HOP-Rec: High-Order Proximity for Implicit Recommendation

Jheng-Hong Yang
Academia Sinica
Taipei, Taiwan
jhyang@citi.sinica.edu.tw

Chih-Ming Chen
National Chengchi University
Taipei, Taiwan
104761501@nccu.edu.tw

Chuan-Ju Wang
Academia Sinica
Taipei, Taiwan
cjwang@citi.sinica.edu.tw

Ming-Feng Tsai
National Chengchi University
Taipei, Taiwan
mftsai@nccu.edu.tw

ABSTRACT

Recommender systems are vital ingredients for many e-commerce services. In the literature, two of the most popular approaches are based on factorization and graph-based models; the former approach captures user preferences by factorizing the observed direct interactions between users and items, and the latter extracts indirect preferences from the graphs constructed by user-item interactions. In this paper we present *HOP-Rec*, a unified and efficient method that incorporates the two approaches. The proposed method involves random surfing on a graph to harvest high-order information among neighborhood items for each user. Instead of factorizing a transition matrix, our method introduces a confidence weighting parameter to simulate all high-order information simultaneously, for which we maintain a sparse user-item interaction matrix and enrich the matrix for each user using random walks. Experimental results show that our approach significantly outperforms the state of the art on a range of large-scale real-world datasets.

KEYWORDS

collaborative filtering; top-N recommendation; random walks; bipartite graph; matrix factorization; implicit feedback

ACM Reference Format:
Jheng-Hong Yang, Chih-Ming Chen, Chuan-Ju Wang, and Ming-Feng Tsai. 2018. HOP-Rec: High-Order Proximity for Implicit Recommendation. In *Twelfth ACM Conference on Recommender Systems (RecSys '18), October 2–7, 2018, Vancouver, BC, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3240323.3240381

1 INTRODUCTION

Recommender systems are ubiquitous in modern times and have been applied to services for recommending items such as music, books, and movies. Real-world recommender systems include a number of user-item interactions that facilitate recommendations, including playing times, likes, sharing, and tags.

RecSys '18, October 2–7, 2018, Vancouver, BC, Canada

ACM ISBN 978-1-4503-5901-6/18/10…$15.00
https://doi.org/10.1145/3240323.3240381

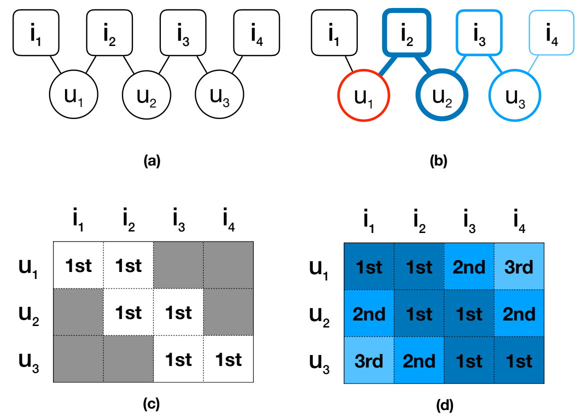

Figure 1: High-order proximity between users and items within observed interactions

Collaborative filtering (CF) is commonly used to leverage this interaction data for recommendation, because it yields reasonable performance among diverse recommendation strategies [15] and because it does not require domain knowledge. There are two mainstream CF models: latent factor models and graph-based models [13]. Latent factor models discover shared latent factors given interactions between users and items by decomposing the user-item matrix; matrix factorization (MF) [8] is the most representative of this type of approaches. Moreover, recent literature has focused more on optimizing item ranks from implicit data than on predicting explicit item scores [14, 17, 19]. Most such methods assume that unobserved items are of less interest to users; thus, these methods are designed to discriminate observed (positive) items from unobserved (negative) items. Graph-based models, in turn, explore the high-order proximity between vertices inherent in a simple user-item bipartite graph constructed by users, items, and their interactions [1, 5, 12, 16]. To some extent, such graph-based methods relax the assumption made by factorization-based models since they explicitly model high-order proximity between users and items in the user-item-interaction bipartite graph.

In this paper, we present *HOP-Rec*, a unified and efficient method[1] that incorporates these two approaches for implicit recommendation. HOP-Rec (high-order proximity augmented recommendation) discovers high-order indirect information of neighborhood items for each user from the user-item bipartite graph by conducting random surfing on the graph. With a confidence parameter for the order of occurrences in a path traveled, our method considers different orders of items simultaneously when decomposing the latent factors of user preferences. Figure 1 illustrates the idea of modeling high-order proximity between users and items within observed interactions. In the figure, panel (a) denotes a bipartite graph constructed from observed user-item interactions; panel (b) demonstrates a path that starts from source node u_1 to target node i_4; panel (c) records the direct interactions between users and items in a matrix form; panel (d) shows the high-order relations between users and items. Specifically, potentially preferred items for user u_1 are ordered by their occurrence in the path in panel (b); for instance, the observed preferred item i_2 and user u_1 have the first-order relation shown in panel (d), and item i_3 is a second-order potential candidate for u_1 since it is in the preferred item list of user u_2, who shares same preferred item i_2 with u_1. Such high-order indirect preference information is explicitly modeled in the proposed HOP-Rec. Experimental results on four large-scale real-world datasets show that HOP-Rec significantly outperforms several state-of-the-art recommendation algorithms and suggest that incorporating high-order proximity with factorization model is promising for general top-N implicit recommendation problems.

2 METHODOLOGY

2.1 Definitions

Here we define the interaction graphs used in our method as well as k-order proximity.

Definition 1. (Interaction Graph) User-item interactions are represented as a $(0,1)$-biadjacency matrix $A = (a_{ij}) \in \mathbb{R}^{|U|\times|I|}$, where U and I denote the sets of users and items, respectively. A bipartite *interaction graph* $G = (V, E)$ is constructed from the biadjacency matrix A, where $V = U \cup I$ and $e_{ij} \in E$ if and only if $a_{ij} = 1$.

Definition 2. (k-order Proximity) Given an interaction graph $G = (U \cup I, E)$, the *k-order proximity* ($k \in \mathbb{N}^+$) between a pair of nodes (u, i), where $u \in U$ and $i \in I$, is defined by the rank of their occurrence and transition probability in a sequence generated by a random walker traveling through the graph. Specifically, let $S_u = (u, i_1, u_1, \ldots, u_{k-1}, i_k, \ldots)$ denote a walk sequence from a source node u, where u_j or i_j denotes the j-th neighbor user or item of node u, respectively.[2] The magnitude of k-order proximity between a source user u and its k-th-neighbor item i_k is measured by $p_u^k(i_k) \times C(k)$, where $p_u^k(i_k)$ is the k-order transition probability from u to i_k and $C(k)$ is a decay factor. If there is no walkable path from u to i_k, the k-order proximity is 0.

[1] https://github.com/cnclabs/proNet-core
[2] Without loss of generality, let $u = u_0$.

2.2 Proposed Model

Having defined the interaction graph and k-order proximity, we formulate the factorization of the interaction matrix with higher-order proximity from an implicit feedback dataset.

Factorization Model. The goal of a factorization model is to estimate the latent factors of the following two sets: $\boldsymbol{\theta}_U, \boldsymbol{\theta}_I \subseteq \Theta$, where $\boldsymbol{\theta}_U \in \mathbb{R}^{|U|\times d}$ for users, $\boldsymbol{\theta}_I \in \mathbb{R}^{|I|\times d}$ for items, d is the dimension of the low-rank latent factor space, and Θ is a superset of $\boldsymbol{\theta}_U$ and $\boldsymbol{\theta}_I$ consisting of all the parameters in our model. Let θ_u (θ_i) denote the row vector for user u (item i, respectively) from $\boldsymbol{\theta}_U$ ($\boldsymbol{\theta}_I$, respectively). In [7], the following objective function is optimized by MF for implicit feedback datasets:

$$\mathcal{L}_{\mathrm{MF}} = \sum_{u,i} c_{ui}\left(a_{ui} - \theta_u^\top \theta_i\right)^2 + \lambda_\Theta \|\Theta\|_2^2, \tag{1}$$

where $u \in U$, $i \in I$, c_{ui} is the confidence level, and a_{ui} is a binary variable indicating the observed interaction for user u and item i in biadjacency matrix A (see Definition 1). Regularization parameter λ_Θ is the hyper-parameter that prevents overfitting the observations.

Among the various kinds of factorization-based models extended from MF, ranking-based models have shown their superiority in top-N recommendation problems. Ranking-based models such as BPR [14] and WARP [17] have shifted from traditional pointwise methods (direct relations) to pairwise methods (relative relations), the ranking loss of which can be defined as

$$\mathcal{L}_{\mathrm{rank}} = \sum_{u,(i,i')} \mathcal{F}\left(\theta_u^\top \theta_{i'}, \theta_u^\top \theta_i\right) + \lambda_\Theta \|\Theta\|_2^2, \tag{2}$$

where $i \in I$ indicates the observed (positive) items and $i' \in I$ indicates unobserved (negative) items for user $u \in U$.

Ranking-based models assume that users prefer observed items to unobserved ones, the goal of which is thus to discriminate the preferred item from each pair of items. In Eq. (2), function $\mathcal{F}$ represents the general objective function for the item ranking problem. Specifically, in BPR [14], $\mathcal{F}$ is the ranking function composed of a logistic sigmoid, a logarithm function, and an indicator function; in WARP [17], $\mathcal{F}$ is composed of a weight-approximated ranking function and an indicator function.

Graph Model. In addition to the commonly studied factorization-based models, several graph-based models have also been proposed that utilize random walk to rank items for recommendation [1–5, 16]. Specifically, the main concept of such models is to treat the user-item interaction matrix A as an interaction graph and bring in indirect preference information using random walks on the graph to provide recommendations. For instance, Fouss et al. [4] propose various scoring algorithms based on the parameters of random walks in the interaction graph G; Cooper et al. [2] extend this with simpler and faster scoring algorithms. In addition, Christoffel et al. [1] present $\mathrm{RP}^3(\beta)$, a graph node ranking recommendation algorithm to re-rank items based on 3-hop random walk transition probabilities.

HOP-Rec. Although factorization-based models implicitly infer user preferences for unobserved items, recent studies have showed that the explicit modeling of such potential preferences can improve the performance of recommender systems [10]. Inspired by these factorization and graph models, we build HOP-Rec, a united framework that (1) captures high-order preference information in a given user-item interaction matrix, and (2) scales to large-scale real-world datasets by using random surfing on the corresponding interaction graph. The objective of HOP-Rec is defined as

$$\mathcal{L}_{HOP} = \sum_{\substack{1 \le k \le K \\ u,(i,i')}} \overbrace{C(k)\,\mathbb{E}_{\substack{i \sim P_u^k \\ i' \sim P_N}}}^{\text{graph model}} \overbrace{\left[\mathcal{F}\left(\theta_u^\top \theta_{i'}, \theta_u^\top \theta_i\right)\right]}^{\text{factorization model}} + \lambda_\Theta \|\Theta\|_2^2, \quad (3)$$

where $P_u^k(\cdot)$ denotes the k-order probability distribution for an item sampled from the walk sequence S_u (see Definition 2), P_N denotes a uniform distribution by which an item is sampled from the set of all items, and K denotes the maximum order modeled in our method. The main idea behind the proposed method is the approximation of high-order probabilistic matrix factorization by conducting random walk (RW) with a decay factor for confidence weighting $C(k)$, where $0 < C(k) \le 1$. Note that instead of factorizing the matrix directly by matrix operations, which is not feasible for large-scale datasets, RW approximation has been proved efficient and accurate [3]. By doing so, we not only smooth the strict boundary between observed and unobserved items by introducing high-order preference information, but also make our method scalable to large-scale real-world datasets.

Specifically, we first introduce RW to explore the interaction graph G with respect to each user u; for a given walk sequence starting from u: $S_u = (u, i_1, u_1, \ldots, u_{k-1}, i_k, \ldots)$, item i_k with order k that user u potentially prefers (i.e., user u's k-th-neighbor item) is sampled. In addition, as the degree of users U and items I are usually power-law distributed in real-world datasets, most of the sampled paths are with low degree users and items when we apply uniform sampling. To take this into consideration, we utilize degree sampling in our RW procedure; that is, for each step in RW sampling, we sample users and items with probabilities $\propto deg(u), deg(i)$, respectively. Therefore, for $x \in U$ and $y \in I$ (or $x \in I$ and $y \in U$), the probability of sampling a k-th order neighbor vertex y for x can be derived as

$$p_x^k(y) = \begin{cases} \frac{a_{xy}\,deg(y)}{\sum_{y'} a_{xy'}\,deg(y')} & \text{if } k = 1 \text{ and } x \in U, \\ \frac{a_{yx}\,deg(y)}{\sum_{y'} a_{y'x}\,deg(y')} & \text{if } k = 1 \text{ and } x \in I, \\ p_x^1(\alpha) p_\alpha^{k-1}(\beta) p_\beta^1(y) & \text{if } k > 1, \end{cases} \quad (4)$$

where $deg(x)$ stands for the degree of x, and α (β) denotes the next node of x (the previous node of y) in the walk. Note that as G is a user-item bipartite graph, if $x \in U$ ($x \in I$), then $y, y' \in I$ ($y, y' \in U$, respectively) for $k = 1$, and the absolute transition probability from x to y can be approximated by RW with various paths from x to y, which simplifies the cumulative process of counting all the probabilities of intermediate nodes α, β.

Additionally, the confidence weighting parameter $C(k)$ is introduced to discriminate the strength between different orders of proximity. Inspired from the studies in [5, 12], we weight the k-order proximity by a decay factor $C(k) = 1/k$, and we update Eq. (3) with different orders of neighborhood items (i.e., $k = 1, 2, \ldots, K$) simultaneously. With this perspective, we enrich the originally sparse graph by inferring high-order proximity from user-item interactions.

Finally, the ranking objective function is composed with an indicator function and a pairwise logistic loss, and we thus define $\mathcal{F}$ in Eq. (3) as

$$\mathcal{F}(\theta_u^\top \theta_{i'}, \theta_u^\top \theta_i) = \mathbb{1}_{\{\theta_u^\top \theta_{i'} - \theta_u^\top \theta_i > \epsilon_k\}} \log \left[\sigma\left(\theta_u^\top \theta_{i'} - \theta_u^\top \theta_i\right)\right], \quad (5)$$

where $\mathbb{1}_B$ denotes the indicator function for condition B and ϵ_k is an order-aware margin set to ϵ/k. Note that item i' is uniformly sampled from the set of all items as we assume a simplified uniform distribution P_N for less preferred items; this assumption is reasonable since the preferred (observed) items are usually far outnumbered by less preferred (unobserved) items for each user in many real-world scenarios.

The objective in Eq. (3) is minimized using asynchronous stochastic gradient descent (ASGD) [11].

3 EXPERIMENTS

3.1 Datasets and Preprocessing

To examine the capability and scalability of HOP-Rec, we conduct experiments on four publicly available datasets that vary in terms of domain, size, and sparsity, as shown in Table 1. For each of the datasets, we discard those users and items with fewer than 5 associated interactions. In addition, we preprocess the interaction records for each dataset to simulate implicit binary feedback from users: 1) for the Amazon-book and MovieLens datasets, which contain explicit rating records, we transform ratings higher than 4 to 1 and the rest to 0 [10]; 2) for the CiteUlike dataset, no transformation is conducted as it is a binary preference dataset.

Table 1: Datasets

Dataset	CiteUlike[a]	MovieLens-1M[b]	MovieLens-20M[b]	Amazon-Book[c]
Users ($\|U\|$)	3,527	6,034	136,674	449,475
Items ($\|I\|$)	6,339	3,125	13,680	292,65
Feedback ($\|E\|$)	77,546	574,376	9,977,451	6,444,944
Density	0.347%	3.046%	0.534%	0.005%

[a] http://www.wanghao.in/data/ctrsr_datasets.rar
[b] https://grouplens.org/datasets/movielens
[c] http://jmcauley.ucsd.edu/data/amazon

3.2 Baseline Methods

Matrix Factorization (MF) [8]. MF is a well developed and commonly used technique for user-item recommendation. In the experiments, we use the *implicit* library,[3] which implements MF with an alternating least-square learning method [7].

Bayesian Personalized Ranking (BPR) [14]. BPR extends the pointwise method, MF, by incorporating pairwise ranking loss for personalized recommendations.

Weighted Approximate-Rank Pairwise (WARP) loss [17, 18]. WARP is an approximated approach to estimating the rank function efficiently, the main idea of which is to weigh pairwise violations depending on their position in the ranked list.

[3] https://github.com/benfred/implicit

Table 2: Performance comparison

	CiteUlike			MovieLens-1M			MovieLens-20M			Amazon-Book		
	P@10	R@10	MAP@10	P@10	R@10	MAP@10	P@10	R@10	MAP@10	P@10	R@10	MAP@10
MF	4.1%	13.1%	6.7%	17.7%	13.1%	11.7%	14.9%	14.0%	11.3%	0.7%	3.7%	1.4%
BPR	3.8%	14.2%	6.4%	18.1%	13.2%	12.5%	13.3%	14.3%	10.4%	1.0%	5.3%	2.5%
WARP	5.4%	18.3%	9.1%	24.8%	18.5%	18.5%	20.7%	21.4%	17.2%	1.4%	7.6%	3.2%
K-OS	5.6%	19.4%	9.5%	23.0%	17.3%	16.4%	19.6%	20.5%	15.7%	1.5%	7.9%	3.5%
$RP^3(\beta)$	5.9%	21.2%	3.2%	22.8%	17.2%	14.2%	17.3%	19.4%	10.3%	-	-	-
HOP	5.9%	**21.3%**	*__10.8%__	*__25.9%__	*__20.5%__	*__19.6%__	*__21.2%__	*__22.3%__	*__17.9%__	1.5%	7.9%	*__3.6%__
%Improv.	0.0%	0.5%	13.7%	4.4%	10.8%	5.9%	2.4%	4.2%	4.1%	0.0%	0.0%	2.9%

***K*-Order Statistic (*K*-OS) loss [19].** *K*-OS generalizes WARP by taking into account the set of positive examples during optimization, where K denotes the number of positive samples considered; note that K-OS degenerates to WARP when $K = 1$.

For the above three methods, we use the *lightfm* [9] library,[4] to conduct the experiments.

Popularity-based Re-ranking ($\mathbf{RP^3(\beta)}$) [1]. This method serves as a strong baseline method among the various graph-based methods. The method re-weights the ranking score proposed in [2] by moderating influence from blockbusters (high degree items). Evaluation of this method was carried out using the original library provided by the authors.[5]

3.3 Evaluation and Settings

For the top-N recommendations, we used the following three common evaluation metrics [6, 10]: (1) precision@N, (2) recall@N, (3) MAP@N. For each dataset, we randomly divided the binary interaction matrix into two parts: 80% as the training set and 20% as the testing set; the reported performance was averaged over 20 repetitions. Apart from $RP^3(\beta)$, the pure graph-based method, which takes the transition probability as its ranking score, the dimension of latent factors d was fixed at 120; in addition, all factorization-based methods together with the proposed HOP-Rec used the dot product of two latent factors as the scoring function. Other than d, all other hyperparameters for the compared methods were selected based on MAP using grid search with the testing set at the first time of evaluation. For the proposed HOP-Rec, we searched the best order K from 1 to 3 and sample times T from 60 to 1,200 million for different dataset sizes (best T: 90/60/300/800 for CiteUlike/MovieLens-1M/MovieLens-20M/Amazon-Book); θ_u for all $u \in U$ and θ_i for all $i \in I$ were uniformly initialized within the range of $\pm 0.5/d$, and $\epsilon = 1$ (see Eq. (5)). Recall that K denotes the maximum order modeled in our method.

3.4 Quantitative Analysis

Table 2 compares the performance of the proposed HOP-Rec with the other five baseline methods, in which $*$ denotes statistical significance at $p < 0.01$ (paired t-test) with respect to all baselines and *%Improv.* denotes the percentage improvement of the proposed method with respect to the best baseline performance. Note that resource limitations with the original library hindered our $\mathrm{RP}^3(\beta)$ experiments on Amazon-Book. As shown in the table, we observe that WARP, K-OS, and $\mathrm{RP}^3(\beta)$ serve as strong baseline methods as there is a clear performance gap between them and MF and BPR, the other two methods.

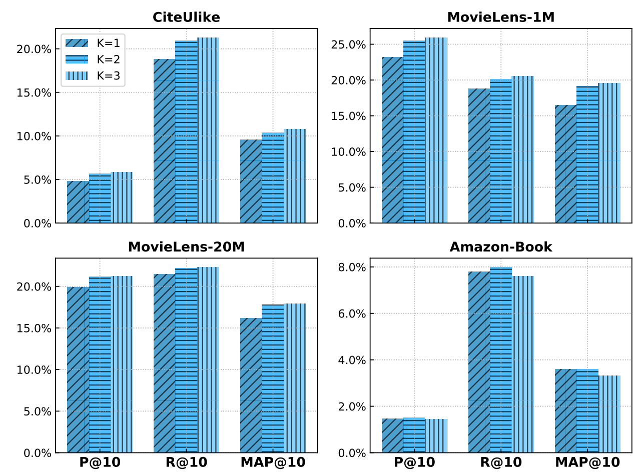

Figure 2: Sensitivity analysis with respect to K

Observe from Table 2, the proposed HOP-Rec generally yields performance superior or comparable to the five baseline methods. Specifically, our method significantly outperforms all of the other state-of-the-art methods in terms of MAP@10 (CiteUlike, MovieLens-1M, MovieLens-20M, and Amazon-Book), Recall@10 (CiteUlike, MovieLens-1M, and MovieLens-20M), and Precision@10 (both MovieLens datasets).

To further examine the effectiveness of the high-order proximity modeled in HOP-Rec, we evaluated the performance with respect to the parameter K, the results of which are illustrated in Figure 2. The results show that the performance of HOP-Rec benefited from incorporating high-order proximity, though for the largest dataset, Amazon-Book, the performance drops when $K = 3$.

4 CONCLUSION AND FUTURE WORK

We present HOP-Rec, a unified and efficient method of factorization and graph models that captures high-order information within a simple user-item interaction matrix. The effectiveness of the method is attested by experiments on four large-scale real-world datasets, the results of which suggest that incorporating high-order proximity with factorization model is promising for general top-N implicit recommendation problems.

[4] https://github.com/lyst/lightfm
[5] https://github.com/faberchri/MyMediaGraph

REFERENCES

[1] Fabian Christoffel, Bibek Paudel, Chris Newell, and Abraham Bernstein. 2015. Blockbusters and Wallflowers: Accurate, Diverse, and Scalable Recommendations with Random Walks. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys)*. 163–170.

[2] Colin Cooper, Sang Hyuk Lee, Tomasz Radzik, and Yiannis Siantos. 2014. Random Walks in Recommender Systems: Exact Computation and Simulations. In *Proceedings of the 23rd International Conference on World Wide Web (WWW)*. 811–816.

[3] Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. 2007. Random-Walk Computation of Similarities Between Nodes of a Graph with Application to Collaborative Recommendation. *IEEE Transactions on Knowledge and Data Engineering* 19, 3 (2007), 355–369.

[4] Francois Fouss, Alain Pirotte, and Marco Saerens. 2005. A Novel Way of Computing Similarities Between Nodes of a Graph, with Application to Collaborative Recommendation. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. 550–556.

[5] Marco Gori and Augusto Pucci. 2007. ItemRank: A Random-walk Based Scoring Algorithm for Recommender Engines. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI)*. 2766–2771.

[6] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems* 22, 1 (2004), 5–53.

[7] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM)*. 263–272.

[8] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[9] Maciej Kula. 2015. Metadata Embeddings for User and Item Cold-start Recommendations. In *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems co-located with 9th ACM Conference on Recommender Systems (CBRecSys@RecSys)*. 14–21.

[10] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M. Blei. 2016. Factorization Meets the Item Embedding: Regularizing Matrix Factorization with Item Co-occurrence. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys)*. 59–66.

[11] Feng Niu, Benjamin Recht, Christopher Re, and Stephen J. Wright. 2011. HOGWILD!: A Lock-free Approach to Parallelizing Stochastic Gradient Descent. In *Proceedings of the 24th International Conference on Neural Information Processing Systems (NIPS)*. 693–701.

[12] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report 1999-66. Stanford InfoLab.

[13] H. Park, J. Jung, and U. Kang. 2017. A comparative study of matrix factorization and random walk with restart in recommender systems. In *2017 IEEE International Conference on Big Data (Big Data)*. 756–765.

[14] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)*. 452–461.

[15] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intellelligence* 2009, Article 4 (2009).

[16] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast Random Walk with Restart and Its Applications. In *Proceedings of the Sixth International Conference on Data Mining (ICDM)*. 613–622.

[17] Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. WSABIE: Scaling Up to Large Vocabulary Image Annotation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*. 2764–2770.

[18] Jason Weston, Chong Wang, Ron Weiss, and Adam Berenzeig. 2012. Latent Collaborative Retrieval. In *Proceedings of the 29th International Coference on International Conference on Machine Learning (ICML)*. 443–450.

[19] Jason Weston, Hector Yee, and Ron J. Weiss. 2013. Learning to Rank Recommendations with the K-order Statistic Loss. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys)*. 245–248.