

PRINCE: Provider-side Interpretability with Counterfactual Explanations in Recommender Systems

Azin Ghazimatin, Oana Balalau, Rishiraj Saha, Gerhard Weikum

▶ **To cite this version:**

Azin Ghazimatin, Oana Balalau, Rishiraj Saha, Gerhard Weikum. PRINCE: Provider-side Interpretability with Counterfactual Explanations in Recommender Systems. WSDM 2020 - 13th ACM International Conference on Web Search and Data Mining, Feb 2020, Houston, Texas, United States. hal-02433443

HAL Id: hal-02433443
https://inria.hal.science/hal-02433443

Submitted on 9 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

PRINCE: Provider-side Interpretability with Counterfactual Explanations in Recommender Systems

Azin Ghazimatin
Max Planck Institute for Informatics, Germany
aghazima@mpi-inf.mpg.de

Oana Balalau*
Inria and École Polytechnique, France
oana.balalau@inria.fr

Rishiraj Saha Roy
Max Planck Institute for Informatics, Germany
rishiraj@mpi-inf.mpg.de

Gerhard Weikum
Max Planck Institute for Informatics, Germany
weikum@mpi-inf.mpg.de

ABSTRACT

Interpretable explanations for recommender systems and other machine learning models are crucial to gain user trust. Prior works that have focused on paths connecting users and items in a heterogeneous network have several limitations, such as discovering relationships rather than true explanations, or disregarding other users' privacy. In this work, we take a fresh perspective, and present PRINCE: a provider-side mechanism to produce tangible explanations for end-users, where an explanation is defined to be *a set of minimal actions performed by the user* that, if removed, changes the recommendation to a different item. Given a recommendation, PRINCE uses a polynomial-time optimal algorithm for finding this minimal set of a user's actions from an exponential search space, based on random walks over dynamic graphs. Experiments on two real-world datasets show that PRINCE provides more compact explanations than intuitive baselines, and insights from a crowdsourced user-study demonstrate the viability of such action-based explanations. We thus posit that PRINCE produces *scrutable*, *actionable*, and *concise explanations*, owing to its use of *counterfactual* evidence, a user's *own actions*, and *minimal* sets, respectively.

1 INTRODUCTION

Motivation. Providing user-comprehensible explanations for machine learning models has gained prominence in multiple communities [33, 39, 56, 59]. Several studies have shown that explanations increase users' trust in systems that generate personalized recommendations or other rankings (in news, entertainment, etc.) [26, 27, 38]. Recommenders have become very sophisticated, exploiting signals from a complex interplay of factors like users' activities, interests and social links [57]. Hence the pressing need for explanations.

Explanations for recommenders can take several forms, depending on the generator (*explanations by whom?*) and the consumer (*explanations for whom?*). As generators, only *service providers* can produce true explanations for how systems compute the recommended items [6, 47, 58]; *third parties* can merely discover relationships and create post-hoc rationalizations for black-box models that may look convincing to users [18, 37, 48]. On the consumer side, *end-users* can grasp tangible aspects like activities, likes/dislikes/ratings or demographic factors. Unlike system developers or accountability engineers, end-users would obtain hardly any insight from transparency of internal system workings. In this work, we deal with explanations *by the provider and for the end-user.*

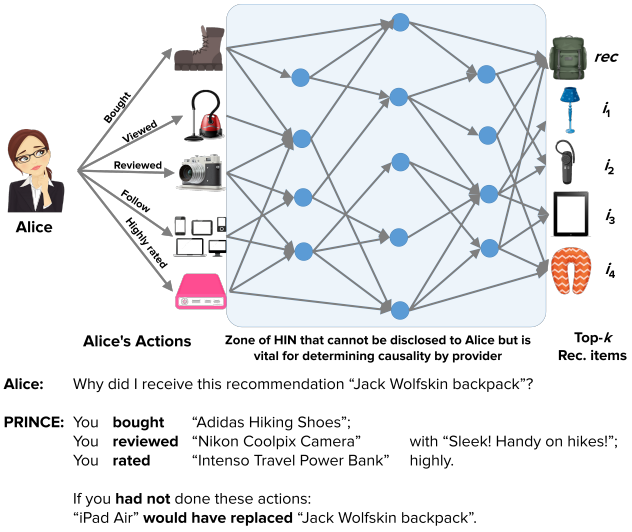


Figure 1: PRINCE generates explanations as a minimal set of actions using counterfactual evidence on user-specific HINs.

Limitations of state-of-the-art. At the core of most recommender systems is some variant of matrix or tensor decomposition (e.g., [25]) or spectral graph analysis (e.g., [21]), with various forms of regularization and often involving gradient-descent methods for parameter learning. One of the recent and popular paradigms is based on *heterogeneous information networks (HIN)* [42, 52–54], a powerful model that represents relevant entities and actions as a directed and weighted graph with multiple node and edge types. Prior efforts towards explanations for HIN-based recommendations have mostly focused on *paths* that connect the user with the recommended item [1, 18, 43, 46, 49–51]. An application of path-based explanations, for an online shop, would be of the form:

> User u received item *rec* because u follows user v, who bought item j, which has the same category as *rec*.

*This work was done while the author was at the MPI for Informatics.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Conference'17, July 2017, Washington, DC, USA
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

However, such methods come with critical privacy concerns arising from nodes in paths that disclose other users' actions or interests to user u, like the purchase of user v above. Even if user v's id was anonymized, user u would know whom she is following and could often guess who user v actually is, that bought item j, assuming that u has a relatively small set of followees [31]. If entire paths containing other users are suppressed instead, then such explanations would no longer be faithful to the true cause. Another family of path-based methods [18, 37, 48] presents plausible connections between users and items as justifications. However, this is merely post-hoc rationalization, and not actual causality.

Approach. This paper presents PRINCE, a method for **Pr**ovider-side **In**terpretability with **C**ounterfactual **E**vidence, that overcomes the outlined limitations. PRINCE is a provider-side solution aimed at detecting the actual cause responsible for the recommendation, in a heterogeneous information network with users, items, reviews, and categories. PRINCE's explanations are grounded in the user's own actions, and thus preclude privacy concerns of path-based models. Fig. 1 shows an illustrative example. Here, Alice's actions like bought shoes, reviewed a camera, and rated a power bank are deemed as explanations for her backpack recommendation. One way of identifying a user's actions for an explanation would be to compute scores of actions with regard to the recommended item. However, this would be an unwieldy distribution over potentially hundreds of actions – hardly comprehensible to an end-user. Instead, we operate in a *counterfactual* setup [32]. PRINCE identifies a small (and actually minimal) set of a user's actions such that removing these actions would result in replacing the recommended item with a different item. In Fig. 1, the item *rec* = "Jack Wolfskin backpack" would be replaced, as the system's top recommendation, by i_3 ="iPad Air" (the i's represent candidate replacement items). Note that there may be multiple such minimal sets, but uniqueness is not a concern here.

Another perspective here is that the goal of an explanation is often to show users *what they can do* in order to receive more relevant recommendations. Under this claim, the end-user has no *control* on the network beyond her immediate neighborhood, i.e., the network beyond is *not actionable* (shaded zone in Fig. 1), motivating PRINCE's choice of grounding explanations in users' own actions.

For true explanations, we need to commit ourselves to a specific family of recommender models. In this work, we choose a general framework based on *Personalized PageRank (PPR)*, as used in the state-of-the-art RecWalk system [35], and adapt it to the HIN setup. The heart of PRINCE is a polynomial-time algorithm for exploring the (potentially exponential) search space of subsets of user actions – the candidates for causing the recommendation. The algorithm efficiently computes PPR contributions for groups of actions with regard to an item, by adapting the reverse local push algorithm of [2] to a dynamic graph setting [55]. In summary, the desiderata for the explanations from PRINCE (in **bold**) connect to the technical approaches adopted (in *italics*) in the following ways. Our explanations are:

- **Scrutable**, as they are derived in a *counterfactual* setup;
- **Actionable**, as they are grounded in the user's own *actions*;
- **Concise**, as they are *minimal* sets changing a recommendation.

Extensive experiments with Amazon and Goodreads datasets show that PRINCE's minimal explanations, achieving the desired item-replacement effect, cannot be easily obtained by heuristic methods based on contribution scores and shortest paths. A crowd-sourced user study on Amazon Mechanical Turk (AMT) provides additional evidence that PRINCE's explanations are more useful than ones based on paths [51]. Our code is public at https://github.com/azinmatin/prince/.

Contributions. Our salient contributions in this work are:

- PRINCE is the first work that explores counterfactual evidence for discovering causal explanations in a heterogeneous information network;
- PRINCE is the first work that defines explanations for recommenders in terms of users' own actions;
- We present an optimal algorithm that explores the search space of action subsets in polynomial time, for efficient computation of a minimal subset of user actions;
- Experiments with two large datasets and a user study show that PRINCE can effectively aid a service provider in generating user-comprehensible causal explanations for recommended items.

2 COMPUTATIONAL MODEL

Heterogeneous Information Networks (HIN). A *heterogeneous graph* $G = (V, E, \theta)$ consists of a set of nodes V, a set of edges $E \subseteq V \times V$, and a mapping θ from each node and each edge to their types, such that $\theta_V : V \to T_V$ and $\theta_E : E \to T_E$ with $|T_V| + |T_E| > 2$. In our work, a heterogenous graph contains at least two node types, users $U \in T_V$ and items $I \in T_V$. For simplicity, we use the notations U and I to refer both to the type of a node and the set of all nodes of that type. A graph is *weighted* if there is a weight assigned to each edge, $w : E \to \mathbb{R}$, and a graph is *directed* if E is a set of ordered pairs of nodes. We denote with $N_{out}(v)$ and $N_{in}(v)$ the sets of out-neighbors and in-neighbors of node v, respectively. A directed and weighted heterogeneous graph where each node $v \in V$ and each edge $e \in E$ belong to exactly one type, is called a *heterogenous information network (HIN)* [42].

Personalized PageRank (PPR) for recommenders. We use Personalized PageRank (PPR) for recommendation in HINs [19, 35]. PPR is the stationary distribution of a random walk in G in which, at a given step, with probability α, a surfer teleports to a set of seed nodes $\{s\}$, and with probability $1 - \alpha$, continues the walk to a randomly chosen outgoing edge from the current node. More precisely, given G, teleportation probability α, a single seed s, the one-hot vector e_s, and the transition matrix W, the Personalized PageRank vector $PPR(s)$ is defined recursively as:

$$PPR(s, \cdot) = \alpha e_s + (1 - \alpha) PPR(s, \cdot) W \tag{1}$$

Let $PPR(s, v)$ be the PPR score of node v personalized for s. We define the *PPR recommendation* for user $u \in U$, or the top-1 recommendation, as:

$$rec = \underset{i \in I \setminus N_{out}(u)}{\arg\max} \; PPR(u, i) \tag{2}$$

Given a set of edges $A \subset E$, we use the notation $PPR(u, i|A)$ to define the PPR of an item i personalized for a user u in the graph $G = (V, E \setminus A, \theta)$. We refer to this graph as $G \setminus A$. To improve top-n

recommendations, Nikolakopoulos et al. [35] define a random walk in an HIN G as follows:

- With probability α, the surfer teleports to u
- With probability $1-\alpha$, the surfer continues the walk in the following manner:
  - With probability $1-\beta$, the random surfer moves to a *node of the same type*, using a similarity-based stochastic transition matrix
  - With probability β, the surfer chooses any outgoing edge at random.

For each node type t in T_V, there is an associated stochastic similarity matrix S_t, which encodes the relationship between the nodes of type t. When nodes of the same type are not comparable, the similarity matrix is the identity matrix, i.e. $S_t = I$. Otherwise, an entry (i, j) in S_t corresponds to the similarity between node i and node j. The stochastic process described by this walk is a nearly uncoupled Markov chain [35]. The stationary distribution of the random walk is the PPR with teleportation probability α in a graph G^β (referred to as *RecWalk* in [35]), where the transition probability matrix of G^β is:

$$W^\beta = \beta W + (1-\beta)S \tag{3}$$

The matrix W is the transition probability matrix of the original graph G. Matrix $S = Diag(S_1, S_2, \cdots, S_{|T_V|})$ is a diagonal matrix of order $|V|$.

Counterfactual Explanations. A user u interacts with items via different types of *actions* A, such as clicks, purchases, ratings or reviews, which are captured as *interaction edges* in the graph G. Our goal is to present user u with a set of interaction edges $A^* \subseteq \{(u, n_i) | (u, n_i) \in A\}$ (where n_i is a neighbor of u) responsible for an item recommendation rec; we refer to this as a *counterfactual explanation*. An explanation is *counterfactual*, if after removing the edges A^* from the graph, the user receives a different top-ranked recommendation rec^*. A counterfactual explanation A^* is *minimal* if there is no smaller set $A' \subseteq A$ such that $|A'| < |A^*|$ and A' is also a counterfactual explanation for rec.

Formal problem statement. Given a heterogenous information network $G = (V, E, \theta)$ and the top-ranked recommendation $rec \in I$ for user $u \in U$, find a minimum counterfactual explanation for rec.

3 THE PRINCE ALGORITHM

In this section, we develop an algorithm for computing a minimum counterfactual explanation for user u receiving recommended item rec, given the PPR-based recommender framework *RecWalk* [35]. A naïve optimal algorithm enumerates all subsets of actions $A^* \subseteq A$, and checks whether the removal of each of these subsets replaces rec with a different item as the top recommendation, and finally selects the subset with the minimum size. This approach is exponential in the number of actions of the user.

To devise a more efficient and practically viable algorithm, we express the *PPR* scores as follows [22], with $PPR(u, rec)$ denoting the PPR of rec personalized for u (i.e., jumping back to u):

$$PPR(u, rec) = (1-\alpha) \sum_{n_i \in N_{out}(u)} W(u, n_i) PPR(n_i, rec) + \alpha \delta_{u, rec} \tag{4}$$

Algorithm 1: PRINCE

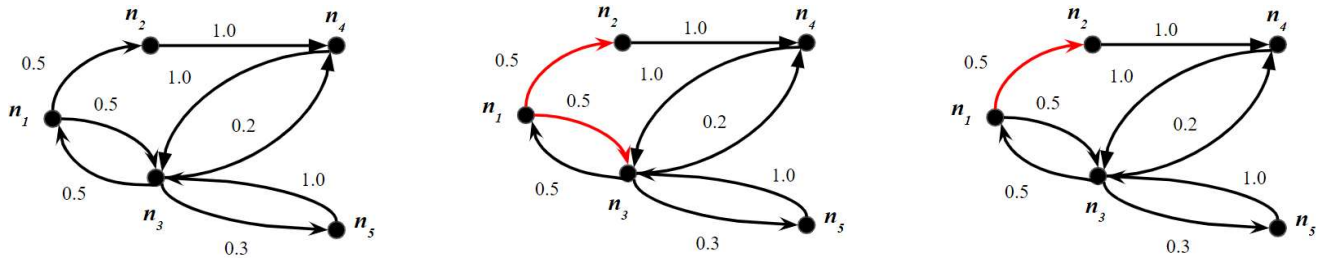
Input: $G = (V, E, \theta)$, $I \subset V$, $u \in V$, $rec \in I$
Output: A^* for (u, rec)

$A^* \leftarrow A$
$rec^* \leftarrow rec$
foreach $i \in I$ **do**
  $A^i \leftarrow$ SwapOrder(G, u, rec, i)
  // Actions A^i swap orders of rec and i
  if $|A^i| < |A^*|$ **then**
    $A^* \leftarrow A^i$
    $rec^* \leftarrow i$
  end
  else if $|A^i| = |A^*|$ *and* $PPR(u, i|A^i) > PPR(u, rec^*|A^i)$ **then**
    $A^* \leftarrow A^i$
    $rec^* \leftarrow i$
  end
end
return A^*, rec^*
Function SwapOrder(G, u, rec, rec^*):
  $A \leftarrow \{(u, n_i) | n_i \in N_{out}(u), n_i \neq u\}$
  $A^* \leftarrow \emptyset$
  $H \leftarrow$ MaxHeap(ϕ)
  $sum \leftarrow 0$
  foreach $(u, n_i) \in A$ **do**
    $diff \leftarrow W(u, n_i) \cdot (PPR(n_i, rec|A) - PPR(n_i, rec^*|A))$
    $H.insert(n_i, diff)$
    $sum \leftarrow sum + diff$
  end
  while $sum > 0$ *and* $|H| > 0$ **do**
    $(n_i, diff) \leftarrow H.delete_max()$
    $sum \leftarrow sum - diff$
    $A^* \leftarrow A^* \cup (u, n_i)$
  end
  if $sum > 0$ **then** $A^* \leftarrow A$
  return A^*
end

where α denotes the teleportation probability (probability of jumping back to u) and δ is the Kronecker delta function. The only required modification, with regard to *RecWalk* [35], is the transformation of the transition probability matrix from W to W^β. For simplicity, we will refer to the adjusted probability matrix as W.

Eq. 4 shows that the PPR of rec personalized for user u, $PPR(u, rec)$, is a function of the PPR values of rec personalized for the neighbors of u. Hence, in order to decrease $PPR(u, rec)$, we can remove edges $(u, n_i), n_i \in N_{out}(u)$. To replace the recommendation rec with a different item rec^*, a simple heuristic would remove edges (u, n_i) in non-increasing order of their contributions $W(u, n_i) \cdot PPR(n_i, rec)$. However, although this would reduce the PPR of rec, it also affects and possibly reduces the PPR of other items, too, due to the recursive nature of PPR, where all paths matter.

Let A be the set of outgoing edges of a user u and let A^* be a subset of A, such that $A^* \subseteq A$. The main intuition behind our algorithm is that we can express $PPR(u, rec)$ after the removal of A^*, denoted by $PPR(u, rec|A^*)$, as a function of two components: $PPR(u, u|A^*)$ and the values $PPR(n_i, rec|A)$, where $n_i \in \{n_i | (u, n_i) \in A \setminus A^*\}$ and



(a) $PPR(n_1, n_4) = 0.160$
$PPR(n_1, n_5) = 0.085$
$PPR(n_1, n_4) > PPR(n_1, n_5)$

(b) $A = \{(n_1, n_2), (n_1, n_3)\}$
$W(n_1, n_2)[PPR(n_2, n_4|A) - PPR(n_2, n_5|A)] = 0.095$
$W(n_1, n_3)[PPR(n_3, n_4|A) - PPR(n_3, n_5|A)] = -0.022$

(c) $A^* = \{(n_1, n_2)\}$
$PPR(n_1, n_4|A^*) = 0.078$
$PPR(n_1, n_5|A^*) = 0.110$
$PPR(n_1, n_5|A^*) > PPR(n_1, n_4|A^*)$

Figure 2: Toy Example. (a) A weighted and directed graph where the *PPR* scores are personalized for node n_1. Node n_4 has higher *PPR* than n_5. (b) Scores in a graph configuration where outgoing edges (n_1, n_2), and (n_1, n_3) are removed (marked in red). (c) Removing (n_1, n_2) causes n_5 to outrank n_4.

$n_i \neq u$. The score $PPR(u, u|A^*)$ does not depend on *rec*, and the score $PPR(n_i, rec|A)$ is independent of A^*.

Based on these considerations, we present Algorithm 1, proving its correctness in Sec. 4. Algorithm 1 takes as input a graph G, a user u, a recommendation *rec*, and a set of items I. In lines 3-13, we iterate through the items I, and find the minimum counterfactual explanation A^*. Here, A^i refers to the actions whose removal swaps the orders of items *rec* and i. In addition, we ensure that after removing A^*, we return the item with the highest PPR score as the replacement item (lines 9-11). Note that in the next section, we propose an equivalent formulation for the condition $PPR(u, i|A^i) > PPR(u, rec^*|A^i)$, eliminating the need for recomputing scores in $G \setminus A^*$.

The core of our algorithm is the function `SwapOrder`, which receives as input two items, *rec* and rec^*, and a user u. In lines 20-24, we sort the interaction edges $(u, n_i) \in A$ in non-increasing order of their contributions $W(u, n_i) \cdot (PPR(n_i, rec|A) - PPR(n_i, rec^*|A))$. In lines 25-29, we remove at each step, the outgoing interaction edge with the highest contribution, and update S and A^* correspondingly. The variable S is strictly positive if in the current graph configuration $(G \setminus A^*)$, $PPR(u, rec) > PPR(u, rec^*)$. This constitutes the main building block of our approach. Fig. 2 illustrates the execution of Algorithm 1 on a toy example.

The time complexity of the algorithm is $O(|I| \times |A| \times \log |A|)$, plus the cost of computing PPR for these nodes. The key to avoiding the exponential cost of considering all subsets of A is the insight that *we need only to compute PPR values for alternative items with personalization based on a graph where all user actions A are removed.* This is feasible because the action deletions affect only outgoing edges of the teleportation target u, as elaborated in Sec. 4.

The PPR computation could simply re-run a power-iteration algorithm for the entire graph, or compute the principal eigenvector for the underlying matrix. This could be cubic in the graph size (e.g., if we use full-fledged SVD), but it keeps us in the regime of polynomial runtimes. In our experiments, we use the much more efficient reverse local push algorithm [2] for PPR calculations.

4 CORRECTNESS PROOF

We prove two main results:

(i) $PPR(u, rec|A^*)$ can be computed as a product of two components where one depends on the modified graph with the edge set $E \setminus A$ (i.e., removing all user actions) and the other depends on the choice of A^* but not on the choice of *rec*.

(ii) To determine if some A^* replaces the top node *rec* with a different node rec^* which is not an out-neighbor of u, we need to compute only the first of the two components in (i).

Theorem 4.1. *Given a graph $G = (V, E)$, a node u with outgoing edges A such that $(u, u) \notin A$, a set of edges $A^* \subset A$, a node $rec \notin N_{out}(u)$, the PPR of rec personalized for u in the modified graph $G^* = (V, E \setminus A^*)$ can be expressed as follows:*

$$PPR(u, rec|A^*) = PPR(u, u|A^*) \cdot f\left(\left\{PPR(n_i, rec|A) \middle| (u, n_i) \in A \setminus A^*\right\}\right)$$

where $f(\cdot)$ is an aggregation function.

Proof. Assuming that each node has at least one outgoing edge, the PPR can be expressed as the sum over the probabilities of walks of length l starting at a node u [3]:

$$PPR(u, \cdot) = \alpha \sum_{l=0}^{\infty} (1 - \alpha)^l e_u W^l \tag{5}$$

where e_u is the one-hot vector for u. To analyze the effect of deleting A^*, we split the walks from u to *rec* into two parts, (i) the part representing the sum over probabilities of walks that start at u and pass again by u, which is equivalent to $\alpha^{-1} PPR(u, u|A^*)$ (division by α is required as the walk does not stop at u), and (ii) the part representing the sum over probabilities of walks starting at node u and ending at *rec* without revisiting u again, denoted by $p_{-u}(u, rec|A^*)$. Combining these constituent parts, PPR can be stated as follows:

$$PPR(u, rec|A^*) = \alpha^{-1} PPR(u, u|A^*) \cdot p_{-u}(u, rec|A^*) \tag{6}$$

As stated previously, $p_{-u}(u, rec|A^*)$ represents the sum over the probabilities of the walks from u to *rec* without revisiting u. We can express these walks using the remaining neighbors of u after

removing A^*:

$$p_{-u}(u, rec|A^*) = (1-\alpha) \sum_{(u,n_i)\in A\setminus A^*} W(u,n_i) \cdot p_{-u}(n_i, rec|A^*) \quad (7)$$

where $p_{-u}(n_i, rec|A^*)$ refers to the walks starting at n_i ($n_i \neq u$) and ending at rec that do not visit u. We replace $p_{-u}(n_i, rec|A^*)$ with its equivalent formulation $PPR(n_i, rec|A)$. $PPR(n_i, rec)$ in graph $G \setminus A$ is computed as the sum over the probabilities of walks that never pass by u. Eq. 6 can be rewritten as follows:

$$\begin{aligned} &PPR(u, rec|A^*) \\ &= PPR(u,u|A^*) \cdot \alpha^{-1}(1-\alpha) \sum_{(u,n_i)\in A\setminus A^*} W(u,n_i|A^*) PPR(n_i, rec|A) \end{aligned} \quad (8)$$

This equation directly implies:

$$PPR(u, rec|A^*) = PPR(u,u|A^*) \cdot f\left(\left\{PPR(n_i, rec|A) \middle| (u,n_i) \in A\setminus A^*\right\}\right) \quad (9)$$

□

Theorem 4.2. *The minimum counterfactual explanation for* (u, rec) *can be computed in polynomial time.*

Proof. We show that there exists a polynomial-time algorithm for finding the minimum set $A^* \subset A$ such that $PPR(u, rec|A^*) < PPR(u, rec^*|A^*)$, if such a set exists. Using Theorem 4.1, we show that one can compute if some rec^* can replace the original rec as the top recommendation, solely based on PPR scores from a single graph where all user actions A are removed:

$$\begin{aligned} &PPR(u, rec|A^*) < PPR(u, rec^*|A^*) \\ &\Leftrightarrow \sum_{(u,n_i)\in A\setminus A^*} W(u,n_i|A^*)\big(PPR(n_i, rec|A) - PPR(n_i, rec^*|A)\big) < 0 \\ &\Leftrightarrow \sum_{(u,n_i)\in A\setminus A^*} W(u,n_i)\big(PPR(n_i, rec|A) - PPR(n_i, rec^*|A)\big) < 0 \end{aligned} \quad (10)$$

The last equivalence is derived from:

$$W(u,n_i|A^*) = \frac{W(u,n_i)}{1 - \sum_{(u,n_j)\in A^*} W(u,n_j)} \quad (11)$$

For a fixed choice of rec^*, the summands in expression 10 do not depend on A^*, and so they are constants for all possible choices of A^*. Therefore, by sorting the summands in descending order, we can greedily expand A^* from a single action to many actions until some rec^* outranks rec. This approach is then guaranteed to arrive at a minimum subset.

□

5 GRAPH EXPERIMENTS

We now describe experiments performed with graph-based recommenders built from real datasets to evaluate Prince.

5.1 Setup

Datasets. We used two real datasets:

(i) The Amazon Customer Review dataset (released by Amazon: s3.amazonaws.com/amazon-reviews-pds/readme.html), and,

(ii) The Goodreads review dataset (crawled by the authors of [45]: sites.google.com/eng.ucsd.edu/ucsdbookgraph/home).

Dataset	#Users	#Items	#Reviews	#Categories	#Actions
Amazon	2k	54k	58k	43	114k
Goodreads	1k	17k	20k	16	45k

Table 1: Properties of the Amazon and Goodreads samples.

Each record in both datasets consists of a user, an item, its categories, a review, and a rating value (on a $1-5$ scale). In addition, a Goodreads data record has the book author(s) and the book description. We augmented the Goodreads collection with social links (users following users) that we crawled from the Goodreads website.

The high diversity of categories in the Amazon data, ranging from household equipment to food and toys, allows scope to examine the interplay of cross-category information within explanations. The key reason for additionally choosing Goodreads is to include the effect of social connections (absent in the Amazon data). The datasets were converted to graphs with "users", "items", "categories", and "reviews" as *nodes*, and "rated" (user-item), "reviewed" (user-item), "has-review" (item-review), "belongs-to" (item-category) and "follows" (user-user) as *edges*. In Goodreads, there is an additional node type "author" and an edge type "has-author" (item-author). All the edges, except the ones with type "follows", are bidirectional. Only ratings with value *higher than three* were considered, as low-rated items should not influence further recommendations.

Sampling. For our experiments, we sampled 500 seed users who had between 10 and 100 actions, from both Amazon and Goodreads datasets. The filters served to prune out under-active and power users (potentially bots). Activity graphs were constructed for the sampled users by taking their four-hop neighborhood from the sampled data (Table 1). Four is a reasonably small radius to keep the items relevant and personalized to the seed users. On average, this resulted in having about 29k items and 16k items for each user in their HIN, for Amazon and Goodreads, respectively.

The graphs were augmented with weighted edges for node similarity. For Amazon, we added review-review edges where weights were computed using the cosine similarity of the review embeddings, generated with Google's Universal Sentence Encoder [7], with a cut-off threshold $\tau = 0.85$ to retain only confident pairs. This resulted in 194 review-review edges. For Goodreads, we added three types of similarity edges: category-category, book-book and review-review, with the same similarity measure (24 category-category, 113 book-book, and 1003 review-review edges). Corresponding thresholds were $0.67, 0.85$ and 0.95. We crawled category descriptions from the Goodreads' website and used book descriptions and review texts from the raw data. Table 1 gives some statistics about the sampled datasets.

Initialization. The replacement item for rec is always chosen from the original top-k recommendations generated by the system; we systematically investigate the effect of k on the size of explanations in our experiments (with a default $k = 5$). Prince does not need to be restricted to an explicitly specified candidate set, and can actually operate over the full space of items I. In practice, however, replacement items need to be guided by some measure of relevance to the user, or item-item similarity, so as not to produce degenerate or trivial explanations if rec is replaced by some arbitrary item from a pool of thousands.

We use the standard teleportation probability $\alpha = 0.15$ [**?**]. The parameter β is set to 0.5. To compute PPR scores, we used the

	Amazon			Goodreads		
k	PRINCE	*HC*	*SP*	PRINCE	*HC*	*SP*
3	**5.09***	6.87	7.57	**2.05***	2.86	5.38
5	**3.41***	4.62	5.01	**1.66***	2.19	4.37
10	**2.66***	3.66	4.15	**1.43**	1.45	3.28
15	**2.13***	3.00	3.68	**1.11**	1.12	2.90
20	**1.80***	2.39	3.28	**1.11**	1.12	2.90

Table 2: Average sizes of counterfactual explanations. The best value per row in a dataset is in bold. An asterisk (*) indicates statistical significance of PRINCE over the closest baseline, under the 1-tailed paired t-test at $p < 0.05$.

Parameter	Amazon		Goodreads	
	Pre-comp	Dynamic	Pre-comp	Dynamic
$k = 3$	0.3ms	39.1s	0.3ms	24.1s
$k = 5$	0.6ms	60.4s	0.4ms	34.7s
$k = 10$	1.3ms	121.6s	0.9ms	60.7s
$k = 15$	2.0ms	169.3s	1.5ms	91.6s
$k = 20$	2.6ms	224.4s	2ms	118.8s
$\beta = 0.01$	0.4ms	1.1s	0.3ms	2.9s
$\beta = 0.1$	0.5ms	15.5s	0.3ms	8.9s
$\beta = 0.3$	0.5ms	17.0s	0.4ms	12.5s
$\beta = 0.5$	0.6ms	60.5s	0.4ms	34.7s

Table 3: Average runtime of PRINCE, when the scores are precomputed (Pre-comp) and when the scores are dynamically computed using the reverse push algorithm [55] (Dynamic).

reverse local push method [55] with $\epsilon = 1.7e-08$ for Amazon and $\epsilon = 2.7e-08$ for Goodreads. With these settings, PRINCE and the baselines were executed on all 500 user-specific HINs to compute an alternative recommendation (i.e., replacement item) rec^* and a counterfactual explanation set A^*.

Baselines. Since PRINCE is an optimal algorithm with correctness guarantees, it always finds minimal sets of actions that replace rec (if they exist). We wanted to investigate, to what extent other, more heuristic, methods approximate the same effects. To this end, we compared PRINCE against two natural baselines:

(i) *Highest Contributions* (*HC*): This is analogous to counterfactual evidence in feature-based classifiers for structured data [9, 34]. It defines the contribution score of a user action (u, n_i) to the recommendation score $PPR(u, rec)$ as $PPR(n_i, rec)$ (Eq. 4), and iteratively deletes edges with highest contributions until the highest-ranked rec changes to a different item.

(ii) *Shortest Paths* (*SP*): *SP* computes the shortest path from u to rec and deletes the first edge (u, n_i) on this path. This step is repeated on the modified graph, until the top-ranked rec changes to a different item.

Evaluation Metric. The metric for assessing the quality of an explanation is its size, that is, the number of actions in A^* for PRINCE, and the number of edges deleted in *HC* and *SP*.

Method	Explanation for "Baby stroller" with category "Baby" [Amazon]
PRINCE	**Action 1:** You rated highly "Badger Basket Storage Cubby" with category "Baby" **Replacement Item:** "Google Chromecast HDMI Streaming Media Player" with categories "Home Entertainment"
HC	**Action 1:** You rated highly "Men's hair paste" with category "Beauty" **Action 2:** You reviewed "Men's hair paste" with category "Beauty" with text "Good product. Great price." **Action 3:** You rated highly "Badger Basket Storage Cubby" with category "Baby" **Action 4:** You rated highly "Straw bottle" with category "Baby" **Action 5:** You rated highly "3 Sprouts Storage Caddy" with category "Baby" **Replacement Item:** "Bathtub Waste And Overflow Plate" with categories "Home Improvement"
SP	**Action 1:** You rated highly "Men's hair paste" with category "Beauty" **Action 2:** You rated highly "Badger Basket Storage Cubby" with category "Baby" **Action 3:** You rated highly "Straw bottle" with category "Baby" **Action 4:** You rated highly "3 Sprouts Storage Caddy" with category "Baby" **Replacement Item:** "Google Chromecast HDMI Streaming Media Player" with categories "Home Entertainment"

Method	Explanation for "The Multiversity" with categories "Comics, Historical-fiction, Biography, Mystery" [Goodreads]
PRINCE	**Action 1:** You rated highly "Blackest Night" with categories "Comics, Fantasy, Mystery, Thriller" **Action 2:** You rated highly "Green Lantern" with categories "Comics, Fantasy, Children" **Replacement item:** "True Patriot: Heroes of the Great White North" with categories "Comics, Fiction"
HC	**Action 1:** You follow User ID x **Action 2:** You rated highly "Blackest Night" with categories "Comics, Fantasy, Mystery, Thriller" **Action 3:** You rated highly "Green Lantern" with categories "Comics, Fantasy, Children" **Replacement item:** "The Lovecraft Anthology: Volume 2" with categories "Comics, Crime, Fiction"
SP	**Action 1:** You follow User ID x **Action 2:** You rated highly "Fahrenheit 451" with categories "Fantasy, Young-adult, Fiction" **Action 3:** You rated highly "Darkly Dreaming Dexter (Dexter, #1)" with categories "Mystery, Crime, Fantasy" **And 6 more actions** **Replacement item:** "The Lovecraft Anthology: Volume 2" with categories "Comics, Crime, Fiction"

Table 4: Anecdotal examples of explanations by PRINCE and the counterfactual baselines.

5.2 Results and Insights

We present our main results in Table 2 and discuss insights below. These comparisons were performed for different values of the parameter k. Wherever applicable, statistical significance was tested under the 1-tailed paired t-test at $p < 0.05$. Anecdotal examples of explanations by PRINCE and the baselines are given in Table 4. In the Amazon example, we observe that our method produces a topically coherent explanation, with both the recommendation and the explanation items in the same category. The *SP* and *HC* methods give larger explanations, but with poorer quality, as the first action in both methods seems unrelated to the recommendation. In the Goodreads example, both *HC* and *SP* yield the same replacement item, which is different from that of PRINCE.

Approximating PRINCE is difficult. Explanations generated by PRINCE are more concise and hence more user-comprehensible than those by the baselines. This advantage is quite pronounced; for example, in Amazon, all the baselines yield at least one more action in the explanation set on average. Note that this translates into unnecessary effort for users who want to act upon the explanations.

Explanations shrink with increasing *k*. The size of explanations shrinks as the top-k candidate set for choosing the replacement item is expanded. For example, the explanation size for PRINCE on Amazon drops from 5.09 at $k = 3$ to 1.80 at $k = 20$. This is due to the fact that with a growing candidate set, it becomes easier to find an item that can outrank *rec*.

PRINCE is efficient. To generate a counterfactual explanation, PRINCE only relies on the scores in the graph configuration $G \setminus A$ (where all the outgoing edges of u are deleted). Pre-computing $PPR(n_i, rec|A)$ (for all $n_i \in \mathcal{N}_{out}(u)$), PRINCE could find the explanation for each $(user, rec)$ pair in about 1 millisecond on average (for $k \leq 20$). Table 3 shows runtimes of PRINCE for different parameters. As we can see, the runtime grows linearly with k in both datasets. This is justified by Line 3 in Algorithm 1. Computing $PPR(n_i, rec|A)$ on-the-fly slows down the algorithm. The second and the fourth columns in Table 3 present the runtimes of PRINCE when the scores $PPR(n_i, rec|A)$ are computed using the reverse push algorithm for dynamic graphs [55]. Increasing β makes the computation slower (experimented at $k = 5$). All experiments were performed on an Intel Xeon server with 8 cores @ 3.2 GHz CPU and 512 GB main memory.

6 USER STUDY

Qualitative survey on usefulness. To evaluate the usefulness of counterfactual (action-oriented) explanations, we conducted a survey with Amazon Mechanical Turk (AMT) Master workers (www.mturk.com/help#what_are_masters). In this survey, we showed 500 workers three recommendation items ("Series Camelot", "Pregnancy guide book", "Nike backpack") and two different explanations for each. One explanation was limited to only the user's own actions (action-oriented), and the other was a path connecting the user to the item (connection-oriented).

We asked the workers three questions: (i) *Which method do you find more useful?*, where 70% chose the action-oriented method; (ii) *How do you feel about being exposed through explanations to others?*, where $\simeq$ 75% expressed a privacy concern either through complete disapproval or through a demand for anonymization; (iii) *Personally, which type of explanation matters to you more: "Action-oriented" or "connection-oriented"?*, where 61.2% of the workers chose the action-oriented explanations. We described action-oriented explanations as those allowing users to control their recommendation, while connection-oriented ones reveal connections between the user and item via other users and items.

Method	Mean	Std. Dev.	#Samples
PRINCE	1.91*	0.66	200
CredPaths [51]	1.78	0.63	200
PRINCE (Size=1)	1.87	0.66	154
PRINCE (Size=2)	1.88*	0.70	28
PRINCE (Size=3)	2.21*	0.52	18

Table 5: Results from the AMT measurement study on usefulness conducted on the Amazon data. An asterisk (*) indicates statistical significance of PRINCE over CredPaths (1-tailed paired t-test at $p < 0.05$).

Quantitative measurement of usefulness. In a separate study (conducted only on Amazon data for resource constraints), we compared PRINCE to a path-based explanation [51] (later referred to as CredPaths). We used the *credibility* measure from [51], scoring paths in descending order of the product of their edge weights. We computed the best path for all 500 user-item pairs (Sec. 5.1). This resulted in paths of a maximum length of three edges (four nodes including user and *rec*). For a fair comparison in terms of cognitive load, we eliminated all data points where PRINCE computed larger counterfactual sets. This resulted in about 200 user-item pairs, from where we sampled exactly 200. As explanations generated by PRINCE and CredPaths have a different format of presentation (a list of actions vs. a path), we evaluated each method separately to avoid presentation bias. For the sake of readability, we broke the paths into edges and showed each edge on a new line. Having three AMT Masters for each task, we collected 600(200×3) annotations for PRINCE and the same number for CredPaths.

A typical data point looks like a row in Table 6, that shows representative examples (Goodreads shown only for completeness). We divided the samples into ten HITs (Human Intelligence Tasks, a unit of job on AMT) with 20 data points in each HIT. For each data point, we showed a recommendation item and its explanation, and asked users about the usefulness of the explanation on a scale of $1 - 3$ ("Not useful at all", "Partially useful", and "Completely useful"). For this, workers had to imagine that they were a user of an e-commerce platform who received the recommendations as result of doing some actions on the platform. Only AMT Master workers were allowed to provide assessments.

To detect *spammers*, we planted one *honeypot* in each of the 10 HITs, that was a completely impertinent explanation. Subsequently, all annotations of detected spammers (workers who rated such irrelevant explanations as "completely useful") were removed (25% of all annotations).

Table 5 shows the results of our user study. It gives average scores and standard deviations, and it indicates statistical significance of pairwise comparisons with an asterisk. PRINCE clearly obtains higher usefulness ratings from the AMT judges, on average. Krippendorff's alpha [**?**] for PRINCE and CredPaths were found to

Method	Explanation for "Baby stroller" with category "Baby" [Amazon]
Prince	**Action 1:** You rated highly "Badger Basket Storage Cubby" with category "Baby" **Replacement Item:** "Google Chromecast HDMI Streaming Media Player" with category "Home Entertainment"
CredPaths	You rated highly "Men's hair paste" with category "Beauty" that was rated by "Some user" who also rated highly "Baby stroller" with category "Baby"
Method	**Explanation for "The Multiversity" with categories "Comics, Historical-fiction, Biography, Mystery" [Goodreads]**
Prince	**Action 1:** You rated highly "Blackest Night" with categories "Comics, Fantasy, Mystery, Thriller" **Action 2:** You rated highly "Green Lantern" with categories "Comics, Fantasy, Children" **Replacement Item:** "True Patriot: Heroes of the Great White North" with categories "Comics, Fiction, Crime, Fiction"
CredPaths	You follow "Some user" who has rated highly "The Multiversity" with categories "Comics, Historical-fiction, Biography, Mystery"

Table 6: Explanations from Prince vis-à-vis CredPaths [51].

Based on multiple actions explained simply and clearly. [Prince]
The recommendation is for a home plumbing item, but the action rated a glue. [Prince]
The explanation is complete as it goes into full details of how to use the product, which is in alignment of my review and useful to me. [CredPaths]
It's weird to be given recommendations based on other people. [CredPaths]

Table 7: Turkers' comments on their score justifications.

be $\simeq 0.5$ and $\simeq 0.3$ respectively, showing moderate to fair inter-annotator agreement. The superiority of Prince also holds for slices of samples where Prince generated explanations of size 1, 2 and 3. We also asked Turkers to provide *succinct justifications* for their scores on each data point. Table 7 shows some typical comments, where methods for generating explanations are in brackets.

7 RELATED WORK

Foundational work on explainability for collaborative-filtering-based recommenders was done by Herlocker et al. [20]. Over time, generating explanations (like [51]) has become tightly coupled with building systems that are geared for producing more transparent recommendations (like [6]). For broad surveys, see [44, 57]. With methods using matrix or tensor factorization [11, 47, 58], the goal has been to make latent factors more tangible. Recently, interpretable neural models have become popular, especially for text [8, 12, 40] and images [10], where the attention mechanism over words, reviews, items, or zones in images has been vital for interpretability. Efforts have also been made on generating readable explanations using models like LSTMs [28] or GANs [29].

Representing users, items, categories and reviews as a knowledge graph or a heterogeneous information network (HIN) has become popular, where explanations take the form of paths between the user and an item. This paradigm comprises a variety of mechanisms: learning path embeddings [1, 47], propagating user preferences [46], learning and reasoning with explainable rules [30, 50], and ranking user-item connections [18, 51]. In this work, we choose the recent approach in [51] as a representative for the family of path-based recommenders to compare Prince with. Finally, post-hoc or model-agnostic rationalizations for black-box models have attracted interest. Approaches include association rule mining [37], supervised ranking of user-item relationships [18], and reinforcement learning [48].

Random walks over HIN's have been pursued by a suite of works, including [13, 14, 16, 17, 23]. In a nutshell, the Personalized PageRank (PPR) of an item node in the HIN is used as a ranking criterion for recommendations. [35] introduced the RecWalk method, proposing a random walk with a nearly uncoupled Markov chain. Our work uses this framework. As far as we know, we are the first to study the problem of computing minimum subsets of edge removals (user actions) to change the top-ranked node in a counterfactual setup. Prior research on dynamic graphs, such as [15, 24], has addressed related issues, but not this very problem. A separate line of research focuses on the efficient computation of PPR. Approximate algorithms include power iteration [36], local push [2, 3, 55] and Monte Carlo methods [4, 5].

8 CONCLUSIONS AND FUTURE WORK

This work explored a new paradigm of action-based explanations in graph recommenders, with the goal of identifying minimum sets of user actions with the counterfactual property that their absence would change the top-ranked recommendation to a different item. In contrast to prior works on (largely path-based) recommender explanations, this approach offers two advantages: (i) explanations are *concise, scrutable*, and *actionable*, as they are *minimal* sets derived using a *counterfactual* setup over a user's *own* purchases, ratings and reviews; and (ii) explanations do not expose any information about other users, thus avoiding privacy breaches by design.

The proposed Prince method implements these principles using random walks for Personalized PageRank scores as a recommender model. We presented an efficient computation and correctness proof for computing counterfactual explanations, despite the potentially exponential search space of user-action subsets. Extensive experiments on large real-life data from Amazon and Goodreads showed that simpler heuristics fail to find the best explanations, whereas Prince can guarantee optimality. Studies with AMT Masters showed the superiority of Prince over baselines in terms of explanation usefulness.

ACKNOWLEDGEMENTS

This work was partly supported by the ERC Synergy Grant 610150 (imPACT) and the DFG Collaborative Research Center 1223. We would like to thank Simon Razniewski from the MPI for Informatics for his insightful comments on the manuscript.

REFERENCES

[1] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms* 11, 9 (2018).

[2] Reid Andersen, Christian Borgs, Jennifer Chayes, John Hopcraft, Vahab S Mirrokni, and Shang-Hua Teng. 2007. Local computation of PageRank contributions. In *WAW*.

[3] Reid Andersen, Fan Chung, and Kevin Lang. 2006. Local graph partitioning using Pagerank vectors. In *FOCS*.

[4] Konstantin Avrachenkov, Nelly Litvak, Danil Nemirovsky, and Natalia Osipova. 2007. Monte Carlo methods in PageRank computation: When one iteration is sufficient. *SIAM J. Numer. Anal.* 45, 2 (2007).

[5] Bahman Bahmani, Abdur Chowdhury, and Ashish Goel. 2010. Fast incremental and personalized PageRank. In *VLDB*.

[6] Krisztian Balog, Filip Radlinski, and Shushan Arakelyan. 2019. Transparent, Scrutable and Explainable User Models for Personalized Recommendation. In *SIGIR*.

[7] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder for English. In *EMNLP*.

[8] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *WWW*.

[9] Daizhuo Chen, Samuel P. Fraiberger, Robert Moakler, and Foster Provost. 2017. Enhancing transparency and control when drawing data-driven inferences about individuals. *Big data* 5, 3 (2017).

[10] Xu Chen, Hanxiong Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2019. Personalized Fashion Recommendation with Visual Explanations based on Multimodal Attention Network: Towards Visually Explainable Recommendation. In *SIGIR*.

[11] Xu Chen, Zheng Qin, Yongfeng Zhang, and Tao Xu. 2016. Learning to Rank Features for Recommendation over Multiple Categories. In *SIGIR*.

[12] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *WSDM*.

[13] Fabian Christoffel, Bibek Paudel, Chris Newell, and Abraham Bernstein. 2015. Blockbusters and Wallflowers: Accurate, Diverse, and Scalable Recommendations with Random Walks. In *RecSys*.

[14] Colin Cooper, Sang-Hyuk Lee, Tomasz Radzik, and Yiannis Siantos. 2014. Random walks in recommender systems: Exact computation and simulations. In *WWW*.

[15] Balázs Csanád Csáji, Raphaël M Jungers, and Vincent Blondel. 2014. PageRank optimization by edge selection. *Discrete Applied Mathematics* 169 (2014).

[16] Christian Desrosiers and George Karypis. 2011. A Comprehensive Survey of Neighborhood-based Recommendation Methods. In *Recommender Systems Handbook*.

[17] Chantat Eksombatchai, Pranav Jindal, Jerry Zitao Liu, Yuchen Liu, Rahul Sharma, Charles Sugnet, Mark Ulrich, and Jure Leskovec. 2018. Pixie: A System for Recommending 3+ Billion Items to 200+ Million Users in Real-Time. In *WWW*.

[18] Azin Ghazimatin, Rishiraj Saha Roy, and Gerhard Weikum. 2019. FAIRY: A Framework for Understanding Relationships between Users' Actions and their Social Feeds. In *WSDM*.

[19] Taher H. Haveliwala. 2003. Topic-sensitive Pagerank: A context-sensitive ranking algorithm for Web search. *TKDE* 15, 4 (2003).

[20] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. 2000. Explaining Collaborative Filtering Recommendations. In *CSCW*.

[21] Mohsen Jamali and Martin Ester. 2009. *TrustWalker: A* random walk model for combining trust-based and item-based recommendation. In *KDD*.

[22] Glen Jeh and Jennifer Widom. 2003. Scaling personalized Web search. In *WWW*.

[23] Zhengshen Jiang, Hongzhi Liu, Bin Fu, Zhonghai Wu, and Tao Zhang. 2018. Recommendation in heterogeneous information networks based on generalized random walk model and Bayesian personalized ranking. In *WSDM*.

[24] Jian Kang, Meijia Wang, Nan Cao, Yinglong Xia, Wei Fan, and Hanghang Tong. 2018. AURORA: Auditing PageRank on Large Graphs. In *Big Data*.

[25] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009).

[26] Pigi Kouki, James Schaffer, Jay Pujara, John O'Donovan, and Lise Getoor. 2019. Personalized explanations for hybrid recommender systems. In *IUI*.

[27] Johannes Kunkel, Tim Donkers, Lisa Michael, Catalin-Mihai Barbu, and Jürgen Ziegler. 2019. Let Me Explain: Impact of Personal and Impersonal Explanations on Trust in Recommender Systems. In *CHI*.

[28] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *SIGIR*.

[29] Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Why I like it: Multi-task learning for recommendation and explanation. In *RecSys*.

[30] Weizhi Ma, Min Zhang, Yue Cao, Woojeong Jin, Chenyang Wang, Yiqun Liu, Shaoping Ma, and Xiang Ren. 2019. Jointly Learning Explainable Rules for Recommendation with Knowledge Graph. In *WWW*.

[31] Ashwin Machanavajjhala, Aleksandra Korolova, and Atish Das Sarma. 2011. Personalized social recommendations: Accurate or private?. In *VLDB*.

[32] David Martens and Foster Provost. 2014. Explaining Data-Driven Document Classifications. *MIS Quarterly* 38, 1 (2014).

[33] Tim Miller, Rosina Weber, David Aha, and Daniele Magazzeni. 2019. IJCAI 2019 Workshop on Explainable AI (XAI).

[34] Julie Moeyersoms, Brian d'Alessandro, Foster Provost, and David Martens. 2016. Explaining classification models built on high-dimensional sparse data. *arXiv preprint arXiv:1607.06280* (2016).

[35] Athanasios N. Nikolakopoulos and George Karypis. 2019. Recwalk: Nearly uncoupled random walks for top-n recommendation. In *WSDM*.

[36] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the Web.* Technical Report. Stanford InfoLab.

[37] Georgina Peake and Jun Wang. 2018. Explanation mining: Post hoc interpretability of latent factor models for recommendation systems. In *KDD*.

[38] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should I trust you?: Explaining the predictions of any classifier. In *KDD*.

[39] Joy Rimchala, Jineet Doshi, Qiang Zhu, Diane Chang, Nick Hoh, Conrad De Peuter, Shir Meir Lador, and Sambarta Dasgupta. 2019. KDD Workshop on Explainable AI for Fairness, Accountability, and Transparency.

[40] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *RecSys*.

[41] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. 2016. A survey of heterogeneous information network analysis. *TKDE* 29, 1 (2016).

[42] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and Philip S. Yu. 2017. A Survey of Heterogeneous Information Network Analysis. *TKDE* 29, 1 (2017).

[43] Chuan Shi, Zhiqiang Zhang, Ping Luo, Philip S Yu, Yading Yue, and Bin Wu. 2015. Semantic path based personalized recommendation on weighted heterogeneous information networks. In *CIKM*.

[44] Nava Tintarev and Judith Masthoff. 2007. A survey of explanations in recommender systems. In *Workshop on Ambient Intelligence, Media and Sensing*.

[45] Mengting Wan and Julian McAuley. 2018. Item recommendation on monotonic behavior chains. In *RecSys*.

[46] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *CIKM*.

[47] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018. Explainable recommendation via multi-task learning in opinionated text data. In *SIGIR*.

[48] Xiting Wang, Yiru Chen, Jie Yang, Le Wu, Zhengtao Wu, and Xing Xie. 2018. A Reinforcement Learning Framework for Explainable Recommendation. In *ICDM*.

[49] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *AAAI*.

[50] Yikun Xian, Zuohui Fu, S. Muthukrishnan, Gerard de Melo, and Yongfeng Zhang. 2019. Reinforcement Knowledge Graph Reasoning for Explainable Recommendation. In *SIGIR*.

[51] Fan Yang, Ninghao Liu, Suhang Wang, and Xia Hu. 2018. Towards Interpretation of Recommender Systems with Sorted Explanation Paths. In *ICDM*.

[52] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *WSDM*.

[53] Xiao Yu, Xiang Ren, Yizhou Sun, Bradley Sturt, Urvashi Khandelwal, Quanquan Gu, Brandon Norick, and Jiawei Han. 2013. Recommendation in heterogeneous information networks with implicit user feedback. In *RecSys*.

[54] Chuxu Zhang, Ananthram Swami, and Nitesh Chawla. 2019. SHNE: Representation Learning for Semantic-Associated Heterogeneous Networks. In *WSDM*.

[55] Hongyang Zhang, Peter Lofgren, and Ashish Goel. 2016. Approximate personalized pagerank on dynamic graphs. In *KDD*.

[56] Quanshi Zhang, Lixin Fan, Bolei Zhou, Sinisa Todorovic, Tianfu Wu, and Ying Nian Wu. 2019. CVPR-19 Workshop on Explainable AI.

[57] Yongfeng Zhang and Xu Chen. 2018. Explainable recommendation: A survey and new perspectives. *arXiv preprint arXiv:1804.11192* (2018).

[58] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *SIGIR*.

[59] Yongfeng Zhang, Yi Zhang, Min Zhang, and Chirag Shah. 2019. EARS 2019: The 2nd International Workshop on ExplainAble Recommendation and Search. In *SIGIR*.