gamma}_i \rangle. \tag{9}$$

- **Vista** [5]: in this method, each user has two latent vectors $\boldsymbol{\nu}_u^c$ and $\boldsymbol{\phi}_u$. The $\widehat{x}_{ui}$ is then given by:

$$\widehat{x}_{ui} = \langle \boldsymbol{\nu}_u^c, \boldsymbol{\gamma}_i \rangle + \langle \boldsymbol{\phi}_u, \boldsymbol{\phi}_{p_i} \rangle. \tag{10}$$

  That is, this method uses different vectors for computing u's affinity with i and u's affinity with p_i.
- **Factorization Machines (FMs)** [8]: this method can deal with interactions between consumers, producers, and items. We use the second-order estimator of FMs as follows:

$$\widehat{x}_{ui} = \alpha + \beta_u + \beta_i + \beta_{p_i} + \langle \boldsymbol{\nu}_u^c, \boldsymbol{\gamma}_i \rangle + \langle \boldsymbol{\nu}_u^c, \boldsymbol{\nu}_{p_i}^p \rangle + \langle \boldsymbol{\gamma}_i, \boldsymbol{\nu}_{p_i}^p \rangle. \tag{11}$$

Note that in these baselines, there is no constraint between vectors. We also use the following comparative methods.

- **CPRec** [6]: this is the state-of-the-art method for UGC recommendations. The details can be seen in Section 2.2.
- **NBCPRec**: this method does not put a constraint between $\boldsymbol{\nu}_u^c$ and $\boldsymbol{\nu}_u^p$. That is, the preference score is computed by Eq. 3 and the optimization criterion is given by Eq. 5. NBCPRec means no-bridging CPRec.

Regarding our proposed model, models computing similarity with Eq. 7 and 8 are called **ABCPRec$^{\mathbf{H1}}$** and **ABCPRec$^{\mathbf{H2}}$**, respectively.

3.3 Evaluation Methodology

For each user, we split $\mathcal{I}_u^+$ into training/validation/test sets. To this end, we randomly select one consumed item (*i.e.*, $i \in \mathcal{I}_u^+$) for validation $\mathcal{V}_u$ and another for testing $\mathcal{T}_u$. All the remaining items are used for training $\mathcal{R}_u$. We carry out this random selection independently five times and report the average results in Section 3.4. For a fair comparison, the same five training/validation/test sets are used for all methods. The recommendation performance is evaluated by the AUC (Area Under the ROC Curve):

$$AUC = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{D}_u|} \sum_{(i,j) \in \mathcal{D}_u} \delta\left(\widehat{x}_{ui} > \widehat{x}_{uj}\right), \tag{12}$$

where $\mathcal{D}_u = \{(i,j) \mid i \in \mathcal{T}_u \wedge j \in \mathcal{I} \setminus \mathcal{I}_u^+\}$, and $\delta(a)$ is 1 when a is true and 0 otherwise. For all methods, we use Tensorflow [1] with Adam optimizer [7] to learn model parameters. The regularization hyperparameters are tuned on a validation set in terms of the AUC, where the hyperparameters are selected from {0.0001, 0.001, 0.01, 0.1, 1}. We set the learning rate to 0.01.

3.4 Experimental Results

Table 2 shows the comparative results for latent dimensionality K = 20, 50, and 80 with the results of Tukey's HSD test. It can be observed that even NBCPRec, which has no constraint between $\boldsymbol{\nu}_u^c$ and $\boldsymbol{\nu}_u^p$, significantly outperforms CPRec in both datasets. According to these results, the answer to RQ1 is that eliminating core vectors from CPRec is certainly effective for improving UGC recommendation performance. We can also see that ABCPRec$^{\text{H1}}$ and ABCPRec$^{\text{H2}}$ demonstrated better performance than CPRec and NBCPRec. These results confirm the effectiveness of adaptively bridging $\boldsymbol{\nu}_u^c$ and $\boldsymbol{\nu}_u^p$ rather than always bridging roles (CPRec) and not bridging roles at all (NBCPRec); this is the answer to our RQ2.

Because ABCPRec$^{\text{H2}}$ outperforms ABCPRec$^{\text{H1}}$ in both datasets at all Ks, the following answer to our RQ3 can be given: when we compute similarities between the user's nature as a consumer and that as a producer, H2 (which focuses more on the user's produced items) is more effective. To analyze why H2 is better than H1, we show in Table 3 the distribution of similarity scores computed by each hypothesis. In both datasets, the ratio of users who have no constraints between $\boldsymbol{\nu}_u^c$ and $\boldsymbol{\nu}_u^p$ (*i.e.*, $sim(u) = 0$) is almost the same in both hypotheses. However, in H1, most of the remaining users have $0 < sim(u) \leq 0.1$; while in H2, they are moderately dispersed in $0 < sim(u) \leq 1.0$ on Flickr, and most of them have $0.9 < sim(u) \leq 1.0$ on Reddit. Hence, stronger constraints between $\boldsymbol{\nu}_u^c$ and $\boldsymbol{\nu}_u^p$ were appropriately added in H2, which led to better performance. We also analyze similarities in users' tastes for items in $\mathcal{U}_u^c$ and $\mathcal{U}_u^p$. To this end, for arbitrary user pairs in $\mathcal{U}_u^c$, we count the number of items consumed by both users. Higher values indicate that both users have similar tastes for the consumed

[1]https://www.reddit.com/comments/6607j2

[2] As mentioned by Kang and McAuley [6], since social network of users is not always available on UGC services, we also do not use socially-aware methods such as SBPR [11] for comparison. Rather, in our experiments, we focus more on how to leverage users' two roles to improve UGC recommendations.

Table 2: Performance comparison in terms of the AUC for latent dimensionality K = 20, 50, and 80. Significant differences (α = 0.01) with PopRec, BPR, Vista, FMs, CPRec, NBCPRec, and ABCPRec^{H1} are indicated by †, ‡, ∗, ⋆, ♣, ◇, and ♠, respectively.

Dataset	K	PopRec	BPR	Vista	FMs	CPRec	NBCPRec	ABCPRec^{H1}	ABCPRec^{H2}
Flickr	20	0.6737	0.8698	0.8436	0.8764	0.8563	0.8839 †‡∗⋆♣	0.8861 †‡∗⋆♣	**0.8900** †‡∗⋆♣◇
	50	0.6737	0.8772	0.8435	0.8822	0.8664	0.8937 †‡∗⋆♣	0.8949 †‡∗⋆♣	**0.8992** †‡∗⋆♣◇♠
	80	0.6737	0.8777	0.8394	0.8810	0.8712	0.8955 †‡∗⋆♣	0.8988 †‡∗⋆♣	**0.9028** †‡∗⋆♣◇
Reddit	20	0.6392	0.8713	0.8829	0.8960	0.9138	0.9209 †‡∗⋆♣	0.9296 †‡∗⋆♣◇	**0.9340** †‡∗⋆♣◇♠
	50	0.6392	0.8721	0.8918	0.8999	0.9201	0.9302 †‡∗⋆♣	0.9346 †‡∗⋆♣◇	**0.9391** †‡∗⋆♣◇♠
	80	0.6392	0.8709	0.8946	0.9001	0.9211	0.9322 †‡∗⋆♣	0.9376 †‡∗⋆♣◇	**0.9408** †‡∗⋆♣◇♠

Table 3: Distribution of $sim(u)$.

	H1		H2	
	Flickr	Reddit	Flickr	Reddit
$sim(u) = 0$	62.70%	13.77%	62.87%	14.17%
$0 < sim(u) \leq 0.1$	36.46%	47.34%	2.93%	0.82%
$0.1 < sim(u) \leq 0.2$	0.54%	10.68%	4.92%	1.79%
$0.2 < sim(u) \leq 0.3$	0.08%	6.54%	5.25%	2.67%
$0.3 < sim(u) \leq 0.4$	0.02%	4.39%	4.78%	2.97%
$0.4 < sim(u) \leq 0.5$	0.01%	2.93%	4.53%	2.76%
$0.5 < sim(u) \leq 0.6$	0.0%	2.93%	4.71%	5.56%
$0.6 < sim(u) \leq 0.7$	0.0%	2.2%	4.07%	6.21%
$0.7 < sim(u) \leq 0.8$	0.0%	1.63%	3.21%	6.79%
$0.8 < sim(u) \leq 0.9$	0.0%	1.33%	2.02%	9.68%
$0.9 < sim(u) \leq 1.0$	0.17%	6.25%	0.72%	46.59%

items. We did this process for all users' $\mathcal{U}_u^c$. The same goes for $\mathcal{U}_u^p$. Fig. 1 shows the distribution of the user pair ratio on Reddit. In $\mathcal{U}_u^c$, almost half of the user pairs do not consume the same items at all. In $\mathcal{U}_u^p$, the ratios of user pairs with ≥ 2 (which means that both users have relatively high similarity in taste) are always higher than those in $\mathcal{U}_u^c$, and as many as 14.06% of the user pairs consumed more than 10 of the same items. A similar tendency was found on Flickr. These results indicate that users in $\mathcal{U}_u^c$ are more diverse in terms of item tastes while users in $\mathcal{U}_u^p$ have similar tastes. That is why H2, which focuses more on $\mathcal{U}_u^p$, was able to add more reliable constraints and outperform H1.

Finally, Fig. 2 plots the transition of the AUC for all methods except PopRec. It can be observed that ABCPRec^{H2} achieved the best performance for any K in both datasets. Its performance is almost saturated when K is 80 on Flickr and 60 on Reddit. FMs is similar to NBCPRec in that FMs also considers the affinity between the consumer and the item and the affinity between the consumer and the item's producer without bridging $\boldsymbol{v}_u^c$ and $\boldsymbol{v}_u^p$. However, NBCPRec always outperforms FMs. This result indicates that it is not effective to consider the affinity between an item and its producer ($\langle \gamma_i, \boldsymbol{v}_{p_i}^p \rangle$ in Eq. 11) for UGC recommendations. As mentioned in Section 3.1, the producer ratio and prosumer ratio on Flickr are relatively low. In such a dataset, the performance of CPRec is below the baselines (BPR and FMs). On the other hand, our model works well in both datasets; this shows the versatility of our model.

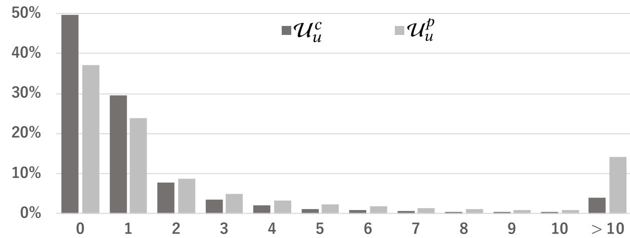

Figure 1: Distribution of user pair ratio in Reddit. (x-axis: the number of items consumed by both users in a user pair)

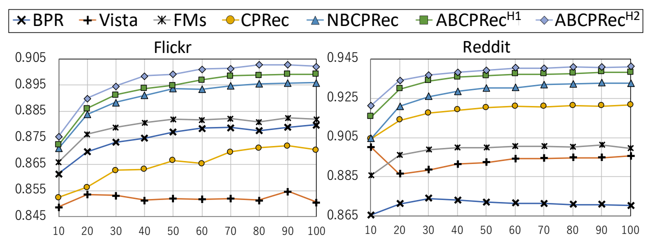

Figure 2: Transition of the AUC associated with latent dimensionality K. (x-axis: K, y-axis: AUC)

4 CONCLUSION

This paper proposed a method that considers two user roles and adaptively bridges them according to the similarity between the user's nature as a consumer and that as a producer. The experimental results showed the effectiveness of our method, especially when the user's similarity was computed with a focus on users who like items produced by the user. As future work, using visual features (*e.g.*, colors) and textual features (*e.g.*, tags) of items produced and consumed by the same user is another possible approach to compute similarity between the user's two roles. We also plan to extend our method by adding constraints, for example, on the similarity between two users in terms of their producer roles.

[Acknowledgments] This work was supported by JSPS KAKENHI Grant Number 17K12688 and JST ACCEL Grant Number JPMJAC1602, Japan.

REFERENCES

[1] M. Abadi *et al.* Tensorflow: A system for large-scale machine learning. In *OSDI*, pages 265–283, 2016.
[2] J. Bobadilla *et al.* Recommender systems survey. *Knowledge-Based Systems*, 46:109–132, 2013.
[3] M. Cha *et al.* I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *IMC*, pages 1–14, 2007.
[4] M. Cha *et al.* A measurement-driven analysis of information propagation in the flickr social network. In *WWW*, pages 721–730, 2009.
[5] R. He *et al.* Vista: A visually, socially, and temporally-aware model for artistic recommendation. In *RecSys*, pages 309–316, 2016.
[6] W. C. Kang and J. McAuley. Learning consumer and producer embeddings for user-generated content recommendation. In *RecSys*, pages 407–411, 2018.
[7] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
[8] S. Rendle. Factorization machines. In *ICDM*, pages 995–1000, 2010.
[9] S. Rendle *et al.* BPR: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.
[10] A. Wilson *et al.* Hospitality and travel: The nature and implications of user-generated content. *Cornell Hospitality Quarterly*, 53(3):220–228, 2012.
[11] T. Zhao *et al.* Leveraging social connections to improve personalized ranking for collaborative filtering. In *CIKM*, pages 261–270, 2014.