Title	An Explainable Recommendation Based on Acyclic Paths in an Edge-Colored Graph
Author(s)	Chinone, Kosuke; Nakamura, Atsuyoshi
Citation	Lecture Notes in Computer Science, 13151, 40-52 https://doi.org/10.1007/978-3-030-97546-3_4
Issue Date	2022-03-19
Doc URL	http://hdl.handle.net/2115/91071
Rights	This version of the contribution has been accepted for publication, after peer review (when applicable) but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: https://doi.org/10.1007/978-3-030-97546-3_4. Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms
Type	article (author version)
File Information	ajcai2021cr.pdf

An Explainable Recommendation Based on Acyclic Paths in an Edge-Colored Graph

Kosuke Chinone and Atsuyoshi Nakamura[0000−0001−7078−8655]

Graduate School of Information Science and Technology, Hokkaido University
{chinone,atsu}@ist.hokudai.ac.jp

Abstract. We propose a novel recommendation algorithm based on acyclic paths in an edge-colored graph. In our method, all the objects including users, items to recommend, and other things usable to recommendation are represented as vertices in an edge-colored directed graph, in which edge color represents relation between the objects of its both ends. By setting each edge weight appropriately so as to reflect how much the object corresponding to its one end is preferred by people who prefer the object corresponding to its other end, the probability of an s-t path, which is defined as the product of its component edges' weights, can be regarded as preference degree of item t (item corresponding to vertex t) by user s (user corresponding to vertex s) in the context represented by the path. Given probability threshold θ, the proposed algorithm recommends user s to item t that has high sum of the probabilities of all the acyclic s-t paths whose probability is at least θ. For item t recommended to user s, the algorithm also shows high probability color sequences of those s-t paths, from which we can know main contexts of the recommendation of item t for user s. According to our experiments using real-world datasets, the recommendation performance of our method is comparable to the non-explainable state-of-the-art recommendation methods.

Keywords: recommender system · explainablity · graph algorithm

1 Introduction

Now, recommender systems are used in many web services to present items suitable for user's preference, and explainability is perceived as one of the points to be improved in them [17]. In addition to recommendation, explainable systems can also provide recommendation reasons. Explainability improves the transparency, effectiveness, trustworthiness, satisfaction, and persuasiveness of recommender systems. The classical recommendation method of user or item similarity-based collaborative filtering [1, 13] can be seen as implicitly providing explanations like "the item was purchased by users whose purchase histories are similar to that of yours" or "the item was purchased by a group of users who purchased the items you bought". Recent recommendation models with high preference-prediction accuracy, such as models that learn latent factors using matrix factorization to predict items' ratings by users [9], and deep learning models [16], however, are difficult to understand the reason for recommendation intuitively.

On the other hand, there are efforts to use various relations between users, items and other related objects, to improve the accuracy and interpretability of recommendations [15]. Such relations between objects can be represented by a graph, in which objects are corresponding to vertices and relations between objects are represented by edges between vertices of corresponding objects. Considering preference-propagation degree based on each relation in the graph, items can be recommended to users by calculating estimated preference degree of items for the users. One merit of this graph-based method is that the paths most contributing to the estimated preference degree can be shown as explanation for the recommendation. In previous work, such preference degree estimation is done by calculating the stationary distribution of the Markov process with transition probability corresponding to preference-propagation degree. The approximate calculation of this method is efficiently done by repeated matrix multiplication, but contributed paths for items with high estimated preference degree can not be obtained as by-product.

In this paper, we propose a novel recommendation algorithm based on acyclic paths in an edge-colored directed graph constructed from various relations between recommendation-related objects. In an edge-colored directed graph, an edge color represents a relation between the objects that correspond to the vertices of its both ends. Given appropriate edge propagation probabilities and threshold θ, our algorithm calculates the probability sum of all the acyclic s-t paths whose probability is at least θ. On the way of the calculation, our algorithm also calculates the probability sum of each color sequence of those s-t paths. Then, high probability items t are recommended to user s accompanied by high probability color sequences of the s-t paths, from which we can know main contexts of the recommendation of item t for user s. According to our experimental results using real-world datasets, the recommendation performance of our proposed method is comparable to the non-explainable state-of-the-art recommendation methods. By checking recommendation list with the highest probability color sequences, color sequences are confirmed to be suitable for explanation of the recommendation. Furthermore, different trends are observed between the lists with different color sequences, which indicates the possibility of using color sequences to provide the recommendation lists of different trends.

Related Work

Though the term "explainable recommendation system" has only recently begun to be used [18], such recommender systems had existed before then. User-based or item-based collaborative filtering [1, 8, 13] can be seen as a kind of explainable recommender system because they recommend items based on user-similarity or item-similarity of ratings, and such similarities can be seen as explanation.

Explicit Factor Model (EFM) [4, 19] is an explainable recommendation method that is made from Latent Factor Model (LFN) [9] by modifying it so as to have explainability. Some of recent recommendation systems using deep neural networks such as CNN [14] and RNN [5] are working on automatic generation of

explanations by natural language generation models [10]. It is said to be a difficult but important problem to explain the recommendation produced by latent factor and deep learning models [17].

Recent explainable recommendation systems use social information such as purchase histories, review information, and friendships between users and items [11, 15]. There is one that uses tripartite graphs to represent the relationship between users, items, and aspects [7], where an aspect is a feature of an item extracted from a user's review. Knowledge graphs, which contain a wealth of information about users and items, have been used to generate intuitive and more suitable explanations for recommended items [3].

2   Preliminary

An *edge-weighted edge-colored directed graph* G is a triplet $G(V, E, p)$, where V is a set of vertices, $E \subseteq V \times V \times C$ is a set of directed colored edges with a color set C, and p is an edge weight function $p : E \to (0, \infty)$. Note that multiple edges $(u, v, c_1), (u, v, c_2) \in V \times V \times C$ are allowed if their colors c_1, c_2 are different. As for an edge weight function, we only consider the function p satisfying the probability condition that $\sum_{(v,c):(u,v,c) \in E} p((u, v, c)) = 1$ for all $u \in V$. An *acyclic s-t path* in G is defined as an edge sequence $(s, v_1, c_1)(v_1, v_2, c_2) \cdots (v_{k-1}, v_k, c_k)(v_k, t, c_{k+1})$ for some $k \le |V|$ in which $s, v_1, v_2, \ldots, v_k, t$ must be different from each other. For an *s-t* path $(s, v_1, c_1)(v_1, v_2, c_2) \cdots (v_{k-1}, v_k, c_k)(v_k, t, c_{k+1})$, we call $c_1 c_2 \cdots c_{k+1}$ the *color sequence* of the *s-t* path, and define the *probability* of the *s-t* path as $p((s, v_1, c_1)) \times p((v_1, v_2, c_2)) \times \cdots \times p((v_{k-1}, v_k, c_k)) \times p((v_k, t, c_{k+1}))$.

3   Proposed Method

In this section, we propose an explainable recommendation model based on acyclic paths in an edge-colored graph. First, we represent the information used for recommendation as an edge-colored graph, and introduce our recommendation method based on acyclic paths. Then, we describe our recommendation explanation by color sequences.

3.1   Graph Construction

We construct an edge-weighted edge-colored directed graph $G(V, E, p)$ for recommendation using various information such as rating, tag, item attribute and so on. Let V be the set of all the objects that appear in the information used in recommendation, that is, V is composed of users, items, tags, attributes, etc. Each directed colored edge (u, v, c) is created if object u has a certain relation (that is represented by color c) with object v. For example, use color $\mathbf{R}$ for rating relation. Then, edge $(u, v, \mathbf{R})$ is created if user u rated item v. We are dealing with a directed graph, so we use different color for reverse relation. Thus, in the case that *rated* relation (that is represented by color $\mathbf{r}$) is considered in addition

to rating relation, edge $(v, u, \mathbf{r})$ is also created. There is a case that different colored edges from u to v are created. Assume that color $\mathbf{T}$ is used for tagging relation, edge $(u, v, \mathbf{T})$ is created if user u tagged item v with some tag, which is independently done from the creation of edge $(u, v, \mathbf{R})$ for rating relation. An edge weight function $p((u, v, c))$ is set to the value that represents how much object v is preferred if object u is preferred. Such preference propagation degree depends on the relation corresponding to the color c of the edge. So, first, we construct a function $w_c : E \cap (V \times V \times \{c\}) \to (0, \infty)$ for each color c depending on its representing relation. Then, an edge weight function p is constructed by normalizing w_c for all colors c.

3.2 Acyclic-Path-Based Recommendation

We propose a recommendation method that recommends an item t to a user s based on acyclic s-t paths on an edge-colored directed graph $G(V, E, p)$. For each user s, the sum P_t of the probabilities of all the acyclic s-t paths in G is calculated for each item t, then items t with the largest P_t are recommended. One merit of this acyclic path-based recommendation, we can show high probability path information as explanation for the recommendation. As such information, we use the color sequences of high probability acyclic s-t paths, which are relation sequences because each color is corresponding to different relation. For each item t, the sum $P_t[\mathbf{c}]$ of the probabilities of all the acyclic s-t paths in G with color sequence $\mathbf{c}$ is calculated for each appearing color sequence $\mathbf{c}$, then color sequences $\mathbf{c}$ with the largest $P_t[\mathbf{c}]$ are shown as explanation for recommendation of item t.

The largest problem of this proposed method is its high computational cost. To overcome this issue, we give up summing up the probabilities of all the acyclic s-t paths and exclude the small probability paths from the summation. Concretely, we set threshold θ and we sum up the probabilities of the acyclic s-t paths whose probabilities are at least θ. If θ is small enough, the calculated probabilities are expected to be good approximations, and the ranking by such approximated probabilities is expected to be correct.

For fixed user s, probability sum P_t^θ of all the acyclic s-t path with probability at least θ can be efficiently calculated by the depth first search of G. APBRec(s, T_s), which is shown in Algorithm 1, is an algorithm that recommends the items t in T_s with the top-k P_t^θ for user s. Furthermore, for each recommended item t, the algorithm shows the color sequences $\mathbf{c}$ with top-ℓ probability sum $P_t^\theta[\mathbf{c}]$ among those of the acyclic s-t paths as explanation, where $P_t^\theta[\mathbf{c}]$ is the sum of the probabilities of all the acyclic s-t paths in G with color sequence $\mathbf{c}$ whose probability is at least θ. The depth first search from s can be done by executing Cal_Path_Prob$(s, 1, 0)$ at Line 2. In each recursive call of Cal_Path_Prob (Line 10), one edge is appended to the current path from s.

To prevent the current path from being cyclic, the Boolean variables visited$[v]$ are set to true for all the vertices v on the current path, and the current path is not extended to the vertices with visited$[v]$ = true (Line 7). Algorithm Cal_Path_Prob(v, P, d) is given the probability P of current s-v path, and edge (v, v', c_d) is not appended to the current path unless the probability

---

Algorithm 1 APBRec(s, T_s)

Input: s: vertex for target user, T_s: set of recommendable items for user s
Output: $t_1, \ldots, t_k$: top-k recommended items in T_s
 1: Initialization: for $v \in V$
    visited[v] ← false

    recommendable[v] ← $\begin{cases} \text{true} & (v \in T_s) \\ \text{false} & (v \notin T_s) \end{cases}$

    $P_v^\theta \leftarrow 0$
 2: Cal_Path_Prob($s, 1, 0$)
 3: Calculate items $t = t_1, \ldots, t_k$ with the k largest P_t^θ
 4: Output item t_i as the ith recommendation
 5: **for** $i = 1$ to k **do**
 6:     Output color sequences $\mathbf{c} = \mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_\ell$ with ℓ largest $P_{t_i}^\theta[\mathbf{c}]$ as explanation

---

Algorithm 2 Cal_Path_Prob(v, P, d)

Input: v: current vertex, P: probability of the current s-v path, d: current path length
 1: visited[v] ← true
 2: **if** recommendable[v] = true **then**
 3:     $P_v^\theta \leftarrow P_v^\theta + P$,
 4:     **if** $P_v^\theta[c[0..d-1]]$ is not registered **then**
 5:         $P_v^\theta[c[0..d-1]] \leftarrow 0$
 6:     $P_v^\theta[c[0..d-1]] \leftarrow P_v^\theta[c[0..d-1]] + P$
 7: **for** $(v', c_d) : (v, v', c_d) \in E$ and visited[v'] = false **do**
 8:     **if** $p((v, v', c_d)) \times P \geq \theta$ **then**
 9:         $c[d] \leftarrow c_d$
10:         Cal_Path_Prob($v', p((v, v', c_d)) \times P, d + 1$)
11: visited[v] ← false

---

of the path extended by the edge, which can be calculated by $p((v, v', c_d)) \times P$, is at least θ (Line 8). The color sequence of the current path is represented by a string $\mathbf{c} = c[0]c[1] \cdots c[d-1]$, which is denoted by $c[0..d-1]$. In Algorithm Cal_Path_Prob(v, P, d), the current path probability P is added to the acyclic s-v paths' probability P_v^θ if v is a vertex corresponding to a recommendable item (Line 3). In that case, the probability P is also added to the probability $P_v^\theta[c[0..d-1]]$ of its color sequence $c[0..d-1]$ (Line 6).

3.3 Explanation by Color Sequences

As explanation for the recommendation of an item t to a user s, we use the color sequences $\mathbf{c}$ of s-t paths with high probabilities $P_t^\theta[\mathbf{c}]$. Since the item t is recommended to the user u because the probability sum P_t^θ of the s-t paths is high, showing the color sequences $\mathbf{c}$ that are contributed to high P_t^θ is reasonable explanation in the context of transparency considering the fact that $P_t^\theta = \sum_{\mathbf{c}} P_t^\theta[\mathbf{c}]$. Furthermore, color sequences are relation paths from the recommended user to the recommended item, which clarifies what his/her actions

Table 1. Dataset statistics

Dataset	#users	#items	#tags	#ratings	#taggings
Movielens 20M	138493	27278	38644	20000263	465564
Food.com Recipes and Interactions	6389	197317	532	719548	2798545

(watching, rating, tagging and so on) strongly connect to the item through what other relations between objects. For example, color sequences can reveal that the recommended movie is a movie of the same genre as a movie the user has watched in the past, or a movie watched by other users who have watched the same movie as the user watched.

4 Experiments

We conduct experiments to compare the recommendation performance of our proposed algorithm APBRec with those of the state-of-the-art methods using two real-world datasets. We also check the effectiveness of high-probability color sequences as recommendation explanation.

4.1 Experimental Setup

We set the threshold θ used in Cal_Path_Prob to 10^{-7} in our experiment. Other settings are described below.

Datasets The datasets we use in our experiments are Movielens 20M [6] and Food.com Recipes and Interactions[1], which were collected on a movie review site and a recipe aggregator site, respectively. Both the datasets are composed of rating and tagging data whose statistics are shown in Table 1. The rating scales of Movielens and Food.com[2] are 1-5 and 1-6, respectively, and every user in both the datasets has at least 20 ratings. The datasets contain tags which are words tagged to movies and recipes by users.

We randomly divide the set of ratings into 60% of the ratings for training and the other 40% for testing. We report the average results over five such random splits.

Graphs Construction We construct the following edge-colored graph $G(V, E, p)$ from two datasets. The vertex set V is the set of all the objects, that is, users, items and tags. We abuse notation and let user u, item m, tag t represent the corresponding vertices themselves. The edge set E is constructed from ratings

---

[1] https://www.kaggle.com/shuyangli94/food-com-recipes-and-user-interactions

[2] The scale of the original dataset is 0-5 and it contains users who rated less than 20 recipes. We shifted the scale by one to use the same weight function and removed users who rated less than 20 recipes.

Table 2. Colored edges in $G(V, E, p)$ for the datasets, their related information that is used for determining their weight, and representing relations. We let $r_{u,m}$ denote the rating value of item m by user u, and let $t_{u,m}$ denote the set of tags to item m tagged by user u. The notation $|\cdot|$ for set '·' means the number of elements in set '·'.

edge	rel. inf.	relation	edge	rel. inf.	relation
$(u, m, \mathbf{R})$	$r_{u,m}$	user u rated item m at $r_{u,m}$.	$(u, t, \mathbf{U})$	$|\{m \mid t \in t_{u,m}\}|$	user u tagged with tag t.
$(m, u, \mathbf{r})$	$r_{u,m}$	item m was rated at $r_{u,m}$ by user u.	$(t, u, \mathbf{u})$	$|\{m \mid t \in t_{u,m}\}|$	with tag t user u tagged.
$(u, m, \mathbf{T})$	$|t_{u,m}|$	user u tagged item m.	$(m, t, \mathbf{M})$	$|\{u \mid t \in t_{u,m}\}|$	item m was tagged with tag t.
$(m, u, \mathbf{t})$	$|t_{u,m}|$	item m was tagged by user u.	$(t, m, \mathbf{m})$	$|\{u \mid t \in t_{u,m}\}|$	with tag t item m was tagged.

and taggings as follows. For each rating of item m by user u, two colored directed edges $(u, m, \mathbf{R})$ and $(m, u, \mathbf{r})$ are created. For each tagging of item m with tag t by user u, six colored directed edges $(u, m, \mathbf{T})$, $(m, u, \mathbf{t})$, $(u, t, \mathbf{U})$, $(t, u, \mathbf{u})$, $(m, t, \mathbf{M})$ and $(t, m, \mathbf{m})$ are created. So, the color set is $C = \{\mathbf{R}, \mathbf{r}, \mathbf{T}, \mathbf{t}, \mathbf{U}, \mathbf{u}, \mathbf{M}, \mathbf{m}\}$, and the relation that each color represents is shown in Table 2.

To construct a function w_c for each color c, we apply the same normalization used for rating values in [15], which is defined as follows. Let $w_{v_1,v_2,c}$ denote the related information for edge (v_1, v_2, c) shown in the column 'rel. inf.' of Table 2. Then, the function $w_c : E \to (0, \infty)$ for color $c \in C$ is defined as

$$w_c((v_1, v_2, c)) = \frac{w'((v_1, v_2, c))}{\text{ave}(w'((v_1, \cdot, c)))},$$

where $\text{ave}(w'((v_1, \cdot, c)))$ is the average of $w'((v_1, v, c))$ over $\{v \mid (v_1, v, c) \in E\}$, and the function w' is defined as

$$w'((v_1, v_2, c)) = w'((v_2, v_1, \bar{c})) = \frac{w_{v_1,v_2,c}}{\sqrt{\sum_{v_2':(v_1,v_2',c)\in E} w_{v_1,v_2',c}} \sqrt{\sum_{v_1':(v_2,v_1',\bar{c})\in E} w_{v_1',v_2,\bar{c}}}},$$

where $\bar{c}$ is the color representing the reverse relation to the relation represented by color c, that is, $\bar{\mathbf{R}} = \mathbf{r}$, $\bar{\mathbf{T}} = \mathbf{t}$, $\bar{\mathbf{U}} = \mathbf{u}$ and $\bar{\mathbf{M}} = \mathbf{m}$. The edge weight function p is a normalized w_c so as to satisfy $\sum_{(v,c):(v_1,v,c)\in E} p((v_1, v, c)) = 1$, that is,

$$p((v_1, v_2, c)) = \frac{w_c((v_1, v_2, c))}{\sum_{(v,c):(v_1,v,c)\in E} w_c((v_1, v, c))}.$$

Evaluation Metrics We evaluate recommendation performance by nDCG (normalized Discounted Cumulative Gain) [2], Precision, Recall and F1-measure. DCG(Discounted Cumulative Gain) for recommended item ranking $m_1, m_2, \cdots$ to user u is a ranking metric that discounts the score towards the bottom, that is,

$$DCG@n = \sum_{i=1}^{n} \frac{2^{r_{u,m_i}} - 1}{\log_2(i + 1)},$$

where r_{u,m_i} is the rating value of the ith item m_i in the recommended ranking for user u, and n is the maximum rank used for the evaluation. nDCG is a normalized DCG that is the DCG divided by the IDCG (Ideal DCG), where the IDCG is the DCG for the item m ranking that is sorted in descending order of $r_{u,m}$.

Precision, Recall and F1-measure were used to measure recommendation performance for high-rated items, which are defined as the items rated at least 4 in Movielens and at least 5 in Food.com. Let M_u^{high} denote the set of items rated high by user u and let $M_u^{\text{rec}}(n)$ be the set of top n recommended items. Precision@n, Recall@n and F1@n for user u are defined as

$$\text{Precision@}n = \frac{|M_u^{\text{high}} \cap M_u^{\text{rec}}(n)|}{n}, \quad \text{Recall@}n = \frac{|M_u^{\text{high}} \cap M_u^{\text{rec}}(n)|}{|M_u^{\text{high}}|}, \text{ and}$$

$$\text{F1@}n = \frac{2\text{Precision@}n \cdot \text{Recall@}n}{\text{Precision@}n + \text{Recall@}n}.$$

They take values between 0 and 1, and the closer to 1, the better the recommendation performance is.

Comparison Methods We compare the accuracy of the proposed method with those of baseline and state-of-the-art models. We used four models, AVE, IBCF, MRH and BPR models. AVE model is a baseline model, which ranks items by the ratings averaged over all the users who rated the item. IBCF model is an item-based collaborative filtering [13], which predicts rating $r_{u,m}$ by

$$\frac{\sum_{m' \in M_u} \text{cosine}(r_{\cdot,m}, r_{\cdot,m'}) r_{u,m}}{\sum_{m' \in M_u} \text{cosine}(r_{\cdot,m}, r_{\cdot,m'})},$$

where

$$\text{cosine}(r_{\cdot,m}, r_{\cdot,m'}) = \frac{\sum_{u \in U_m \cap U_{m'}} r_{u,m} r_{u,m'}}{\sqrt{\sum_{u \in U_m \cap U_{m'}} r_{u,m}^2} \sqrt{\sum_{u \in U_m \cap U_{m'}} r_{u,m'}^2}},$$

M_u is the set of movies rated by user u, and U_m is the set of users who rated item m. BPR (Bayesian Personalized Ranking) model is a kind of matrix factorization [12], which optimizes the posterior probability of correctly ranking item pairs for each user. We use the implementation shown at http://ethen8181.github.io/machine-learning/recsys/4_bpr.html with default parameter values except the following parameters: learning_rate=0.1, n_iters=3000, n_factors=60 and batch_size=100. MRH (Music Recommendation via Hypergraph) model is a graph-based method that uses social information to make recommendations [15]. We construct a hypergraph using rating and tagging information and make a transition matrix A for Random Walks with Restarts version of MRH model. The stationary distribution $\lim_{t \to \infty} \mathbf{f}^{(t)}$ for $\mathbf{f}^{(t+1)} = \alpha A \mathbf{f}^{(t)} + (1 - \alpha)\mathbf{y}_u$ is used for recommendation to user u, where $\mathbf{y}_u$ is a query vector for user u and whose nonzero entries are the entry corresponding to u, which is set to 1, and the entries corresponding to

Table 3. Recommendation performance of the five recommendation models for two datasets by nDCG@10, Precision@10, Recall@10 and F1@10. The values are averaged over all users and 5 runs and 95% confidence intervals are parenthesized. The highest values are bolded and those which are not statistically significantly different from them are italicized.

dataset	method	nDCG	Precision	Recall	F1
	AVE	$0.4508(\pm 0.0003)$	$0.3658(\pm 0.0006)$	$0.3625(\pm 0.0002)$	$0.3106(\pm 0.0004)$
	IBCF	$\mathbf{0.7667}(\pm 0.0001)$	$\mathbf{0.6901}(\pm 0.0005)$	$\mathbf{0.5239}(\pm 0.0005)$	$\mathbf{0.4926}(\pm 0.0004)$
Movielens	MRH	$0.7354(\pm 0.0003)$	$0.6478(\pm 0.0004)$	$0.4991(\pm 0.0005)$	$0.4657(\pm 0.0004)$
	BPR	$0.6885\ (\pm 0.0047)$	$0.6123\ (\pm 0.0026)$	$0.4770(\pm 0.0016)$	$0.4419(\pm 0.0018)$
	APBRec	$0.7023\ (\pm 0.0002)$	$0.6241\ (\pm 0.0004)$	$0.4799\ (\pm 0.0003)$	$0.4474\ (\pm 0.0003)$
	AVE	$0.8328(\pm 0.0011)$	$0.8587(\pm 0.0014)$	$0.5673(\pm 0.0018)$	$0.6094(\pm 0.0017)$
	IBCF	$0.8745(\pm 0.0012)$	$\mathit{0.8785}(\pm 0.0018)$	$\mathit{0.5755}(\pm 0.0016)$	$\mathit{0.6199}(\pm 0.0018)$
Food.com	MRH	$0.8659(\pm 0.0012)$	$0.8743\ (\pm 0.0021)$	$\mathit{0.5740}(\pm 0.0014)$	$\mathit{0.6179}(\pm 0.0020)$
	BPR	$\mathbf{0.8804}(\pm 0.0008)$	$\mathbf{0.8806}(\pm 0.0018)$	$\mathbf{0.5761}(\pm 0.0011)$	$\mathbf{0.6208}(\pm 0.0015)$
	APBRec	$0.8735(\pm 0.0009)$	$\mathit{0.8777}(\pm 0.0019)$	$\mathit{0.5752}(\pm 0.0013)$	$\mathit{0.6196}(\pm 0.0018)$

v that is connected to u in the graph, which are set to $A_{u,v}$. In the experiment, the restart probability $1 - \alpha$ is set to 0.04, and the stationary distribution is approximated by $\mathbf{f}^{(t)}$ for $t = 80$.

4.2   Results

Recommendation Performance The nDCG@10, Precision@10, Recall@10 and F1@10 averaged over all users and 5 runs with 95% confidence intervals for the five methods are shown in Table 3. APBRec outperforms the baseline method AVE and has comparable performance to the other three non-explainable state-of-the-art recommendation methods.

Explainability Two users' top-10 recommendation lists by APBRec for Food.com dataset are shown in Table 4. As explanation of each recommended recipe, three highest-probability color sequences are shown. The color sequences give contexts of the recommendation and makes the recommendation more transparent. For example, color sequence "**RrR**" can be interpreted as "the recipes rated high by other users who highly rated your high-rating recipes". You can see tag's contribution to the recommendation lists. Color sequence "**Um**" is the highest ranked color sequence for five recipes recommended to user 231. Interpretation of "**Um**" can be "the recipes tagged with the same tag that you tagged to other recipes".

If the highest-probability color sequences are different, the main recommendation reasons are different. Thus, we can expect to get a different type recommendation list through filtering recommended items by the highest-probability color sequences. The recommendation list to user 1741 of Movielens dataset by APBRec in Table 5 supports this expectation. The first and last sublists are

Table 4. Top-10 recommended recipes to users 133 and 231 by APBRec in Food.com dataset. The three highest probability color sequences are also shown for each recommended recipe.

user	rank recipe	color sequence ranking		
		1st(%)	2nd(%)	3rd(%)
133	1 javanese roasted salmon and wilted spinach	**RrR**(92.3)	**TrR**(7.6)	
	2 beer	**RrR**(91.6)	**TrR**(8.3)	
	3 chilli beer damper	**RrR**(94.2)	**TrR**(5.7)	
	4 meat pinwheels	**TrR**(68.4)	**RtR**(31.5)	
	5 spicy calamari with bacon and scallions	**RrR**(86.5)	**TrR**(12.3)	**Um**(1.0)
	6 roasted green beans with garlic and pine nuts	**RrR**(91.7)	**TrR**(5.6)	**RMm**(2.5)
	7 bubble and squeak	**RrR**(92.5)	**TrR**(7.4)	
	8 tamarind sauce	**TrR**(100)		
	9 citrus morning sunrise	**TrR**(96.5)	**RrR**(3.4)	
	10 maple sugar pumpkin pie	**TmM**(68.7)	**Um**(31.2)	
231	1 baked sausage stuffed jumbo pasta shells	**Um**(63.2)	**TMm**(33.1)	**Uut**(1.2)
	2 greek spinach rice balls	**RrR**(100)		
	3 raspberry wine toaster bread	**RrR**(100)		
	4 crunchy poppy seed chicken salad	**RrR**(100)		
	5 salt rising bread	**RrR**(100)		
	6 garlic lime chicken	**Um**(62.8)	**TMm**(32.8)	**RMm**(3.1)
	7 the best mexican tortilla roll ups	**RrR**(93.7)	**TrR**(5.3)	**Um**(0.9)
	8 quick fish stew	**Um**(62.8)	**TMm**(32.8)	**RMm**(3.1)
	9 mustard herb flank steak	**Um**(63.0)	**TMm**(32.9)	**RMm**(3.2)
	10 caramel apple cupcakes	**Um**(63.5)	**TMm**(33.2)	**RMm**(3.2)

divided depending on whether the item's highest-probability color sequence is **RrR** or not. The genre distributions of the two sublists look different: genres Sci-Fi, Film-Noir, Mystery and War appear in the first sublist only while genres Children, Musical, Horror, Action, Adventure and IMAX appear in the last sublist only.

5   Conclusion

We proposed a recommendation algorithm that shows high-probability color sequences (high-contributed relation path) as explanation for recommendation. To calculate high-probability color sequences, our algorithm does depth first search on an edge-colored directed graph that represents various relations between recommendation-related objects. The probability sum of all the above-threshold-probability acyclic s-t paths are used for the recommendation score of item t for user s. According to the experiments using real-world datasets, the recommendation performance of our proposed method is comparable to non-explainable state-of-the-art recommendation methods. Color sequences are confirmed to be useful not only for recommendation explanation but also for realizing recommendation lists of various trends.

Table 5. The top 30 movies recommended to user 1741 by APBRec in Movielens dataset. The first and last sublists are divided depending on whether the item's highest-probability color sequence is **RrR** or not. The italic genres indicate sublist-uniqueness, that is, they do not appear in the other sublist.

rank movie	genres	color sequence ranking		
		1st	2nd	3rd
1 The Man Who Wouldn't Die	Crime,Drama,Thriller	**RrR**	**TrR**	**UmrR**
2 Slam Dance	Thriller	**RrR**	**TrR**	**UmrR**
3 Shepherd	*Sci-Fi*	**RrR**	**TrR**	**UmrR**
9 Dragon Age	Animation,Fantasy	**RrR**	**TrR**	**UmrR**
16 Lady on a Train	Comedy,Crime,*Film-Noir*,*Mystery*,Romance, Thriller	**RrR**	**TrR**	**UmrR**
17 Long Night's Journey Into Day	Documentary	**RrR**	**TrR**	**UmrR**
20 Jim Jefferies Fully Functional	Comedy	**RrR**	**TrR**	**UmrR**
22 Napoleon	Drama,*War*	**RrR**	**TrR**	**UmrR**
26 Big Parade	Drama,Romance,*War*	**RrR**	**TrR**	**UmrR**
27 Fawlty Towers	Comedy	**RrR**	**TrR**	**UmrR**
4 Momo	Animation,*Children*,Fantasy	**Um**	**TrR**	**UuR**
5 Houseboat	Comedy,Romance	**Um**	**TrR**	**RrR**
6 Pina	Documentary,*Musical*	**Um**	**TrR**	**UuR**
7 Alvin and the Chipmunks	*Children*,Comedy	**Um**	**RMm**	**TrR**
8 The Wolf Brigade	Animation,Fantasy,Thriller	**Um**	**RrR**	**TrR**
10 My Name Is Bruce	Comedy,*Horror*	**Um**	**TrR**	**RrR**
11 Party Monster	Comedy,Crime,Drama,Thriller	**Um**	**TrR**	**RrR**
12 Out of Africa	Drama,Romance	**Um**	**RrR**	**TtR**
13 Shattered Glass	Crime,Drama	**Um**	**TrR**	**RrR**
14 Bat	*Horror*	**Um**	**TrR**	**UuR**
15 Children of the Corn	*Horror*	**Um**	**TrR**	**UmrR**
18 V.I.P.s	Drama	**TrR**	**UmrR**	**TrRrR**
19 A Question of Faith	Drama	**Um**	**UmMm**	**TMm**
21 Double Dynamite	Comedy,*Musical*	**TrR**	**UmrR**	
23 Big Fella	Drama,*Musical*	**TrR**	**UmrR**	
24 The Guardian	*Action*,Thriller	**Um**	**TrR**	**UmrR**
25 The Killer Whale	*Action*,Drama,*Horror*,Thriller	**Um**	**TrR**	**RrR**
28 Bella	Drama,Romance	**Um**	**RrR**	**TrR**
29 How to Train Your Dragon	*Adventure*,Animation,*Children*,Fantasy,*IMAX*	**Um**	**TrR**	**UuUm**
30 Art and Craft	Documentary	**Um**	**UuR**	**TMm**

Acknowledgments

References

1. Aggarwal, C.C., Wolf, J.L., Wu, K.L., Yu, P.S.: Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 201–212 (1999)
2. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: Proceedings of the 22nd international conference on Machine learning. p. 89–96 (2005)

3. Catherine, R., Mazaitis, K., Eskenazi, M., Cohen, W.: Explainable entity-based recommendations with knowledge graphs. arXiv preprint arXiv:1707.05254 (2017)
4. Chen, X., Qin, Z., Zhang, Y., Xu, T.: Learning to rank features for recommendation over multiple categories. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 305–314 (2016)
5. Donkers, T., Loepp, B., Ziegler, J.: Sequential user-based recurrent neural network recommendations. In: Proceedings of the Eleventh ACM Conference on Recommender Systems. pp. 152–160 (2017)
6. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis) **5**(4), 1–19 (2015)
7. He, X., Chen, T., Kan, M.Y., Chen, X.: Trirank: Review-aware explainable recommendation by modeling aspects. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. pp. 1661–1670 (2015)
8. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining collaborative filtering recommendations. In: Proceedings of the 2000 ACM conference on Computer supported cooperative work. pp. 241–250 (2000)
9. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 426–434 (2008)
10. Li, P., Wang, Z., Ren, Z., Bing, L., Lam, W.: Neural rating regression with abstractive tips generation for recommendation. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 345–354 (2017)
11. Papadimitriou, A., Symeonidis, P., Manolopoulos, Y.: A generalized taxonomy of explanations styles for traditional and social recommender systems. Data Mining and Knowledge Discovery **24**(3), 555–583 (2012)
12. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012)
13. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web. pp. 285–295 (2001)
14. Seo, S., Huang, J., Yang, H., Liu, Y.: Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In: Proceedings of the eleventh ACM conference on recommender systems. pp. 297–305 (2017)
15. Tan, S., et al.: Using rich social media information for music recommendation via hypergraph model. ACM Transactions on Multimedia Computing, Communications, and Applications **7**(1), 22 (2011)
16. Wang, H., Wang, N., Yeung, D.Y.: Collaborative deep learning for recommender systems. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1235–1244 (2015)
17. Zhang, Y., Chen, X.: Explainable recommendation: A survey and new perspectives. arXiv preprint arXiv:1804.11192 (2018)
18. Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., Ma, S.: Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. pp. 83–92 (2014)
19. Zhang, Y., Zhang, M., Zhang, Y., Lai, G., Liu, Y., Zhang, H., Ma, S.: Daily-aware personalized recommendation based on feature-level time series analysis. In: Proceedings of the 24th international conference on world wide web. pp. 1373–1383 (2015)