Bayesian Matrix Factorization

Subham Kumar
160707

Sudhanshu Bansal
160710

Suryateja BV
160729

1 Motivation

Cheap and widespread internet access has resulted in a huge burst in online activity. Consumer over-choice is a pressing problem of todays online world. This has rendered Recommender Systems an ubiquitous part of various e-commerce and media websites. Recommender Systems (or RecSys) make a users life easy by assisting in the decision making process of what item to choose. Some of the worlds best tech companies like Netflix, YouTube are solely reliant on how well their Recommender Systems work. Recommender Systems are one of the most widely researched areas in AI. While state-of-the-art systems continue to outperform their previous iterations, some problems continue to exist such as - the cold start problem, lack of diversity in recommendations, explainability, robustness of recommendations etc. Recent events of Brexit and the US Presidential Elections have sparked debates on whether Recommender Systems are leading to formation of filter bubbles. Thus, fairness in these systems also presents itself as a crucial area of research. Our focus in this project was to approach these problems in a Bayesian setting - we wanted to ascertain uncertainty in our predictions and make them more explainable. To that end, we studied and implemented papers that work with a generative story in place. To tackle inductive bias, we followed Liang et al's (3) method of modeling user exposure to items as latent variables. We incorporated a regularizer term in the rudimentary matrix factorization model following Liang et al's (4) item co-occurrence matrix approach. We also studied a more recent idea of bringing about fairness in these systems via constrained bandits problem (1). Finally, we unified the ideas in these papers to build our own recommendation model.

2 Previous Work

2.1 Probabilistic Matrix Factorization

Matrix factorization models for inferring user preferences are ubiquitous in today's systems. Probabilistic matrix factorization technique proposed by Salakhutdinov and Mnih (5) in 2008 outperformed the other models in the Netflix Prize competition. From a generative story perspective, the model is as follows - we generate user latent factors and item latent factors, and use their dot product to parametrize the Gaussian distribution of user-item matrix entries.

$$\boldsymbol{\theta}_u \sim \mathcal{N}\left(\boldsymbol{\theta}_u | 0, \lambda_\theta \mathbf{I}_K^{-1}\right)$$
$$\boldsymbol{\beta}_i \sim \mathcal{N}\left(\boldsymbol{\beta}_i | 0, \lambda_\beta \mathbf{I}_K^{-1}\right)$$
$$y_{ui} \sim \mathcal{N}\left(y_{ui} | \boldsymbol{\theta}_u^T \boldsymbol{\beta}_i, \sigma_y^2\right)$$

While the model works fine for explicit data, it suffers some serious setbacks when dealing with implicit data. Implicit data, like clicks, are usually binary decision entries in the user-item matrix. A *zero* in the matrix could represent either user's dislike for the item or a lack of exposure to the item. This error is induced by the assumptions of our model, and hence is termed *Inductive bias.*

2.2 Weighted Matrix Factorization

Also known as one-class collaborative filtering, WMF model proposed by Hu et al (2) tries to overcome inductive bias by setting up 2 hyperparameters, $c_{y=1}$ and $c_{y=0}$. The idea is to choose a small $c_{y=0}$ so that implicit data entries with a value of zero (no click) are downweighted.

$$\boldsymbol{\theta}_u \sim \mathcal{N}\left(\boldsymbol{\theta}_u | 0, \lambda_\theta \mathbf{I}_K^{-1}\right)$$
$$\boldsymbol{\beta}_i \sim \mathcal{N}\left(\boldsymbol{\beta}_i | 0, \lambda_\beta \mathbf{I}_K^{-1}\right)$$
$$y_{ui} \sim \mathcal{N}\left(y_{ui} | \boldsymbol{\theta}_u^T \boldsymbol{\beta}_i, {c_{ui}}^{-1}\right) \qquad c_{y=1} > c_{y=0}$$

The problem with this approach is the lack of a fully generative story. We need to set the hyperparameters either via grid search, random search or other heuristics, but we can't fully ascertain the uncertainty in the model.

We studied and implemented some recent papers that focus on tackling the shortcomings of the above two models. The report in subsequent sections is organized as follows - Section 3 describes the model exposure method in detail, and provides a lazy algorithm for quick inference. Section 4 deals with the addition of a regularizer term to the WMF loss function, which is nothing but a non-linear transformation of user-item matrix. We also discuss some intuitive ideas behind how the addition of regularizer imparts diversity in recommendations. In Section 5, we present a new model which combines both regularizer and user exposure latent variable approaches. Finally, Section 6 deals with imparting fairness in recommendations via a constrained bandit approach.

3 Modelling User Exposure

We often have plenty of data in implicit form. For instance how many times a user listened to a song. While modelling such implicit data for recommendation(implicit feedback model) we often have inductive bias mostly due to the absence of information in terms of exposure.The paper (4) proposes a bayesian model to distinguish between whether a user has decided to not to click on item i or hasn't be ever exposed to item i.

3.1 Model

We consider the setting where we have a binary click matrix $\mathbf{Y} = \{y_{ui}\}$ indicating whether user clicked on item i and a binary exposure matrix $\mathbf{A} = \{a_{ui}\}$ to indicate whether user u is exposed to item i. Both $\mathbf{Y}, \mathbf{A}$ are $U \times I$ dimensional. Note that this exposure matrix is generated using additional informations like for venue recommendation it might be location of venues.
The full **generative story** is as follows :

$$\boldsymbol{\theta}_u \sim \mathcal{N}\left(\boldsymbol{\theta}_u | 0, \lambda_\theta \mathbf{I}_K^{-1}\right)$$
$$\boldsymbol{\beta}_i \sim \mathcal{N}\left(\boldsymbol{\beta}_i | 0, \lambda_\beta \mathbf{I}_K^{-1}\right)$$
$$a_{ui} \sim \text{Bernoulli}\left(\mu_{ui}\right)$$
$$y_{ui} | a_{ui} = 1 \sim \mathcal{N}\left(y_{ui} | \boldsymbol{\theta}_u^T \boldsymbol{\beta}_i, \lambda_y^{-1}\right)$$
$$y_{ui} | a_{ui} = 0 \sim \delta_0$$

Here δ_0 is 1 if y_{ui} is zero simply implying that if the user has not been exosed to item i then the probability of clicking on the item i is 0. Also if $y_{ui} > 0$ then we take $a_{ui} = 1$.Only when $y_{ui} = 0$ then a_{ui} is latent implying we don't know why user u didn't click on the item,whether the reason was not exposed to item i or he didn't like it at all. To mention μ_{ui} is the prior belief on exposure for item i to the user u. We learn this μ_{ui} in two ways:

- Based on **item popularity**: The basic assumption in this method is that an item which is popular, people are more likely to be exposed to it.Note that in this case μ_{ui} will be same for all the users w.r.t. to item i. The overall model is shown in Figure 1(a).
- Using **exposure covariates**: In this method we assume that we have some additional information about items. For instance,while recommending venues we have location of venues or say for recommending documents

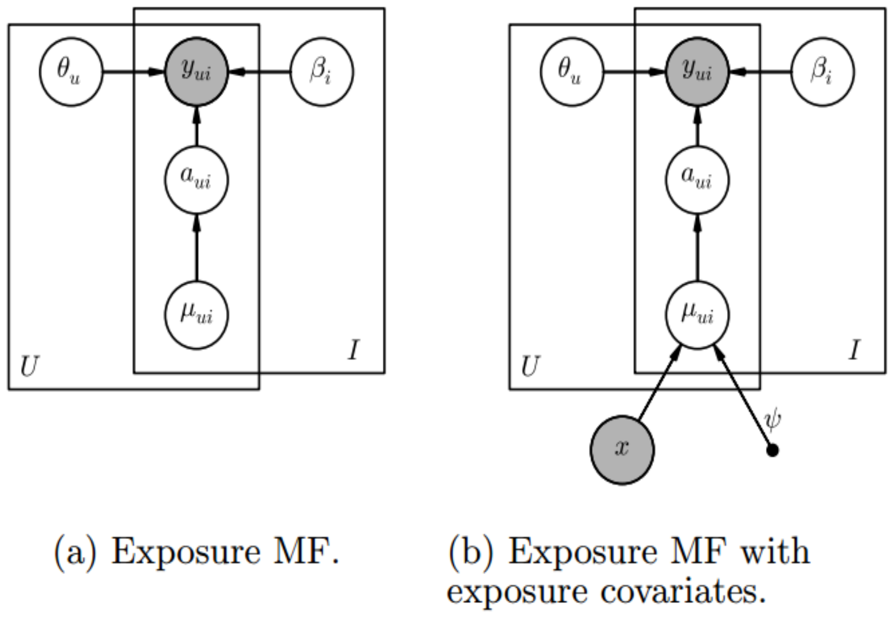

Figure 1: Plate notations of the model with and without exposure covariates (4)

we have words from the documents. We use a general representation $\mathbf{x}_i$ of each item to be recommended like for documents we use the topic scores of each topic in a document (similar to LDA). Then we take $\mu_{ui} = \sigma(\boldsymbol{\psi}_u^T \mathbf{x}_i)$. Here we learn $\boldsymbol{\psi}_i$ while doing for inference for other variables. An intuitive explanation of $\boldsymbol{\psi}_u$ could be if user u is interested in topic recommender system and x_{il} denotes this topic then we want ψ_{ul} to be high.The overall model is shown in Figure 1(b).

The log-joint of exposure and click for user u and item i can be written as:

$$\log\left(a_{ui}, y_{ui} | \mu_{ui}, \boldsymbol{\theta}_u, \boldsymbol{\beta}_i, \lambda_y^{-1}\right) = \log Bern(a_{ui}|\mu_{ui}) + a_{ui} \log \mathcal{N}(y_{ui}|\boldsymbol{\theta}_u^T \boldsymbol{\beta}_i, \lambda_y^{-1}) + (1 - a_{ui}) \log \mathbb{I}[y_{ui} = 0]$$

A few things to note here. First if a_{ui} is 0 then the model mimics the well known Probabilistic Matrix Factorization (5). Also the WMF (2) is a special case of it (can be noted when $a_{ui} = 0$).

3.2 Inference

For inference the paper uses Expectation-Maximization. All the updates are in closed form due to conditional conjugacy except when we use exposure covariates.The details are below.

E-step

In this step we calculate the expectation of a_{ui} w.r.t. its posterior. Recall that we only need to calculate the expectations for the case when $y_{ui} = 0$ as only in this case a_{ui} behaves as latent variable.
Mathematically:

$$\mathbb{E}\left[a_{ui} | \boldsymbol{\theta}_u, \boldsymbol{\beta}_i, \mu_{ui}, y_{ui} = 0\right] = \gamma_{ui} = \frac{\mu_{ui} \mathcal{N}(0|\boldsymbol{\theta}_u^T \boldsymbol{\beta}_i, \lambda_y^{-1})}{\mu_{ui} \mathcal{N}(0|\boldsymbol{\theta}_u^T \boldsymbol{\beta}_i, \lambda_y^{-1}) + (1 - \mu_{ui})}$$

For $y_{ui} = 1$, $\gamma_{ui} = 1$ as a_{ui} will be 1 in that case.

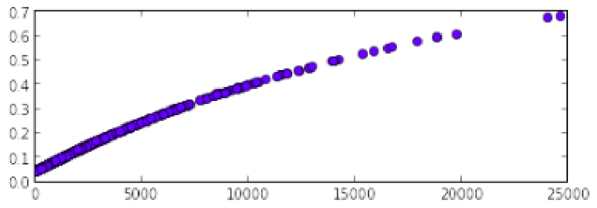

Figure 2: A plot of empirical item popularity(x-axis) vs prior exposure belief(y-axis)

M-step

In this step we update for $\boldsymbol{\beta}_i, \boldsymbol{\theta}_u$ and exposure priors as follows:

$$\boldsymbol{\theta}_u \leftarrow \left(\lambda_y \sum_i \gamma_{ui}\boldsymbol{\beta}_i\boldsymbol{\beta}_i^T + \lambda_\theta \mathbf{I}_K\right)^{-1} \left(\sum_i \lambda_y \gamma_{ui} y_{ui} \boldsymbol{\beta}_i\right) \; \forall \; u \; \epsilon \{1, ..., U\}$$

$$\boldsymbol{\beta}_i \leftarrow \left(\lambda_y \sum_u \gamma_{ui}\boldsymbol{\theta}_u\boldsymbol{\theta}_u^T + \lambda_\beta \mathbf{I}_K\right)^{-1} \left(\sum_u \lambda_y \gamma_{ui} y_{ui} \boldsymbol{\theta}_u\right) \; \forall \; i \; \epsilon \{1, ..., I\}$$

If we are using item popularity based prior exposure belief, the updates are simple as finding the MLE w.r.t. u_i is same as finding the mode of the complete conditional $Beta(\alpha_1 + \sum_u \gamma_{ui}, \alpha_2 + U - \sum_u \gamma_{ui})$.Hence the update is simple.

$$u_i \leftarrow \frac{\alpha_1 + \sum_u \gamma_{ui} - 1}{\alpha_1 + \alpha_2 + U - 2} \; \forall \; i \; \epsilon \{1, ..., I\}$$

If we use exposure covariate the update for $\boldsymbol{\psi}_u$ doesn't turn out to be in closed form, so we use gradient descent based update leading to `Generalized EM algorithm`.We first calculate noisy gradient using a particular batch-size as:

$$\nabla_{\boldsymbol{\psi}_u} \mathcal{L} = \frac{1}{|B_t|} \sum_{i \epsilon B_t} (p_{ui} - \sigma(\boldsymbol{\psi}_u^T \mathbf{x}_i))\mathbf{x}_i$$

and then update $\boldsymbol{\psi}_u$ as:

$$\boldsymbol{\psi}_u^{new} \leftarrow \boldsymbol{\psi}_u + \eta \nabla_{\boldsymbol{\psi}_u} \mathcal{L}$$

3.3 Experiments and Results

Explicitly storing the entire matix $\mathbf{A}$ is highly storage inefficient. He we perform E-step lazily meaning that only the required part of the matrix is constructed for the updates of latent factors(user/item factors) and exposure priors.
Also we tested the model on the `GOWALLA` dataset containing hotel check-ins details along with the venue location.For training with exposure covariate we first did GMM clustering of location with cluster-size=100 and then used it to represent each venue($\mathbf{x}_i$). The results are shown in table below.

	Recall@20	**Recall@50**	**NDCG@100**	**MAP@100**
Without exposure covariate	0.0906	0.1450	0.0917	0.0324
With exposure covariate	0.1288	0.1988	0.1252	0.0477

Note that the value of the measures are high for model using exposure covariate which makes sense since we are incorporating additional information in the model.Also note the plot in Figure 2.There is a strong strong relationship between empirical item popularity and consideration(almost linear). Here we define empirical item popularity as number of users who has clicked a particular item.

4 Item-item Co-occurrence Matrix

This idea of introducing a co-occurrence matrix by Liang, et al (3) is inspired from the word2vec word-word co-occurrence matrix. In this setting, we see the documents as users, and the sequential words that occur in each document as the sequential items that a user picks. The model used is an extension of WMF model discussed before. The richness of the model stems from the fact that the item embeddings $\boldsymbol{\beta}_i$ are jointly learned, as in, they are shared across matrices M and Y. The partial generative story is as follows -

$$\begin{aligned}
\boldsymbol{\theta}_u &\sim \mathcal{N}\left(\boldsymbol{\theta}_u | 0, \lambda_\theta \mathbf{I}_K^{-1}\right) \\
\boldsymbol{\beta}_i &\sim \mathcal{N}\left(\boldsymbol{\beta}_i | 0, \lambda_\beta \mathbf{I}_K^{-1}\right) \\
\boldsymbol{\gamma}_i &\sim \mathcal{N}\left(\boldsymbol{\gamma}_i | 0, \lambda_\gamma \mathbf{I}_K^{-1}\right) \\
y_{ui} &\sim \mathcal{N}\left(y_{ui} | \boldsymbol{\theta}_u^T \boldsymbol{\beta}_i, {c_{ui}}^{-1}\right) \qquad c_{y=1} > c_{y=0} \\
m_{ij} &= \boldsymbol{\beta}_i^T \boldsymbol{\gamma}_j - \mathbf{w}_i - \mathbf{c}_j
\end{aligned}$$

Here, γ_j denotes the context vector. Note that co-occurrence matrix M is not generated. The $(ij)^{th}$ entry of M is directly proportional to the number of users picking both i and j. M is the Shifted Positive Pointwise Mutual Information (SPPMI) matrix of items. M can be seen as a non linear transformation of matrix U. Essentially, we are making the pairwise item occurrence signal more explicit by adding a seperate item exclusively for it.

To compute the $(ij)^{th}$ entry of M, we'll need to know the number of times the word j has occurred in the context of word i. In the document setting, this is easy to obtain - we can follow a skipgram fashion of choosing the nearby words in the sequential order. However, in the case of items and implicit data, we don't usually have access to sequential information. Hence, the context of an item i refers to all the items were picked along with i across all the users.

The inference is straight forward. We get closed form updates for model parameters, which can then be updated in a coordinate ascent procedure. We experimented on Movielens (20M) dataset, with timestamps. To binarize data, all ratings greater than 3 were considered as a click. Training, validation and test sets were prepared from the same data. To evaluate the model, we used Recall@M and NDCG@M as evaluation metrics. Both the metrics work in a similar way - make predictions for the input data, and rank them based on how likely they are to be recommended. Next, look at how many items in the top M ranks were actually picked by that user. Recall@M gives each of the top M ranks an equal weightage. NDCG@M improves upon Recall@M by giving more weightage to the top ranks. Since the dataset was too heavy, and due to limited compute power, we couldn't run this model for more than 5 iterations. We could not achieve results presented in the paper.

Cold-Start: WMF suffers from cold-start for users who have consumed less items since there is not enough data to make inference about item preference. However, co-occurrence matrix delivers good signals of pairwise interaction of items, and learns better latent representations. It makes certain item related signals more explicit which have helped in alleviating the cold-start problem.

Diversity: Another aspect where WMF suffers is diversity. Rare items are ranked middle to low due to their high variance. Essentially, WMF is unsure about what rank to give to rare items. With item embeddings at our disposal, rare items are typically ranked either very high or very low. For example, a person who watches a Taiwanese movie is expected to watch another far-eastern movie. This signal is explicitly captured by the SPPMI matrix. Hence, other rare far-eastern movies are given greater ranks. On the other hand, consider the case of a really bad movie which no one has watched. This item would never occur in the SPPMI matrix and hence is ranked very low.

5 Exposure+Co-occurrence

We've tried to unify the approaches studied and arrived at the following loss function -

$$\mathcal{L} = \sum_{u,i} \lambda_y a_{ui} \left(y_{ui} - \boldsymbol{\theta}_u^T \boldsymbol{\beta}_i\right)^2 - 2a_{ui} \log \mu_{ui} - 2\left(1 - a_{ui}\right)\left(\log \mathbb{I}\left\{y_{ui} = 0\right\} + \log\left(1 - \mu_{ui}\right)\right)$$

$$\sum_{m_{ij} \neq 0} \left(m_{ij} - \boldsymbol{\beta}_i \boldsymbol{\gamma}_j - \mathbf{w}_i - \mathbf{c}_j\right)^2 + \lambda_\theta \sum_u \|\boldsymbol{\theta}_u\|^2 + \lambda_\beta \sum_i \|\boldsymbol{\beta}_i\|^2 + \lambda_\gamma \sum_j \|\boldsymbol{\gamma}_j\|^2$$

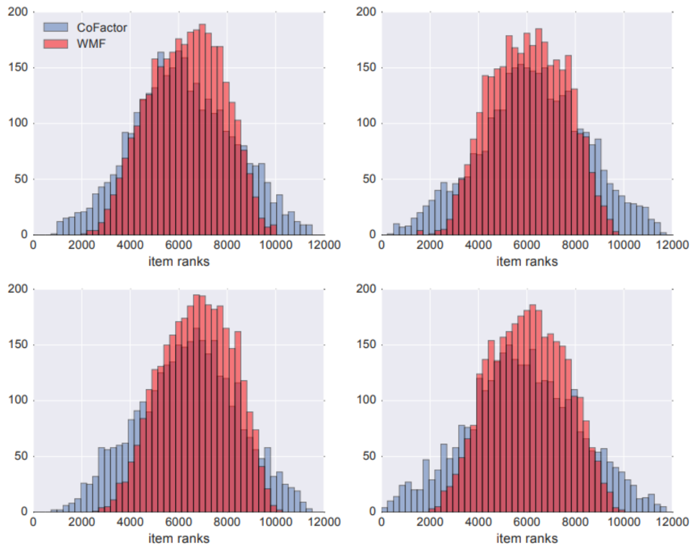

Figure 3: Rare items get either top ranks or bottom ranks with regularizer (blue), whereas WMF (red) ranks them somewhere in the middle (3)

The green terms represent the item embedding related terms as an addition to the exposure model loss function. Deriving inference technique is similar to the exposure model - we followed the generalized EM algorithm, incorporating the additional regularizer term by changing the update equations. We ran the algorithm on MovieLens dataset, however we couldn't derive any satisfactory results. The algorithm took a lot of time to run, and with limited compute power, we couldn't fully explore the hyperparameter space.

6 Constrained Bandits for Fairness

In the multi-armed bandit setting, items are formulated as arms a's, and each arm has an unknown reward r_a. A user has to select an arm a^t for each time step $t = 1, 2, \cdots, T$ such that the Regret (loss) is minimized. More formally, let's say that the rewards come from an unknown distribution $\mathcal{D}$. Based on the previous observations $\left(a^1, r_a^1\right), \cdots, \left(a^{t-1}, r_a^{t-1}\right)$, we draw $a_t \sim p_t$. p_t should be selected such that the regret is minimised. Regret is defined as follows -

$$\text{Regret}_T = \mathbb{E}_{r^t \sim \mathcal{D}} \left[\sum_{t=1}^{T} r_{a^*}^t - \sum_{t=1}^{T} r_a^t \right]$$

where $r_{a^*}^t$ is the maximum expected reward at time t, which is apriori unknown. Bandit algorithms typically follow a exploration vs exploitation procedure; initially we exploit the various arms available and once we find the optimal reward distribution, we continue picking that a^* and hence exploit. In a way, bandit algorithms by definition induce polarization. We tend to pick the same arm again and again after a period of time, which results in a tunnel vision. One possible solution is to cut down the data or hide arms, but this would bring down our algorithm's predictive power. So, the target is to design algorihtms that avoid polarization, yet minimise individual regret.

We studied a novel approach proposed by Celis et al (1) to tackle the issue of polarisation in recommendations. First, we divide the arms into groups $G_1, G_2, \cdots, G_g$ - say, using topic modelling to divide various webpages into g topics. Then, a lower limit l_i and an upper limit u_i are set in a personalized way on the probability masses that the Recommender system puts on each group G_i. Formally,

$$l_i \leq w_a\left(G_i\right) p_a^t \leq u_i \quad \forall i \in [g] \forall t \in [T]$$

wherein $w_a\left(G_i\right)$ denotes the similarity between arm a and group G_i. For example, a right leaning news article a, might have a greater weight when in conservative group G_i, than when in a liberal group G_j. The above approach works well because we are limiting the mass on the probability a particular arm can take at any time. This ensures that the probability on the optimal arm is not too high, hence we explore more, and give out diverse, non-polarized recommendations.

The authors have proposed an algorithm called constrained ϵ-greedy, which is an extension of the standard ϵ-greedy approach for MAB setting. The algorithm manipulates the values taken by ϵ, and ensures there is at least some η mass on each arm, where η is calculated from the group bound specifications.

7 Tools / Software Used

Our code was entirely written in Python, and implemented in Jupyter Notebook. We used the following Machine Learning related libraries - Numpy, Scipy, Pandas, Sklearn. For plotting, we used the standard Matplotlib library along with Seaborn. As mentioned above, the algorithms are pretty time-consuming. In order to parallelize and help in lazy evaluation, we used *joblib* library. We used famous datasets freely available on the Internet to test our algorithms.

8 Future Work

We plan to work further on our proposed algorithm to see if we can satisfactory results by trying on other simple datasets. One direction of work we plan to pursue is to consider a user-user co-occurrence matrix as a regulari zer. We hypothesize that different datasets can be studied better using different regularizers. Also, we plan to implement the constrained epsilon greedy algorithm on MovieLens dataset to see if fairness is actually induced in the recommendations.

References

[1] L. Elisa Celis, Sayash Kapoor, Farnood Salehi, and Nisheeth Vishnoi. Controlling polarization in personalization: An algorithmic framework. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pages 160–169, New York, NY, USA, 2019. ACM.

[2] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 263–272, Washington, DC, USA, 2008. IEEE Computer Society.

[3] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M. Blei. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 59–66, New York, NY, USA, 2016. ACM.

[4] Dawen Liang, Laurent Charlin, James McInerney, and David M. Blei. Modeling user exposure in recommendation. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 951–961, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.

[5] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pages 1257–1264, USA, 2007. Curran Associates Inc.