Trading-off Among Accuracy, Similarity, Diversity, and Long-tail: A Graph-based Recommendation Approach

Lei Shi
Baidu.com, Inc.
P. R. China
shilei06@baidu.com

ABSTRACT

Improving recommendation accuracy is the mostly focused target of recommendation systems, while it has been increasingly recognized that accuracy is not enough as the only quality criterion. More concepts have been proposed recently to augment the evaluation dimensions, such as similarity, diversity, long-tail, etc. Simultaneously considering multiple criteria leads to a multi-task recommendation. In this paper, a graph-based recommendation approach is proposed to effectively and flexibly trade-off among them.

Our approach is considered based a 1st order Markovian graph with transition probabilities between user-item pairs. A "cost flow" concept is proposed over the graph, so that items with lower costs are stronger recommended to a user. The cost flows are formulated in a recursive dynamic form, whose stability is proved to be guaranteed by appropriately lower-bounding the transition costs. Furthermore, a mixture of transition costs is designed by combining three ingredients related to long-tail, focusing degree and similarity. To evaluate the ingredients, we propose an orthogonal-sparse-orthogonal nonnegative matrix tri-factorization model and an efficient multiplicative algorithm. Empirical experiments on real-world data show promising results of our approach, which could be regarded as a general framework for other affects if transition costs are designed in various ways.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering*

General Terms

Algorithm, Proof, Experimentation, Performance

Keywords

Collaborative filtering, graph-based cost flow, fixed point stability, contraction mapping, orthogonal-sparse-orthogonal nonnegative matrix tri-factorization

ACM RecSys '2013 Hong Kong
Copyright 2013 ACM 978-1-4503-2409-0/13/10 ...$15.00.
http://dx.doi.org/10.1145/2507157.2507165.

1. INTRODUCTION

With extensive recommendation system technologies developed over the past decade, there emerge lots of successful efforts in both academia and industry, taking the well-known applications by Amazon and Netflix for example. The most common formulation of a recommendation problem adopts the notion of user-to-item ratings, with boolean, integer, or ordinal values. A recommendation system typically tries to predict the values (or orders) of the ratings of unrated items[1] for each user, with a satisfactory ranking [2, 5, 10].

The major existing efforts in the literature focused on improving the recommendation accuracy, as exemplified by the Netflix Prize competition [20]. However, it has been increasingly recognized that it is not sufficient to have accuracy as the only criterion during measuring the recommendation quality. Recently, more concepts have been considered as alternative evaluation dimensions to the recommended items, such as similarity, diversity, novelty, trust, long-tail, etc., to generate results that are not only accurate but also valuable with additional properties [2, 5, 10, 23, 25, 27].

If considering accuracy together with some of these additional targets, we encounter a multi-task recommendation and need to seek a solution that trades-off among them. Particularly, four evaluation concepts are chosen as our focus:

- **Accuracy** (*Acc*) of the predicted ratings or ranked recommendation list on unrated items are usually evaluated based on a held-out testing set, e.g., RMSE or MAE as the measure on ratings, Normalized Distance-based Performance Measure (NDPM) or Mean Reciprocal Rank (MRR) as the measure on ranking [14].
- **Similarity** (*Sim*) measures how similar the recommended items are (usually in semantics or topics) to the user or the user's rated items [27].
- **Diversity** (*Div*) describes the difference of recommended lists throughout all users [10].
- **Long-tail** (*Lt*) encourages the long-tail items, i.e., those have been rated seldom, to be recommended [2].

In different application scenarios, these four evaluation perspectives play different roles. First, accuracy should be the fundamental target in any recommendation systems. Moreover, as listed in Table 1 for example, the remaining three metrics tend be also valuable for recommendation on consumable goods. However, for the non-consumable goods, the similarity measure might be not that important because,

[1] Following existing recommendation studies, here and throughout this paper, "user" and "item" are not limited to their names, whereas they generally stand for the subjective side and the objective side, respectively.

e.g., a person bought a TV set is unlikely to buy another soon even if they are extremely similar. Moreover for food recommendation, accuracy and similarity that indicate the taste of a user become important, whereas diverse/long-tail items without qualified ratings are not preferred.

Table 1: Evaluations for recommendations. Symbol "✓" or "?" indicates an evaluation is or may be not important for the corresponding scenario

Example scenarios	*Acc*	*Sim*	*Div*	*Lt*
consumable goods	✓	✓	✓	✓
non-consumable goods	✓	?	✓	✓
movie/book/restaurant	✓	✓	✓	?
food	✓	✓	?	?

From another point of view, we'd like to discuss about the "satisfaction" of a recommendation system. Ideally, a recommendation system should receive satisfactions from not only the user-side but also the platform-side (e.g., Amazon, Netflix). Table 2 lists four types of feelings that involve satisfaction. First, highly related to accuracy and similarity, feeling a recommendation *matched* with users' need/tastes is important to both user-side and platform-side. Second, the novelty is usually contributed by diversity and long-tail, is the similarity is guaranteed. Third, to widely cover each user's interest, we'd better to enhance the similarity and diversity. Fourth, a platform is satisfactory and attractive to retailers if the item coverage throughout all users is wide, which concerns the diversity and long-tail factors.

Table 2: Satisfactions to recommendations. Different feelings are important on either or both of user-and-platform sides, which should have different key evaluation metrics (as indicated by ✓)

feeling	important to	*Acc*	*Sim*	*Div*	*Lt*
matched	user+platform	✓	✓	–	–
novel	user+platform	–	✓	✓	✓
user-cover	user	–	✓	✓	–
platform-cover	platform	–	–	✓	✓

Focusing on *Acc*, *Sim*, *Div* and *Lt*, this paper proposes a graph-based recommendation approach that can effectively and flexibly trade-off among them. At the beginning, we construct a directed graph with 1st order Markovian, whose transition probabilities are calculated based on rating weights. Given a user, a cost flow concept is proposed on this graph, and items with lower cost are stronger recommended. The costs are formulated in a recursive dynamic form. Thereafter, based on fixed point theorem, we prove that the stability of the dynamics is guaranteed by appropriately lower-bounding the transition cost of each edge. Accordingly, a mixed version of transition costs is designed by combining three ingredients related to long-tail, focusing degree and similarity. In order to evaluate the latter two ingredients, an orthogonal-sparse-orthogonal nonnegative matrix tri-factorization (OSO-NMTF) model is proposed, together with an efficient multiplicative algorithm with the help of gradients on Stiefel manifold. Empirical experiments on real-world data show promising results of our approach.

The remainder of this paper is organized as follows. In Section 2, we briefly review related efforts in the literature. Based on a directed graph representation, Section 3 introduces the cost flow formulation, provides the stability analysis, and proposes a mixture construction to the transition costs. The specification and learning algorithm are described in Section 4. After reporting experimental results in Section 5, we finally draw concluding remarks in Section 6.

2. RELATED WORKS

Among a large number of recommendation techniques that have been developed over the past decade, collaborative filtering (CF) techniques represent most widely used and well-performing algorithms. Aiming to directly predict the ratings, two representative CF techniques are neighborhood-based CF algorithms and matrix factorization based CF algorithms [14, 20, 21]. Moreover, there exist a stream of graph-based recommendation approaches, aiming to propagate information throughout a graph and mostly predict the ranking of the ratings [1, 3, 11, 15], taking the well-known PageRank and related algorithms for example.

Besides the differences on implementation algorithms, existing efforts also take different objective functions to be optimized. Most approaches that directly predict the ratings [14] adopt the estimation errors (e.g., RMSE, MAE) or further ranking accuracies (e.g., MRR, NDPM). Since it is increasingly recognized that merely caring about accuracy is not enough [23], papers [3, 11] further considered the similarity between items and users' tastes. The authors of [1, 10, 17, 22] focused on optimizing diversity, so that not only the redundancy of recommended items to each user is low, but also the coverage of recommended items to all users is large. Moreover, researchers also observed that most of existing recommender systems, especially CF based methods, can not recommend tail products due to the data sparsity issue. On the other, the success of "infinite-inventory" retailers such as Amazon and Netflix is greatly attributed to a long-tail phenomenon, where tail product availability is able to boost head sales by offering the shopping convenience for both mainstream and niche tastes. Under this motivation, recently several efforts [2, 5, 25, 27, 29] turned to take the long-tail effects into consideration. However, to the best of our knowledge, there is still no effort that considers these criteria simultaneously under a unified framework. This paper is thus motivated for such a study.

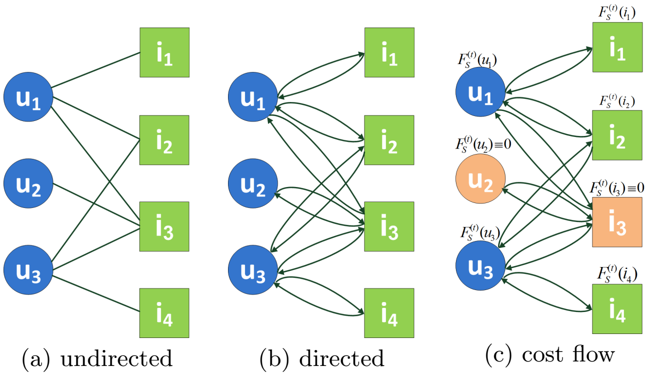
3. GRAPH-BASED COST FLOW FOR RECOMMENDATION

In this section, we introduce the graph representation of the user-item relation and the cost flow concept. Then, after proposing the flow dynamics, we proceed to analyze its stability and the transition cost composed of multiple effects.

3.1 Graph Construction

In recommendation scenarios, the user-item relationship is usually represented by an edge-weighted undirected bipartite graph $G'(V, E, \rho)$ as illustrated in Fig. 1(a), where V represents the set of vertices, E represents the set of edges, and ρ represents the weights on the edges. With V_u and V_i denoting the sets of user-vertices and item-vertices respectively, there is no edge between any two user-vertices or between two item-vertices. Clearly $V = V_u \cup V_i$, and the number of users is $N_u = |V_u|$ and of items is $N_i = |V_i|$. The edge set E has element $e(u, i)$ if and only if user $u \in V_u$ touches item $i \in V_i$ (e.g., rating in user-movie scenario, click in query-). Since the graph is undirected, the edge weights

$\rho(u,i)=\rho(i,u)$ are symmetric, and indicate the strength of the relation between the user-item pair (u,i).



(a) undirected (b) directed (c) cost flow

Figure 1: User(circle)-item(square) relation expressed in bipartite graphs. (a) undirected graph: each edge is associated with symmetric weights ρ. (b) directed graph: each edge is associated with asymmetric transition probability. (c) cost flow: each edge is associated with transition probability and cost; given absorbing nodes $S=S_u\cup S_i$, at time t each vertex j has a cost $F_S^{(t)}(j)$ to reach S, with $S_u=\{u_2\}$ and $S_i=\{i_3\}$ highlighted for instance

Like many existing works on random walk and Markov chain [18, 22, 24, 27], we convert the undirected $G'(V,E,\rho)$ into directed graph $G''(V,E,P)$ as illustrated in Fig. 1(b), where P is the set of transition probabilities on the edges. In graph $G''(V,E,P)$, every undirected edge in $G'(V,E,\rho)$ is converted into two directed edges with opposite directions. In $P=\{p_{iu},p_{ui}\}$, the p_{ui} denotes the transition probability from user-u to item-i, and p_{iu} is the transition probability from item-i to user-u, both obtained by normalization on ρ:

$$p_{ui}=\frac{\rho(u,i)}{\sum_i\rho(i,u)},\qquad p_{iu}=\frac{\rho(i,u)}{\sum_u\rho(i,u)}.\tag{1}$$

3.2 Cost Flow for Recommendation

In the recommendation scenario, given a single user or a user set S_u and its connected item set S_i, our target is to output a ranking list on all items based on some objective/criterion and recommend them to S_u.

Consider a random flow (walk) on graph $G''(V,E,P)$, we define $S=S_u\cup S_i$ as the absorbing/sink nodes, i.e., the flow stops when any node in S is reached. Moreover, we assume that each edge $e(j,\ell)$ has a nonnegative one-step transition cost $C(j|\ell)$ when a flow passes through. Finally, we have the cost flow graph $G(V,E,P,C)$, with an additional one-step transition costs C. An example is illustrated in Fig. 1(c). We define the long-term cost of flowing from a vertex j to be absorbed as $F_S(j)$, and $F_S(\ell)=0$ for $\forall\ell\in S$ by definition. Also, we require the vector $\boldsymbol{F}_S$ to locate in a simplex $\mathbb{S}$, i.e., nonnegative and sum equal to 1. Given $G(V,E,P,C)$ with appropriately defined C, we can output the ranking of $\boldsymbol{F}_S$ given $S=S_u\cup S_i$ as the recommendation list to user set S_u, where item-i with a smaller $F_S(i)$ is more preferred.

However, directly evaluating the flow cost F_S is quite difficult. Instead, we formulate it into the dynamic recursion $\boldsymbol{F}_S^{(t+1)}=f(\boldsymbol{F}_S^{(t)})$ w.r.t. the iteration time t as below.

Definition 1. (Cost flow dynamics). At iteration t, the cost to reach the set $S=S_u\cup S_i$ is defined as $\boldsymbol{F}_S^{(t)}=[F_S^{(t)}(1),\ldots,F_S^{(t)}(N)]$, and it flows in the graph specified by $\{p_{iu}\}\cup\{p_{ui}\}$ in the following dynamic way as t increases:

$$\begin{aligned}
\boldsymbol{F}_S^{(t+1)} &= f(\boldsymbol{F}_S^{(t)}),\quad\text{with}\quad F_S^{(t+1)}(j)=\frac{g^{(t)}(j)}{\sum_j g^{(t)}(j)}\\
g^{(t)}(i) &= \begin{cases}0, & i\in S_i,\\ \sum_{u'}p_{iu'}\left[C(u'|i)+F_S^{(t)}(u')\right], & i\notin S_i,\end{cases}\\
g^{(t)}(u) &= \begin{cases}0, & u\in S_u,\\ \sum_{i'}p_{ui'}\left[C(i'|u)+F_S^{(t)}(i')\right], & u\notin S_u,\end{cases}
\end{aligned}\tag{2}$$

where $C(u|i)$ describes the one-step *non-negative* transition cost from item-i to user-u, and vice versa for $C(i|u)$.

Remember that we start from a simplex initialization $\boldsymbol{F}_S^{(0)}\in\mathbb{S}$. First, the above dynamics capture the one-step cost flow in the user-item bipartite graph. Second, at any time t, it not only guarantees $F_S^{(t)}(j)=0$ for $\forall j\in S$, but also ensures each $F_S^{(t)}(j)\geq 0$ and $\sum_j F_S^{(t)}(j)=1$, i.e., $\boldsymbol{x}\in\mathbb{S}\Rightarrow f(\boldsymbol{x})\in\mathbb{S}$.

3.3 Stability Analysis of Flow Dynamics

One crucial analysis of the dynamics in Eq. (2) is to check its stability, i.e., whether it can converge to a unique equilibrium with any random initialization. This part introduces a sufficient condition that guarantees the stability.

Our analysis begins with the *Banach Fixed Point Theorem* as restated in Theorem 1, and Theorem 2 relates the contraction mapping with its Jacobian matrix [12, 19].

Theorem 1. *Consider a set $\mathcal{D}\in\mathbb{R}^n$ and a function $f:\mathcal{D}\to\mathbb{R}^n$, if (1) $\mathcal{D}$ is a closed set; (2) $\boldsymbol{x}\in\mathcal{D}\Longrightarrow f(\boldsymbol{x})\in\mathcal{D}$; (3) $f(\cdot)$ is a contraction mapping on $\mathcal{D}$, then:*
(1) there is one unique $\boldsymbol{x}^*\in\mathcal{D}$ s.t. $f(\boldsymbol{x}^*)=\boldsymbol{x}^*$; (2) iteration $\boldsymbol{x}^{(t+1)}=f(\boldsymbol{x}^{(t)})$ from $\forall\boldsymbol{x}^{(0)}\in\mathcal{D}$ converges to $\boldsymbol{x}^*$ as $t\to\infty$.

Theorem 2. *Assume $\mathcal{D}\in\mathbb{R}^n$ is convex and $f:\mathcal{D}\to\mathbb{R}^n$ has continuous partial derivatives in $\mathcal{D}$. Denote $f'(\boldsymbol{x})$ as the Jacobian matrix. If $\exists\ q<1$ s.t.,*

$$\forall\boldsymbol{x}\in\mathcal{D},\quad \left|\left|f'(\boldsymbol{x})\right|\right|\leq q$$

for some matrix norm $||\cdot||$, then f is a contraction mapping.

Based on these theorems, we analyze the mapping $\boldsymbol{F}_S^{(t+1)}=f(\boldsymbol{F}_S^{(t)})$ in Eq. (2) and obtain the Jacobian matrix

$$\boldsymbol{J}(\boldsymbol{F}_S^{(t)}) = \begin{bmatrix}\frac{\partial F_S^{(t+1)}(1)}{\partial F_S^{(t)}(1)} & \cdots & \frac{\partial F_S^{(t+1)}(1)}{\partial F_S^{(t)}(N)}\\ \vdots & \ddots & \vdots\\ \frac{\partial F_S^{(t+1)}(N)}{\partial F_S^{(t)}(1)} & \cdots & \frac{\partial F_S^{(t+1)}(N)}{\partial F_S^{(t)}(N)}\end{bmatrix},\tag{3}$$

with the following elements

$$\begin{cases}\frac{\partial F_S^{(t+1)}(u)}{\partial F_S^{(t)}(u')}=\frac{-F_S^{(t+1)}(u)\sum_{i'}p_{i'u'}}{\sum_j g^{(t)}(j)},\\ \frac{\partial F_S^{(t+1)}(u)}{\partial F_S^{(t)}(i')}=\frac{p_{ui'}-F_S^{(t+1)}(u)\sum_{u'}p_{u'i'}}{\sum_j g^{(t)}(j)},\\ \frac{\partial F_S^{(t+1)}(i)}{\partial F_S^{(t)}(i')}=\frac{-F_S^{(t+1)}(i)\sum_{u'}p_{u'i'}}{\sum_j g^{(t)}(j)},\\ \frac{\partial F_S^{(t+1)}(i)}{\partial F_S^{(t)}(u')}=\frac{p_{iu'}-F_S^{(t+1)}(i)\sum_{i'}p_{i'u'}}{\sum_j g^{(t)}(j)}.\end{cases}\tag{4}$$

In the below, we show that simply lower-bounding all one-step transition costs $\{C(u|i),C(i|u)\}_{i,u}$ appropriately is sufficient to guarantee the flow dynamics' stability.

Theorem 3. *(Lower-bounded transition cost.) If all one-step transition costs are lower-bounded by 1, i.e,:*

$$C(u|i) \geq 1, \quad C(i|u) \geq 1, \quad \forall i, u,$$

then the Jacobian matrix in Eq. (3) has a bounded infinite-norm $||\boldsymbol{J}||_\infty \leq q < 1$. Consequently, $f(\cdot)$ is a contraction mapping, and there exists a unique equilibrium $\boldsymbol{F}_S^* \in \mathbb{S}$, s.t. iterating Eq. (2) from $\forall \boldsymbol{F}_S^{(0)} \in \mathbb{S}$ converges to $\boldsymbol{F}_S^*$ as $t \to \infty$.

The proof of Theorem 3 is sketched in Appendix A.

3.4 Mixed Transition Cost

We proceed to design the detailed one-step transition cost $C(u|i)$ and $C(i|u)$ for each linked user-item pair (u, i). It should be noted that, the transition cost is a general concept and not limited to the designs presented below, choices alternative to which deserve study interests in future.

Particularly, we consider three factors that encourage **similarity**, **long tail**, and **focusing degree**, respectively. All of them are nonnegative costs.

- **Similarity.** The intuition is that, a transition should have small/large cost if the current user-item pair is similar/diverse. We define $C_{sim}(u|i)$ as the similarity cost for item-to-user transition and $C_{sim}(i|u)$ for user-to-item transition.
- **Long tail.** As discussed in Section 1, recommending the long-tail items may bring amazing benefits in many recommendation scenarios. We are thus motivated to define the long tail cost $C_{lt}(i)$ on item-i when transiting from some user to item-i. The larger sum of weights connected to item-i, the larger $C_{lt}(i)$ should be. We choose the simplest choice:

$$C_{lt}(i) \propto \sum_u \rho(i, u). \tag{5}$$

  The long tail on the user side does not seem quite meaningful and is not considered, or say $C_{lt}(u) \equiv 0$.
- **Focusing degree.** Suppose there are two users, u_1 and u_2. User u_1 has a wide range of interest types, while user u_2 centers on few specific tastes. Then the information shared by the specific user u_2 may be more important to distinguish the specificity of the items than the ambiguous user u_1. Based on this intuition and following [27], we use the latent topic/cluster to evaluate the focusing degree. Suppose we have clustered users into K_u clusters and items into K_i clusters, the focusing degree cost when transiting to a vertex is chosen as the posterior entropy:

$$\begin{aligned} C_{foc}(i) &= -\sum_{k=1}^{K_i} p(k|i) \log p(k|i), \\ C_{foc}(u) &= -\sum_{k=1}^{K_u} p(k|u) \log p(k|u). \end{aligned} \tag{6}$$

Combining the above three factors, we summarize our final one-step transition cost $C(u|i)$ and $C(i|u)$ as below:

$$\begin{aligned} C(u|i) &= 1 + \pi_{sim} C_{sim}(u|i) + \pi_{foc} C_{foc}(u), \\ C(i|u) &= 1 + \pi_{sim} C_{sim}(i|u) + \pi_{lt} C_{lt}(i) + \pi_{foc} C_{foc}(i), \\ \text{with} & \quad \pi_{sim}, \pi_{lt}, \pi_{foc} \geq 0, \quad \pi_{sim} + \pi_{lt} + \pi_{foc} = 1, \end{aligned} \tag{7}$$

which is a linear mixture of the three factors. The constant 1 is added for stability due to Theorem 3, which also serves as a basic cost similar to the "Absorbing Time" method [27].

Connected Subgraph v.s. Smoothed Full Graph.

In implementation, we find that the iteration by Eq. (2) converges very fast, e.g., almost always within 5 rounds. However, each iteration has a complexity of $\mathcal{O}(|E|)$ with $|E|$ being the number of edges. In order to scale the graph size and improve the efficiency, given the absorbing set S, we consider to iteratively extract a connected subgraph G_S:

- (Step 0) Set $G_S = S$;
- (Step 1) For $\tau = 1, \ldots, \lambda$:
  (1.1) Find $AccG_S = \{\text{node } \ell : \ \ell \text{ is connected to } G_S\}$;
  (1.2) Accumulate the set by $G_S = G_S \cup AccG_S$;

and then propagate the flow costs by Eq. (2) only through the connected subgraph G_S. This is based on a reasonable intuition: a group of users and items they like are connected. The round number is chosen as $\lambda = 2$ and enough to provide desired results based on our experiences. On the other extreme, some works [22] also smooth the graph by adding edges between each pair of users, resulting in a wholly connected full graph. We do not follow that way for efficiency.

Up till now, we still have not explained the detailed designs on C_{foc} and C_{sim}. First, the computation of C_{foc} in Eq. (6) requires clustering both users and items. In the next section, we will propose a model named as OSO-NMTF that clusters both sides simultaneously. Second, one can certainly evaluate the user-item similarity based on meta features if available (e.g., text description or genre of users and items). However for applications with no or limited meta information, the problem is how to measure the user-item similarity C_{sim} given only the graph weights, for which a novel similarity measure is proposed based on the OSO-NMTF model.

4. OSO-NMTF MODELING

There are extensive studies under the name of topic models in the literature [26], which show promising performances on finding meaningful latent topics. Intuitively, with each topic representing a cluster, the topic based recommendation or ranking [15, 27] owe its success to the abstracted high-level topic features, which are more robust and meaningful than the raw low-level features. Nevertheless, most of existing works only focus on clustering either one of users or items. Aiming to not only implement user-item bi-clustering but also extract user-item similarity simultaneously, we adopt the nonnegative matrix tri-factorization (NMTF) formulation [8, 28]. This section introduces the OSO-NMTF model that assigns orthogonal-sparse-orthogonal constraints, and proposes a multiplicative learning algorithm. Thereafter, we construct the *similarity* cost C_{sim} and *focusing degree* cost C_{foc} in Eq. (7) based on OSO-NMTF.

4.1 The OSO-NMTF Model

In our user-item scenario, a nonnegative matrix $\boldsymbol{X} \in \mathbb{R}_+^{N_u \times N_i}$ is constructed with $x_{ij} = \rho(i, j)$, where $\rho(i, j)$ is the graph weight in Eq. (1). Given $\boldsymbol{X}$, NMTF decomposes it into a product of three nonnegative factors $\boldsymbol{W} \in \mathbb{R}_+^{N_u \times K_u}$, $\boldsymbol{A} \in \mathbb{R}_+^{K_u \times K_i}$, and $\boldsymbol{H} \in \mathbb{R}_+^{N_i \times K_i}$, such that $\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{A}\boldsymbol{H}^T$. Further imposing orthogonality on both $\boldsymbol{W}$ and $\boldsymbol{H}$ leads to the orthogonal NMTF [8, 28], which can be achieved via solving the following optimization:

$$\begin{aligned} \min_{\boldsymbol{W}, \boldsymbol{A}, \boldsymbol{H} \geq \boldsymbol{0}} & \quad \frac{1}{2} \left|\left| \boldsymbol{X} - \boldsymbol{W}\boldsymbol{A}\boldsymbol{H}^T \right|\right|_F^2, \\ s.t. & \quad \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}, \ \boldsymbol{H}^T\boldsymbol{H} = \boldsymbol{I}, \end{aligned} \tag{8}$$

where $||\cdot||_F$ denotes the matrix Frobenius norm. Nonnegativity plus strict orthogonality on $\boldsymbol{W}$ and $\boldsymbol{H}$ are difficult to be ensured during optimization, because it will leads each row in both $\boldsymbol{W}$ and $\boldsymbol{H}$ to be all zeros except only one element equals 1, i.e., a typical combinatorial optimization. In consequence, paper [8] proposes a multiplicative algorithm based on an approximate Lagrange, and [28] uses gradients on Stiefel manifold [9] for an approximate updating.

Bi-Clustering Interpretation. From a clustering perspective, the O-NMTF formulation in Eq. (8) corresponds to the simultaneous clustering of the rows (users) and columns (items) of $\boldsymbol{X}$ [7, 8]. Particularly, there are K_u user-clusters and K_i item-clusters, $\boldsymbol{W}$ is the cluster indicator matrix for clustering users, $\boldsymbol{H}$ is the cluster indicator matrix for clustering items, $\boldsymbol{A}$ is the association weight matrix between user-clusters and item-clusters. It should be noted that, despite the approximate orthogonality on $\boldsymbol{W}$ and $\boldsymbol{H}$, they can be still regarded proportional to the *soft*-clustering posteriors. More interpretations and relations to other clustering methods are referred to [7, 13, 16, 21].

The Orthogonal-Sparse-Orthogonal Constraints. As a revised extension to O-NMTF in Eq. (8), we further assume that the association matrix $\boldsymbol{A}$ is sparse. Albeit this tiny revision, it brings the following three main benefits:

- **More interpretable associations.** Constraining $\boldsymbol{A}$ to be sparse encourages the noisy associations between user-clusters and item-clusters to shrink, and thus the learned $\boldsymbol{A}$ is more robust and interpretable [16].
- **Relieve indeterminacy.** Although the uniqueness of exact O-NMTF is claimed in [8], the have-to-be approximate updating in [8, 28] cannot guarantee the uniqueness. Consider a solution $\boldsymbol{WAH}^T$ with $\boldsymbol{W}^T\boldsymbol{W} \approx \boldsymbol{I}$, we may find a rotation matrix $\boldsymbol{\phi}$ with small rotation angle to remain nonnegativity, and let $\widetilde{\boldsymbol{W}} = \boldsymbol{W\phi}$ and $\widetilde{\boldsymbol{A}} = \boldsymbol{\phi}^T\boldsymbol{A}$. As a result, $\widetilde{\boldsymbol{W}}^T\widetilde{\boldsymbol{W}} = \boldsymbol{W}^T\boldsymbol{W}$ and $\boldsymbol{WA} = \widetilde{\boldsymbol{W}}\widetilde{\boldsymbol{A}}$ still hold without changing the objective in Eq. (8), i.e., it is ill-posed with indeterminacy, which can be relived by imposing sparsity on $\boldsymbol{A}$.
- **Model complexity adaptation.** The determination of cluster numbers K_u and K_i is a typical model selection problem [4]. As another by-product, the sparsity also encourages the rows/columns in $\boldsymbol{A}$ to approach zeros, if the corresponding user-clusters/item-clusters are redundant, leading to an adaptive model complexity determination and desired generalization ability.

Mathematically, orthogonal-sparse-orthogonal NMTF (OSO-NMTF) is formulated into the following optimization:

$$\begin{aligned}\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{A}\geq\boldsymbol{0}} \quad & \frac{1}{2}\left|\left|\boldsymbol{X} - \boldsymbol{WAH}^T\right|\right|_F^2 + \lambda\sum_{i,j} A_{ij}, \\ s.t. \quad & \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}, \quad \boldsymbol{H}^T\boldsymbol{H} = \boldsymbol{I}, \end{aligned} \tag{9}$$

where $\lambda > 0$ is the Lagrange multiplier, and the second term is the classic L_1-norm for sparsity [16].

4.2 Multiplicative Updating Algorithm

To solve the constrained optimization of OSO-NMTF in Eq. (9), one may augment it into an approximate Lagrangian similar to [8], and determine three Lagrange multipliers on $\boldsymbol{W}$, $\boldsymbol{H}$, $\boldsymbol{A}$, respectively. In contrast, the orthogonality constraint on $\boldsymbol{W}$ and $\boldsymbol{H}$ indicates the Stiefel manifold, the geometry of which was systematically studied in [9]. Employing the gradient on Stiefel manifold similar to [28], we obtain the following multiplicative updates:

$$\begin{aligned}\boldsymbol{W} &\longleftarrow \boldsymbol{W} \circ \frac{\left[(\boldsymbol{X} + \boldsymbol{WAH}^T)\boldsymbol{HA}^T\right]}{\left[\boldsymbol{WAH}^T(\boldsymbol{HA}^T + \boldsymbol{X}^T\boldsymbol{W})\right]}, \\ \boldsymbol{H} &\longleftarrow \boldsymbol{H} \circ \frac{\left[(\boldsymbol{X}^T + \boldsymbol{HA}^T\boldsymbol{W}^T)\boldsymbol{WA}\right]}{\left[\boldsymbol{HA}^T\boldsymbol{W}^T(\boldsymbol{WA} + \boldsymbol{XH})\right]}, \\ \boldsymbol{A} &\longleftarrow \boldsymbol{A} \circ \frac{\left[\boldsymbol{W}^T\boldsymbol{XH}\right]}{\left[\lambda + \boldsymbol{W}^T\boldsymbol{WAH}^T\boldsymbol{H}\right]}, \end{aligned} \tag{10}$$

where operator $\circ$ denotes the Hadamard (element-wise) product, operator $\frac{[\cdot]}{[\cdot]}$ is the Hadamard (element-wise) division. The derivation of this algorithm is sketched in Appendix B. In experiments, this updating converges quickly and effectively preserves a rough orthogonality, with low generalization errors when missing values are predicted.

4.3 User-Item Bi-Clustering and Similarity

Once an OSO-NMTF model is learned, the *focusing degree* cost C_{foc} and the *similarity* cost C_{sim} in Eq. (7) can be obtained. Remember the bi-clustering properties of OSO-NMTF discussed in Section 4.1, $\boldsymbol{W}$ and $\boldsymbol{H}$ describe the user-cluster and item-cluster posteriors, respectively. Intuitively, a user or item should have a smaller/larger cost C_{foc} if its cluster posteriors are shaper/smoother. More concretely, we employ entropy for the quantification:

$$\begin{aligned} C_{foc}(u) &\propto -\sum_{k=1}^{K_u} \bar{W}_{uk}\log\bar{W}_{uk}, \text{ with } \bar{\boldsymbol{W}} = \text{normalize}(\boldsymbol{W}), \\ C_{foc}(i) &\propto -\sum_{k=1}^{K_i} \bar{H}_{ik}\log\bar{H}_{ik}, \text{ with } \bar{\boldsymbol{H}} = \text{normalize}(\boldsymbol{H}), \end{aligned} \tag{11}$$

where normalize$(\cdot)$ is the row normalization operator such that each row in $\bar{\boldsymbol{W}}$ and $\bar{\boldsymbol{H}}$ has a posterior-over-clusters form. Again, paper [27] only focuses on clustering users by Latent Dirichlet Allocation, while our C_{foc} is constructed by clustering both items and users simultaneously.

For the user-item similarity C_{sim} in Eq. (7), here we introduce a novel user-item similarity measure based on OSO-NMTF. Consider an item i and a user u, they are represented by a column $\boldsymbol{X}_{*i} \in \mathbb{R}_+^{N_u}$ and a row $\boldsymbol{X}_{u*} \in \mathbb{R}_+^{N_i}$, respectively. Using the normalized $\bar{\boldsymbol{W}}$ and $\bar{\boldsymbol{H}}$ as in Eq. (11), we can explain $\boldsymbol{\alpha}_i = \bar{\boldsymbol{W}}^T\boldsymbol{X}_{*i}$ as the accumulated posteriors over the *user-clusters* contributed by this *item i*. The case is similar for $\boldsymbol{\beta}_u = \bar{\boldsymbol{H}}^T\boldsymbol{X}_{u*}^T$. Given the association $\boldsymbol{A}$, we thus transfer to measure the similarity between $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_u$:

$$C_{sim}(i|u) = C_{sim}(u|i) \propto \frac{1}{\boldsymbol{\alpha}_i^T\boldsymbol{A}\boldsymbol{\beta}_u / \sum_{ij} A_{ij}}, \tag{12}$$

where a symmetric similarity is assumed in our implementations. In case of some applications, one may want to use asymmetric to differentiate the bi-directions. In that case, one possible alternative choice would be:

$$\begin{aligned} C_{sim}(i|u) &\propto \frac{1}{\boldsymbol{\alpha}_i^T\text{normalize}(\boldsymbol{A})\boldsymbol{\beta}_u}, \\ C_{sim}(u|i) &\propto \frac{1}{\boldsymbol{\alpha}_i^T[\text{normalize}(\boldsymbol{A}^T)]^T\boldsymbol{\beta}_u}. \end{aligned} \tag{13}$$

That is, normalizing the association matrix $\boldsymbol{A}$ by row will get the transition probability from user-clusters to item-clusters, and vice versa for normalization by column. Moreover, one may also consider to normalize $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_u$ before product, in order to balance the different weights of users and items.

5. EMPIRICAL EXPERIMENTS

5.1 Data Description

In experiments, we consider two real world tagging/rating datasets: `Movielens100K` and `Last.fm`. Collected by the GroupLens Research Project at the University of Minnesota, the `Movielens100K` dataset[2] consists of user-to-movie ratings (in the range of $1 \sim 5$), together with basic tag information of both users and movies. Collected from Last.fm online music system, the `Last.fm` dataset[3] contains user-to-artist tagging (i.e., binary ratings), together with the tagged properties and user social network information.

We preprocess the datasets as follows. For `Movielens100K`, we filter out ratings that are less than 4 (i.e., neutral or negative), and then filter out users who have rated less than 10 items. Moreover, each movie has a genre tag out of 18 possible values. In order to evaluate performance *during testing*, for each user, his/her user-tag vector is obtained by accumulating the genre tags of items rated by him/her. For `Last.fm`, we filter out users who have tagged less than 10 items. Moreover, a user tagged an artist by description phrases. Similarly, we accumulate all tags posted by each user as his/her user-tag vector, and all tags received as the item-tag vector for each item. Thereafter, each tag-vector is normalized so that elements' sum equals to 1. We denote $\boldsymbol{\eta}_u$ and $\boldsymbol{\chi}_i$ as tag-vectors to a user u and an item i, respectively.

Table 3: Description of preprocessed datasets

	`Movielens100K`	`Last.fm`
#users	886	718
#items	1,004	11,924
#ratings	10,514	51,999
sparsity	1.18%	0.61%
#user/item tags	18	8,746
#train : #test	2 : 8	8 : 2

Several key statistics of the data after preprocessing are summarized in Table 3. In order to further decrease the sparsity in `Movielens100K`, we randomly pick 20% ratings for training and leave the remaining 80% for testing, while the training v.s. testing ratio for `Last.fm` is 80% v.s. 20%.

5.2 Performance Evaluation

For each user u, we are given a list of truly rated item set $\boldsymbol{\psi}_u$ based on the testing data. During recommendation, once an algorithm proposes a ranked list $\boldsymbol{\zeta}_u$ of all unrated items (in the training data) by each user u, we use $\zeta_u(\ell)$ to denote the ℓ-th most strongly recommended item, and $\boldsymbol{\zeta}_u^{10}$ to represent the set of top 10 recommended items. Four metrics [6] are chosen for evaluating experimental performances:

- **Accuracy**: The accuracy given testing data is evaluated by the Mean Reciprocal Rank (MRR) of all users:

$$\mathcal{MRR} = \frac{1}{N_u} \sum_{u=1}^{N_u} RR_u,$$
$$RR_u = \max_{\ell} \frac{1}{\ell} \cdot \mathbf{1}\left[\text{item } \zeta_u(\ell) \in \boldsymbol{\psi}_u\right]. \quad (14)$$

  Therein, $\mathbf{1}[x]$ denotes the indicator function and equals to 1 (or 0) if expression x is true (or false). A larger $\mathcal{MRR}$ value indicates better accuracy.

- **Similarity**: The similarity score for each user u is measured by comparing the tag vectors of items in $\boldsymbol{\zeta}_u^{10}$ with the tag vector of user u:

$$\mathcal{SIM} = \frac{1}{N_u} \sum_{u=1}^{N_u} \sum_{\ell \in \boldsymbol{\zeta}_u^{10}} \boldsymbol{\eta}_u^T \boldsymbol{\chi}_\ell \quad , \quad (15)$$

  which uses the inner-products between the strongly recommended items in $\boldsymbol{\zeta}_u^{10}$ with the user. The tag vectors $\boldsymbol{\eta}$ and $\boldsymbol{\chi}$ are described in Section 5.1. A larger $\mathcal{SIM}$ value represents a better result in similarity.

- **Diversity**: The diversity metric is evaluated as

$$\mathcal{DIV} = \left| \cup_{u=1}^{N_u} \boldsymbol{\zeta}_u^{10} \right|, \quad (16)$$

  which describes the coverage size of the recommended item sets throughout all users. A larger $\mathcal{DIV}$ corresponds to a more diverse recommendation result.

- **Long-tail**: We measure the long-tail score by

$$\mathcal{LT} = \frac{1}{N_u} \sum_{u=1}^{N_u} \sum_{\ell \in \boldsymbol{\zeta}_u^{10}} w_\ell, \quad w_\ell = \frac{\sum_{u=1}^{N_u} \rho(\ell, u)}{\sum_{\ell=1}^{N_i} \sum_{u=1}^{N_u} \rho(\ell, u)}, \quad (17)$$

  where $\rho(\cdot,\cdot)$ is rating weight same as in Eq. (1). A smaller $\mathcal{LT}$ value indicates a larger long-tail coverage proportion of the top 10 recommendation sets.

5.3 Experimental Results

In order to investigate how each performance metric changes as the mixing weights $(\pi_{sim}, \pi_{lt}, \pi_{foc})$ in Eq. (7) vary, we traverse on grids of the three weights' configurations within the 3-dimensional simplex, as illustrated in Fig. 2. This enumeration is implemented on both `Movielens100K` and `Last.fm` datasets. Two representative CF methods are considered in comparison, namely Matrix Factorization (MF) [20] and Nonnegative Matrix Factorization (NMF) [21]. The latent bi-dimensionalities of OSO-NMTF model are set as $K_u = 30$ and $K_v = 50$, and the latent dimensionalities of MF and NMF are both fixed as 30 for a fair comparison.

Once OSO-NMTF has been trained, the three ingredients of transition costs are computed according to Eqs. (5&11&12). Enumerating through *hundreds of* gridded configurations of $(\pi_{sim}, \pi_{lt}, \pi_{foc})$ in the simplex, we perform our cost flow approach with each configuration. Finally, the performances on `Movielens100K` and `Last.fm` are reported in Fig. 3 and Fig. 4, respectively, where each subfigure illustrates one metric's values over the simplex same as in Fig. 2. The max and min values of each simplex are reported in Table 4, compared with the results by MF and NMF.

As shown by Figs. 3&4, each metric varies smoothly within the simplex, and the tendencies are similar on the two data. Interestingly, the cost ingredients C_{lt} and C_{sim} play correlated roles in some extent: large weights on them encourage relatively large $\mathcal{MRR}$ (high accuracy), small $\mathcal{DIV}$ (low diversity), and large $\mathcal{LT}$ (low long-tail ratio). Even more, the "focusing-degree" cost C_{foc} brings forward much more diverse and long-tailed results than the "long-tail" cost C_{lt}. Roughly, the "best" values for all metrics are provided with only two cost ingredients combined (i.e., locate near to the simplex edges). Particularly, the best $\mathcal{MRR}$ comes from combining C_{lt} and C_{sim}, while all the best values of $\mathcal{SIM}$, $\mathcal{DIV}$ and $\mathcal{LT}$ result from combining C_{foc} and C_{sim}. We may imagine C_{sim} as a *bridge* that connects C_{lt} and C_{foc} to slither among good environments for different metrics.

[2] http://movielens.umn.edu

[3] http://ir.ii.uam.es/hetrec2011

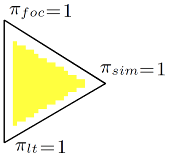
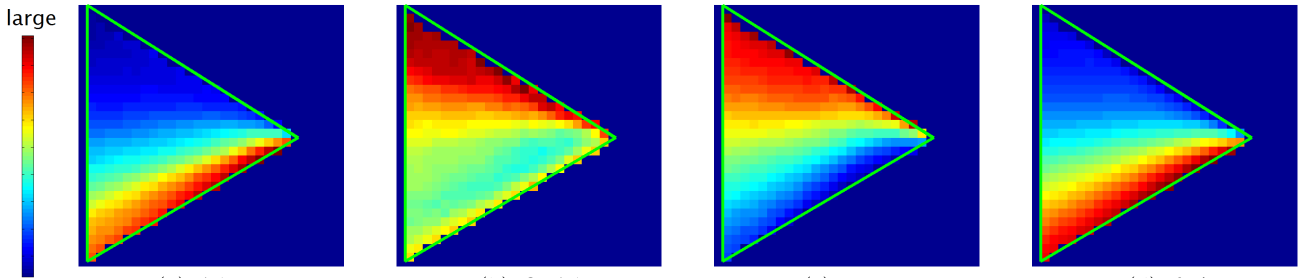


Figure 2: The simplex

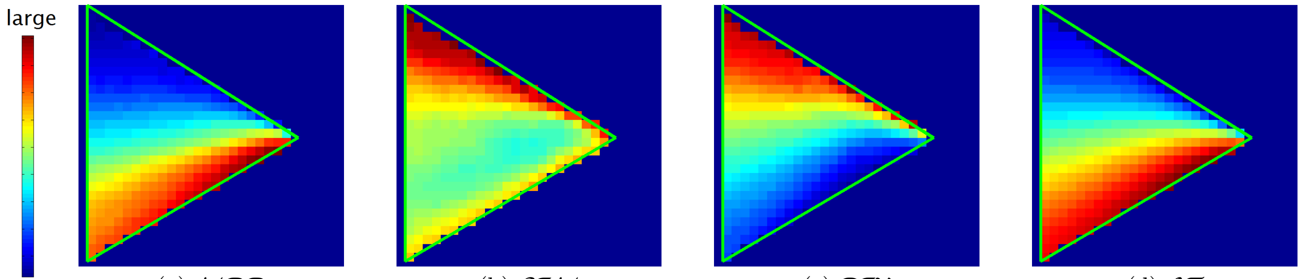
Table 4: Results comparison. For each metric on each data, the "best" is bolded

datasets	`Movielens100K`				`Last.fm`			
evaluation metrics	$\mathcal{MRR}$	$\mathcal{SIM}$	$\mathcal{DIV}$	$\mathcal{LT}$	$\mathcal{MRR}$	$\mathcal{SIM}$	$\mathcal{DIV}$	$\mathcal{LT}$
MF	0.493	1.05	19	0.059	0.121	0.105	35	0.015
NMF	0.512	1.04	17	0.060	0.127	0.109	32	0.016
max value in the simplex	**0.632**	**1.23**	**535**	0.056	**0.205**	**0.182**	**1041**	0.014
min value in the simplex	0.120	1.04	73	**0.014**	0.064	0.111	588	**0.003**



(a) $\mathcal{MRR}$ (b) $\mathcal{SIM}$ (c) $\mathcal{DIV}$ (d) $\mathcal{LT}$

Figure 3: Performances of our approach on `Movielens100K` within the simplex of Fig. 2 (best viewed in color)



(a) $\mathcal{MRR}$ (b) $\mathcal{SIM}$ (c) $\mathcal{DIV}$ (d) $\mathcal{LT}$

Figure 4: Performances of our approach on `Last.fm` within the simplex of Fig. 2 (best viewed in color)

In Table 4, all the "best" performances of each metric over the simplex are much better than those by MF and NMF. Moreover, even the "worst" performances, except $\mathcal{MRR}$, are better than or at least comparable to MF and NMF. Specifically, MF and NMF tend to recommend top items with small $\mathcal{DIV}$ and large $\mathcal{LT}$. In contrast, in the simplex not only the smallest $\mathcal{DIV}$ is higher, but also the largest $\mathcal{LT}$ is smaller than MF/NMF. That is, our approach performs robust with much better similarity, diversity and long-tail measures.

Time cost. Once transition costs are calculated, our approach includes no training phase. The average prediction time per user is within 0.2 second on `Movielens100K` and within 3 seconds on `Last.fm`, implemented in Python and using an Intel Xeon 2.4G single core CPU.

6. CONCLUSIONS

It is increasingly recognized that accuracy is not enough as the only quality criterion to a recommendation system, and more concepts have been proposed recently as additional evaluation dimensions. Simultaneously considering accuracy, similarity, diversity, and long-tail, this paper proposes a graph-based recommendation approach that effectively and flexibly trades-off among them. Particularly, a cost flow concept is proposed based on a 1st order Markovian graph with transition probabilities between user-item pairs. The costs are further formulated in a recursive dynamic form, whose stability is proved to be guaranteed if the transition costs are appropriately lower-bounded. Furthermore, a mixed version of transition costs is designed by combining three ingredients related to long-tail, focusing degree and similarity. To evaluate the ingredients, we propose an orthogonal-sparse-orthogonal nonnegative matrix tri-factorization model and an efficient multiplicative updating algorithm. Experiments on real-world data show valid performances of our approach.

Consequently, for applications with a specific composition of different objectives, we can easily adjust the mixing weights in Eq. (7) appropriately, hinted by Figs. 3&4. Additionally, although in experiments we only consider the recommendation task to a single user, the extension to a group of users can be obtained by revising the absorbing node set to include those users. More applications and systematic comparisons with other collaborative filtering methods are expected in future. Last but not the least, our approach could be regarded as a general framework with different affects if alternative schemes are considered and designed directly in place of Eq. (7) and the transition costs therein.

7. REFERENCES

[1] G. Adomavicius and Y. Kwon. Maximizing aggregate recommendation diversity: A graph-theoretic approach. In *Proc. Workshop on Novelty and Diversity in Recommender Systems, held in conjunction with ACM RecSys*, pages 3–10, 2011.

[2] C. Anderson. *The Long Tail: Why the Future of Business is Selling Less of More.* Hyperion, 2006.

[3] M. Brand. A random walks perspective on maximizing satisfaction and profit. In *SIAM International Conference on Data Mining*, pages 12–19, 2005.

[4] K. P. Burnham and D. Anderson. *Model Selection and Multi-Model Inference*. Springer, July 2002.

[5] O. Celma. *Music Recommendation and Discovery in the Long Tail*. Springer, 2010.

[6] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proc. ACM RecSys*, pages 39–46, 2010.

[7] C. Ding, X. He, H. Zha, and H. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proc. SIAM Data Mining*, 2005.

[8] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix tri-factorizations for clustering. In *Proc. 12th ACM SIGKDD*, pages 126–135, 2006.

[9] A. Edelman, T. Arias, and S. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, 1998.

[10] D. M. Fleder and K. Hosanagar. Recommender systems and their impact on sales diversity. In *Proc. 8th ACM Conf. Electronic Commerce (EC'2007)*, pages 192–199, 2007.

[11] F. Fouss, A. Pirotte, and M. Saerens. A novel way of computing similarities between nodes of a graph, with application to collaborative recommendation. In *Proc. IEEE/WIC/ACM WI'2005*, pages 550–556, 2005.

[12] D. J. H. Garling. *Inequalities: A Journey into Linear Analysis*. Cambridge, 2007.

[13] E. Gaussier and C. Goutte. Relation between PLSA and NMF and implications. In *Proc. SIGIR*, 2005.

[14] A. Gunawardana and G. Shani. A survey of accuracy evaluation metrics of recommendation tasks. *JMLR*, 10:2935–2962, 2009.

[15] T. H. Haveliwala. Topic-sensitive pagerank. In *Proc. WWW*, pages 517–526, 2002.

[16] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[17] N. Hurley and M. Zhang. Novelty and diversity in top-n recommendation – analysis and evaluation. *ACM Trans. Internet Technol.*, 10(4):14:1–14:30, 2011.

[18] J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. Springer-Verlag, 1976.

[19] W. A. Kirk and M. A. Khamsi. *An Introduction to Metric Spaces and Fixed Point Theory*. John Wiley, New York, 2001.

[20] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

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

[22] H. Ma, M. R. Lyu, and I. King. Diversitying query suggestion results. In *Proc. AAAI*, 2010.

[23] S. M. McNee, J. Riedl, and J. A. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *Proc. CHI EA'06*, pages 1097–1101, 2006.

[24] J. Norris. *Markov Chains*. Cambridge University Press, 1997.

[25] Y.-J. Park and A. Tuzhilin. The long tail of recommender systems and how to leverage it. In *Proc. ACM RecSys*, pages 11–18, 2008.

[26] M. Steyvers and T. Griffiths. *Probabilistic Topic Models*. Lawrence Erlbaum Associates, 2007.

[27] H. Yin, B. Cui, J. Li, J. Yao, and C. Chen. Challenging the long tail recommendation. In *Proc. VLDB'2012*, volume 5, pages 896–907. 2012.

[28] J. Yoo and S. Choi. Nonnegative matrix factorization with orthogonality constraints. *Journal of Computing Science and Engineering*, 4(2):97–109, 2010.

[29] M. Zhang and N. Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *Proc. ACM RecSys*, pages 123–130, 2008.

APPENDIX

A. PROOF OF THEOREM 3

According to Theorem 1, since $f(\cdot)$ in Eq. (2) satisfies $f : \mathbb{S} \to \mathbb{S}$ and the simplex $\mathbb{S}$ is closed convex, we need to find one sufficient condition such that $f(\cdot)$ is a contraction mapping, which leads to the stability.

Based on Theorem 2, we evaluate the infinite matrix-norm $||\boldsymbol{J}||_\infty = \max_i \sum_j |J_{ij}|$, i.e., the maximum absolute row sum. Specifically, we derive

$$\sum_j |J_{ij}| \le \frac{1 + N \cdot F_S^{(t+1)}(i)}{\sum_j g^{(t)}(j)}, \quad \text{for} \quad \forall i,$$

$$\Longrightarrow ||\boldsymbol{J}||_\infty = \max_i \sum_j |J_{ij}| \le \sum_{i,j} |J_{ij}| = \frac{1+N}{\sum_j g^{(t)}(j)}, \tag{18}$$

where the first inequality is based on $|a - b| \le |a| + |b|$ with $a, b \ge 0$. To let $||\boldsymbol{J}||_\infty \le q < 1$, our target becomes to find the condition that can ensure $\sum_j g^{(t)}(j) > 1 + N$.

On the other side, if we lower-bound each one-step transition cost $C(u|i)$ and $C(i|u)$ by $C_{lb} \ge 0$, we have

$$\sum_j g^{(t)}(j) \ge 1 + N \cdot C_{lb}, \tag{19}$$

based on the definition in Eq. (2).

In consequence, a necessary condition to $||\boldsymbol{J}||_\infty \le q < 1$ is letting the lower-bound $C_{lb} \ge 1$, which completes the proof.

B. DERIVATION OF EQ. (10)

For the optimization task in Eq. (9), we denote $\mathcal{L} = 0.5 \left|\left|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{A}\boldsymbol{H}^T\right|\right|_F^2 + \lambda \sum_{i,j} A_{ij}$, the partial derivatives w.r.t. parameters $\boldsymbol{W}$, $\boldsymbol{H}$ and $\boldsymbol{A}$ are given as

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}} &= -\boldsymbol{H}\boldsymbol{A}^T\boldsymbol{X} + \boldsymbol{H}\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{A}\boldsymbol{H}^T, \\
\frac{\partial \mathcal{L}}{\partial \boldsymbol{H}} &= -\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{A} + \boldsymbol{H}\boldsymbol{A}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{A}, \\
\frac{\partial \mathcal{L}}{\partial \boldsymbol{A}} &= -\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{H} + \boldsymbol{W}^T\boldsymbol{W}\boldsymbol{A}\boldsymbol{H}^T\boldsymbol{H}.
\end{aligned}$$

Due to the orthogonality $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}$ and $\boldsymbol{H}^T\boldsymbol{H} = \boldsymbol{I}$, we employ the gradients $\nabla\boldsymbol{W}$ and $\nabla\boldsymbol{H}$ on Stiefel manifold [9]:

$$\begin{aligned}
\nabla\boldsymbol{W} &= \frac{\partial \mathcal{L}}{\partial \boldsymbol{W}} - \boldsymbol{W}\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}}\right)^T \boldsymbol{W} \\
&= \boldsymbol{W}\boldsymbol{A}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{A}^T + \boldsymbol{X}^T\boldsymbol{W}) - (\boldsymbol{X} + \boldsymbol{W}\boldsymbol{A}\boldsymbol{H}^T)\boldsymbol{H}\boldsymbol{A}^T, \\
\nabla\boldsymbol{H} &= \frac{\partial \mathcal{L}}{\partial \boldsymbol{H}} - \boldsymbol{H}\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{H}}\right)^T \boldsymbol{H} \\
&= \boldsymbol{H}\boldsymbol{A}^T\boldsymbol{W}^T(\boldsymbol{W}\boldsymbol{A} + \boldsymbol{X}\boldsymbol{H}) - (\boldsymbol{X} + \boldsymbol{W}\boldsymbol{A}\boldsymbol{H}^T)^T\boldsymbol{W}\boldsymbol{A}.
\end{aligned}$$

The above yields the multiplicative updates in Eq. (10).