SAShA: Semantic-Aware Shilling Attacks on Recommender Systems Exploiting Knowledge Graphs

Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, Eugenio Di Sciascio, and Felice Antonio Merra(✉)

Politecnico di Bari, Bari, Italy
{vitowalter.anelli,yashar.deldjoo,tommaso.dinoia,
eugenio.disciascio,felice.merra}@poliba.it

Abstract. Recommender systems (RS) play a focal position in modern user-centric online services. Among them, collaborative filtering (CF) approaches have shown leading accuracy performance compared to content-based filtering (CBF) methods. Their success is due to an effective exploitation of similarities/correlations encoded in user interaction patterns, which is computed by considering common items users rated in the past. However, their strength is also their weakness. Indeed, a malicious agent can alter recommendations by adding fake user profiles into the platform thereby altering the actual similarity values in an engineered way.

The spread of well-curated information available in knowledge graphs ($\mathcal{KG}$) has opened the door to several new possibilities in compromising the security of a recommender system. In fact, $\mathcal{KG}$ are a wealthy source of information that can dramatically increase the attacker's (and the defender's) knowledge of the underlying system. In this paper, we introduce *SAShA*, a new attack strategy that leverages semantic features extracted from a knowledge graph in order to strengthen the efficacy of the attack to standard CF models. We performed an extensive experimental evaluation in order to investigate whether *SAShA* is more effective than baseline attacks against CF models by taking into account the impact of various semantic features. Experimental results on two real-world datasets show the usefulness of our strategy in favor of attacker's capacity in attacking CF models.

Keywords: Recommender system · Knowledge graph · Shilling attack

Authors are listed in alphabetical order.

A. Harth et al. (Eds.): ESWC 2020, LNCS 12123, pp. 307–323, 2020.
https://doi.org/10.1007/978-3-030-49461-2_18

1 Introduction

Recommender Systems (RS) are nowadays considered as the pivotal technical solution to assist users' decision-making process. They are gaining momentum as the overwhelming volume of products, services, and multimedia contents on the Web has made the users' choices more difficult. Among them, Collaborative filtering (CF) approaches have shown very high performance in real-world applications (e.g., Amazon [26]). Their key insight is that users prefer products experienced by similar users and then, from an algorithmic point of view, they mainly rely on the exploitation of user-user and item-item similarities. Unfortunately, malicious users may alter similarity values. Indeed, these similarities are vulnerable to the insertion of fake profiles. The injection of such manipulated profiles, named shilling attack [20], aims to *push* or *nuke* the probabilities of items to be recommended.

Recently, several works have proposed various types of attacks, classified into two categories [9]: *low-knowledge* and *informed* attack strategies. In the former attacks, the malicious user (or adversary) has poor system-specific knowledge [25,28]. In the latter, the attacker has precise knowledge of the attacked recommendation model and the data distribution [12,25].

Interestingly, the astonishing spread of knowledge graphs ($\mathcal{KG}$) may suggest new knowledge-aware strategies to mine the security of RS. In a Web mainly composed of unstructured information, $\mathcal{KG}$ are the foundation of the Semantic Web. They are becoming increasingly important as they can represent data exploiting a manageable and inter-operable semantic structure. They are the pillars of well-known tools like IBM Watson [7], public decision-making systems [34], and advanced machine learning techniques [2,4,13]. Thanks to the Linked Open Data (LOD) initiative[1], we have witnessed the growth of a broad ecosystem of linked data datasets known as LOD-cloud[2]. These $\mathcal{KG}$ contain detailed information about several domains. In fact, if a malicious user would attack one of these domains, items' semantic descriptions would be priceless.

The main contributions envisioned in the present work is to study the possibility of leveraging semantic-encoded information with the goal to improve the efficacy of an attack in favor/disfavor of (a) given target item(s). Particularly, one of the features distinguishing this work from previous ones is that it exploits publicly available information resources obtained from $\mathcal{KG}$ to generate more influential fake profiles that are able to undermine the performance of CF models. This attack strategy is named semantic-aware shilling attack *SAShA* and extends state-of-the-art shilling attack strategies such as *Random*, *Love-hate*, and *Average* based on the gathered semantic knowledge. It is noteworthy that the extension we propose solely relies on publicly available information and does not provide to the attacker any additional information about the system.

[1] https://data.europa.eu/euodp/en/linked-data.
[2] https://lod-cloud.net/.

In this work, we aim at addressing the following research questions:

RQ1 Can public available semantic information be exploited to develop more effective shilling attack strategies against CF models, where the effectiveness is measured in terms of overall prediction shift and overall hit ratio?

RQ2 Can we assess which is the most impactful type of semantic information? Is multiple hops extraction of semantic-features from a knowledge graph more effective than single-hop features?

To this end, we have carried out extensive experiments to evaluate the impact of the proposed *SAShA* against standard CF model using two real-world recommender systems datasets (`LibraryThing` and `Yahoo!Movies`). Experimental results indicate that $\mathcal{KG}$ information is a rich source of knowledge that can in fact worryingly improve the effectiveness of attacks.

The remainder of the paper is organized as follows. In Sect. 2, we analyze the state-of-the-art of CF models as well as shilling attacks. In Sect. 3, we describe the proposed approach (SAShA). Section 4 focuses on experimental validation of the proposed attacks scenarios, where we provide a discussion of the experimental results. Finally, in Sect. 5, we present conclusions and introduce open challenges.

2 Related Work

In this Section, we focus on related literature on recommender systems and state-of-the-art of attacks on collaborative recommender models.

2.1 Recommender Systems (RSs)

Recommendation models can be broadly categorized as content-based filtering (CBF), collaborative filtering (CF) and hybrid. On the one hand, CBF uses items' content attributes (features) together with target user's own interactions in order to create a user profile characterizing the nature of her interest(s). On the other hand, CF models generate recommendation by solely exploiting the similarity between interaction patterns of users. Today, CF models are the mainstream of academic and industrial research due to their state-of-the-art recommendation quality particularly when sufficient amount of interaction data—either explicit (e.g., rating scores) or implicit (previous clicks, check-ins etc.)—are available. Various CF models developed today can be classified into two main groups: memory-based and model-based. While memory-based models make recommendations exclusively based on similarities in user's interactions (user-based CF [23,32]) or items' interactions (item-based CF [23,33]), model-based approaches compute a latent representation of items and users [24], whose linear interaction can explains an observed feedback. Model-based approaches can be implemented by exploiting different machine learning techniques. Among them, matrix factorization (MF) models play a paramount role.

It should be noted, that modern RS nowadays may exploit a variety of side information such as metadata (tags, reviews) [29], social connections [6], image

and audio signal features [14] and users-items contextual data [3] to build more in-domain (i.e., domain-dependent) or context-aware recommendations models. $\mathcal{KG}$ are another rich source of information that have gained increased popularity in the community of RS for building knowledge-aware recommender systems (KARS). These models can be classified into: (i) path-based methods [19,37], which use meta-paths to evaluate the user-item similarities and, (ii) $\mathcal{KG}$ embedding-based techniques, that leverages $\mathcal{KG}$ embeddings to semantically regularize items latent representations [16,21,35]. More recently, $\mathcal{KG}$ have also been used to support the reasoning and explainability of recommendations [5,36].

For the simplicity of the presentation, in this work we step our attention aside (shilling attacks against) CF models leveraging these side information for the core task of recommendation, and leave it for an extension in future works. We do however make a fundamental assumption in all considered scenarios that the "attacker can have access to $\mathcal{KG}$, given their free accessibility and use them to shape more in-domain attacks."

2.2 Shilling Attacks on Recommender System

Despite the widespread application of customer-oriented CF models by online services adopted to increase their traffic and promote sales, the reliance of these models on the so-called "word-of-mouth" (i.e., what other people like and dislike), makes them at the same time vulnerable to meticulously crafted profiles that aim to alter distribution of ratings so to misuse this dependency toward a particular (malicious) purpose. The motivation for such shilling attacks can be many unfortunately, including personal gain, market penetration by rival companies [25], malicious reasons and even causing complete mischief on an underlying system [20].

In the literature, one standard way to classify these shilling attacks is based on the *intent* and amount of *knowledge* required to perform attacks. According to the intent, generally attacks are classified as *push attacks* that aim to increase the appeal of some targeted items, and *nuke items*, which conversely aim to lower the popularity of some targeted items. As for the knowledge level, they can be categorized according to *low-knowledge attacks* and *informed attack* strategies. Low-knowledge attacks require little or no knowledge about the rating distribution [25,28], while, informed attacks assume adversaries with knowledge on dataset rating distribution, which use this knowledge to generate effective fake profiles for shilling attacks [25,30].

A large body of research work has been devoted on studying shilling attacks from multiple perspectives: altering the performance of CF models [12,15,25], implementation attack detection policies [8,11,38] and building robust recommendation models against attacks [28,30]. Regardless, a typical characteristic of the previous literature on shilling attack strategies is that they usually target the relations between users, and items, based on similarities scores estimated on their past feedback (e.g., ratings). However, these strategies do not consider the possibility of exploiting publicly available $\mathcal{KG}$ to gain more information on the semantic similarities between the items available in the RS catalogue.

Indeed, considering that product or service providers' catalogues are freely accessible to everyone, this work presents a novel attack strategy that exploits a freely accessible knowledge graph (DBpedia) to assess if attacks based on semantic similarities between items are more effective than baseline versions that rely only on rating scores of users.

3 Approach

In this section, we describe the development of a novel method for integrating information obtained from a knowledge graph into the design of shilling attacks against targeted items in a CF system. We first introduce the characteristics of $\mathcal{KG}$ in Sect. 3.1. Afterwards, we present the proposed semantic-aware extensions to variety of popular shilling attacks namely: *Random*, *Love-Hate*, and *Average* attacks in Sect. 3.2.

3.1 Knowledge Graph: Identification of Content from $\mathcal{KG}$

A knowledge graph can be seen as a structured repository of knowledge, represented in the form a graph, that can encode different types of information:

- **Factual.** General statements as *Rika Dialina was born in Crete* or *Heraklion is the capital of Crete* where we describe an entity by its attributes which are in turn connected to other entities (or literal values);
- **Categorical.** These statements bind the entity to a specific category (i.e., the categories associated to an article in Wikipedia pages). Often, categories are part of a hierarchy. The hierarchy lets us define entities in a more generic or specific way;
- **Ontological.** We can classify entities in a more formal way using a hierarchical structure of classes. In contrast to categories, sub-classes and super-classes are connected through IS-A relations.

In a knowledge graph we can represent each entity through the triple structure $\sigma \xrightarrow{\rho} \omega$, with a *subject* ($\sigma$), a *relation* (*predicate*) ρ and an *object* (ω). Among the multiple ways to represent features coming from a knowledge graph, we have chosen to represent each distinct triple as a single feature [5]. Hence, given a set of items $I = \{i_1, i_2, \ldots, i_N\}$ in a collection and the corresponding triples $\langle i, \rho, \omega \rangle$ in a knowledge graph, we can build the set of 1-hop features as $1\text{-}HOP\text{-}F = \{\langle \rho, \omega \rangle \mid \langle i, \rho, \omega \rangle \in \mathcal{KG} \text{ with } i \in I\}$.

In an analogous way we can identify 2-hop features. Indeed, we can continue exploring $\mathcal{KG}$ by retrieving the triples $\omega \xrightarrow{\rho'} \omega'$, where ω is the *object* of a 1-hop triple and the *subject* of the new triple. Here, the double-hop *relation* (*predicate*) is denoted by ρ' while the new *object* is referred as (ω'). Hence, we define the overall feature set as $2\text{-}HOP\text{-}F = \{\langle \rho, \omega, \rho', \omega' \rangle \mid \langle i, \rho, \omega, \rho', \omega' \rangle \in \mathcal{KG} \text{ with } i \in I\}$. With respect to the previous classification of different types of information in a knowledge graph, we consider a 2-hop feature as Factual if and only if both relations (ρ, and ρ') are Factual. The same holds for the other types of encoded information.

3.2 Strategies for Attacking a Recommender System

A shilling attack against a recommendation model is based on a set of fake profiles meticulously created by the attacker and inserted into the system. The ultimate goal is to alter recommendation in favor of (push scenario) or organist (nuke scenario) a single target item i_t. In this work, we focus on the push attack scenario but everything can be reused also in case of a nuke one. The fake user profile (attack profile) follows the general structure proposed by Bhaumik [8] shown in Fig. 1. It is built up of a rating vector of dimensionality N where N is the entire items in the collection ($N = |I_S| + |I_F| + |I_\emptyset| + |I_T|$). The profile is subdivided into four non-overlapping segments:

I_S			I_F			$I_\emptyset$			I_T
$i_s^{(1)}$	...	$i_s^{(\alpha)}$	$i_f^{(1)}$	...	$i_f^{(\phi)}$	$i_\emptyset^{(1)}$	...	$i_\emptyset^{(\chi)}$	i_t

Fig. 1. General form of a fake user profile

- I_T: This is the *target item* for which a rating score will be predicted by the recommendation model. Often, this rating is assigned to be the maximum or minimum possible score based on the attack goal (push or pull).
- $I_\emptyset$: This is the *unrated item* set, i.e., items that will not contain any ratings in the profile.
- I_F: The *filler item* set. These are items for which rating scores will be assigned specific to each attack strategy.
- I_S: The *selected item* set. These items are selected in the case of *informed attack* strategies, which exploit attacker's knowledge to maximize the attack impact, for instance by selecting items with the higher number of ratings.

The ways I_S and I_F are chosen depend on the attack strategy. The attack size is defined as the number of injected fake user profiles. Hereafter, $\phi = |I_F|$ indicates the filler size, $\alpha = |I_S|$ the selected item set size and $\chi = |I_\emptyset|$ is the size of unrated items. In this paper, we focus our attention on the selection process of I_F since I_S is built by exploiting the attacker's knowledge of the data distribution.

Semantic-Aware Shilling Attack Strategies (SAShA). While previous work on RS has investigated the impact of different standard attack models on CF system, in this work, we propose to strengthen state-of-the-art strategies via the exploitation of semantic similarities between items.

This attack strategy generates fraudulent profiles by exploiting $\mathcal{KG}$ information to fill I_F. The key idea is that we can compute the semantic similarity of the target item i_t with all the items in the catalog using $\mathcal{KG}$-derived features. Then, we use this information to select the filler items of each profile to generate the set I_F.

The insight of our approach is that a similarity value based on semantic features leads to more natural and coherent fake profiles. These profiles are

indistinguishable from the real ones, and they effortlessly enter the neighborhood of users and items. In order to compute the semantic similarity between items, in our experimental evaluation, we exploit the widely adopted Cosine Vector Similarity [17].

To test our semantic-aware attacks to recommender systems, we propose three original variants of low-knowledge and informed attack strategies: random attack, love-hate attack, and s average attack.

- *Semantic-aware Random Attack (SAShA-random)* is an extension of Random Attack [25]. The baseline version is a naive attack in which each fake user is composed only of random items ($\alpha = 0, \phi = \textit{profile-size}$). The fake ratings are sampled from all items using a uniform distribution. We modify this attack by changing the set to extract the items. In detail, we extract items to fill I_F from a subset of items that are most similar to i_t. We compute the item-item *Cosine Similarity* using the semantic features as introduced in Sect. 3.1. Then, we build a set of most-similar items, considering the first quartile of similarity values. Finally, we extract ϕ items from this set, adopting a uniform distribution.
- *Semantic-aware Love-Hate Attack* (*SAShA-love-hate*) is a low-knowledge attack that extends the standard Love-Hate attack [28]. This attack randomly extracts filler items I_F from the catalog. All these items are associated with the minimum possible rating value. The Love-Hate attack aims to reduce the average rating of all the platform items but the target item. Indeed, even though the target item is not present in the fake profiles, its relative rank increases. We have re-interpreted the rationale behind the Love-Hate attack by taking into account the semantic description of the target item and its similarity with other items within the catalogue. In this case, we extract items to fill I_F from the 2nd, 3rd, and 4th quartiles. As in the original variant, the rationale is to select the most dissimilar items.
- *Semantic-aware Average Attack* (*SAShA-average*) is an informed attack that extends the AverageBots attack [28]. The baseline attack takes advantage of the mean and the variance of the ratings. Then, it randomly samples the rating of each filler item from a normal distribution built using the previous mean and variance. Analogously to *SAShA-random*, we extend the baseline by extracting the filler items from the sub-set of most similar items. We use as candidate items the ones in the first quartile regarding their similarity with i_t.

4 Experimental Evaluation

This section is devoted to comparing the proposed approaches against baseline attack strategies. We first introduce the experimental setup, where we present the two well-known datasets for recommendation scenarios. Then, we describe the feature extraction and selection procedure we have adopted to form semantic-aware shilling attacks. Finally, we detail the three canonical CF models we have

analyzed. We have carried extensive experiments intented to answer the research questions in Sect. 1. In particular, we aim to assess: (i) whether freely available semantic knowledge can help to generate stronger shilling attacks; (ii) if $\mathcal{KG}$ features types have a different influence on *SAShA* effectiveness; (iii) what is the most robust CF-RS against *SAShA* attacks.

4.1 Experimental Setting

Datasets. In the experiments, we have exploited two well-known datasets with explicit feedbacks to simulate the process of a recommendation engine: `LibraryThing` [18] and `Yahoo!Movies`. The first dataset is derived from the social cataloging web application `LibraryThing`[3] and contains ratings ranging from 1 to 10. To speed up the experiments, we have randomly sampled with a uniform distribution the 25% of the original items in the dataset. Moreover, in order to avoid cold situations (which are usually not of interest in attacks to recommender systems) we removed users with less than five interactions. The second dataset contains movie ratings collected on `Yahoo!Movies`[4] up to November 2003. It contains ratings ranging from 1 to 5, and mappings to `MovieLens` and `EachMovie` datasets. For both datasets, we have used the items-features sets 1-*HOP-F* and 2-*HOP-F* extracted from `DBpedia` by exploiting mappings which are publicly available at https://github.com/sisinflab/LinkedDatasets. We show datasets statistics in Table 1.

Table 1. Datasets statistics.

Dataset	#Users	#Items	#Ratings	Sparsity	#F-1Hop	#F-2Hops
`LibraryThing`	4816	2,256	76,421	99.30%	56,019	4,259,728
`Yahoo!Movies`	4000	2,526	64,079	99.37%	105733	6,697,986

Feature Extraction. We have extracted the semantic information to build *SAShA* exploiting the public available item-entity mapping to `DBpedia`. We did not consider noisy features containing the following predicates: `owl:sameAs`, `dbo:thumbnail`, `foaf:depiction`, `prov:wasDerivedFrom`, `foaf:isPrimaryTopicOf`, as suggested in [5].

Feature Selection. To analyze the impact of different feature types, we have performed experiments considering categorical (CS), ontological (OS) and factual (FS) features. We have chosen to explore those classes of features since they are commonly adopted in the community [5]. For the selection of single-hop (1H) features, the employed policies are:

[3] http://www.librarything.com/.
[4] http://research.yahoo.com/Academic_Relations.

- **CS-1H**, we select the features containing the property `dcterms:subject`;
- **OS-1H**, we consider the features including the property `rdf:type`;
- **FS-1H**, we pick all the features but ontological and categorical ones.

For the selection of double-hops (2H) features, the applied policies are:

- **CS-2H**, we select the features with properties equal to either `dcterms:subject` or `skos:broader`;
- **OS-2H**, we consider the features including the properties `rdf:type`, `rdf-schema:subClassOf` or `owl:equivalentClass`;
- **FS-2H**, we pick up the features which are not in the previous two categories.

Noteworthy, we have not put any categorical/ontological features into the noisy list. If some domain-specific categorical/ontological features are not in the respective lists, we have considered them as factual features.

Feature Filtering. Following the aforementioned directions, we have extracted $1H$, and $2H$ features for `LibraryThing`, and `Yahoo!Movies`. Due to the extent of the catalogs, we obtained millions of features. Consequently, we removed irrelevant features following the filtering technique proposed in [18,31]. In detail, we dropped off all the features with more than 99.74% (t) of missing values and more than t of distinct values. In detail, we dropped off all the features with more than 99.74% of missing values and distinct values. The statistics of the resulting datasets is depicted in Table 2.

Table 2. Selected features in the different settings either for single and double hops.

Dataset	CS-1H		OS-1H		FS-1H		CS-2H		OS-2H		FS-2H	
	Tot.	*Selected*	*Tot.*	*Selected*	*Tot.*	*Selected*	*Tot.*	*Selected*	*Tot.*	*Selected*	*Tot.*	*Selected*
`LibraryThing`	3890	458	2090	367	53929	2398	9641	1140	3723	597	4256005	306289
`Yahoo!Movies`	5555	1125	3036	691	102697	7050	8960	1956	3105	431	6694881	516114

Recommender Models. We have conducted experiments considering all the attacks described in Sect. 3.2 on the following baseline Collaborative Filtering Recommender Systems:

- **User-kNN** [23,32] predicts the score of unknown user-item pairs ($\hat{r}_{ui}$) considering the feedback of the users in the neighborhood. We have tested *SAShA* using the formulation mentioned in [23]. It considers the user and item's ratings biases. Let u be a user inside the set of users U, and i be an item in the set of items I, we estimate the rating given by u to i based on the following Equation:

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{v \in U_i^k(u)} \delta(u,v) \cdot (r_{vi} - b_{vi})}{\sum_{v \in U_i^k(u)} \delta(u,v)} \tag{1}$$

where δ is the distance metric to measure the similarity between users, $U_i^k(u)$ is the set of k-neighborhood users v of u. We define b_{ui} as $\mu + b_u + b_i$, where μ, b_u, b_i are the overall average rating, the observed bias of user u and item i, respectively. Following directions suggested in [10], we apply as distance metric δ the *Pearson Correlation* and a number of neighbors k equal to 40.

– **Item-kNN** [23,33] estimates the user-item rating score ($\hat{r}_{ui}$) using the recorded feedback given by u to the k-items j in the neighborhood of the item i. Equation 2 defines the rating prediction formula for Item-kNN.

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in I_u^k(i)} \delta(i,j) \cdot (r_{uj} - b_{uj})}{\sum_{j \in I_u^k(i)} \delta(i,j)} \tag{2}$$

In Eq. 2, the set of k items inside the i neighborhood is denoted as $I_u^k(i)$. The similarity function δ and the number of considered neighbors k are selected as in User-kNN.

– **Matrix Factorization (MF)** [24] is a latent factor model used for items recommendation task that learns user-item preferences, by factorizing the sparse user-item feedback matrix. The learned user and item representation, fitted on previously recorder interactions, are exploited to predict $\hat{r}_{ui}$ as follows:

$$\hat{r}_{ui} = b_{ui} + \mathbf{q}_i^T \mathbf{p}_u \tag{3}$$

In Eq. 3, $\mathbf{q}_i \in \mathbb{R}^f$ and $\mathbf{p}_u \in \mathbb{R}^f$ are the latent vectors for item i and user u learned by the model. We set the number of latent factors f to 100, as suggested in [22].

Evaluation Metrics. We have evaluated our attack strategy by adopting *Overall Prediction Shift*, and *Overall Hit-Ratio@k*. Let I_T be the set of attacked items, and U_T be the set of users that have not rated the items in I_T. We define the *Overall Prediction Shift* (PS) [1] as the average variation of the predicted score for the target item.

$$PS(I_T, U_T) = \frac{\sum_{i \subset I_T, u \subset U_T} (\hat{r}_{ui} - r_{ui})}{|I_T| \times |U_T|} \tag{4}$$

where $\hat{r}_{ui}$ is the predicted rating on item i for user u after the shilling attack, and r_{ui} is the prediction without (before) attack. We define the *Overall Hit-Ratio@k* ($HR@k$) [1] as the average of $hr@k$ for each attacked item. Equation 5 defines $HR@k$ as:

$$HR@k(I_T, U_T) = \frac{\sum_{i \subset I_T} hr@k(i, U_T)}{|I_T|} \tag{5}$$

where $hr@k(i, U_T)$ measures the number of occurrences of the attacked item i in the top-k recommendation lists of the users in $|U_T|$.

Evaluation Protocol. Inspired by the evaluation proposed in [25,27], we have performed a total of 126 experiments. For each dataset, we have generated the recommendations concerning all users using the selected CF models (i.e., User-kNN, Item-kNN and MF). Then, we have added the fake profiles generated according to the baseline attack strategies, and we have re-computed the recommendation lists. We have evaluated the effectiveness of each attack by measuring the above-mentioned metrics on both the initial and the new recommendation lists. After this step, we have performed a series of *SAShA* attacks as described in Sect. 3. In detail, we have considered different feature types (i.e., categorical, ontological and factual) extracted at 1 or 2 hops. Finally, we have evaluated the $HR@k$ and PS for each *SAShA* variant comparing it against baselines. It is worth to note that, in our experiments, each attack is a *push attack*. Indeed, the attacker's purpose is to increase the probability that the target item is recommended. Moreover, by adopting the evaluation protocol proposed in [15,28], we have performed the attacks considering a different amount of added fake user profiles: $1\%, 2.5\%$ and 5% of the total number of users. We have tested the attacks considering 50 randomly sampled target items.

4.2 Results and Discussion

The discussion of results is organized accordingly to the research questions stated in Sect. 1. Firstly, we describe the influence of semantic knowledge on attack strategies. Later, we compare the impact of the different types of semantic information.

Analysis of the Effectiveness of Semantic Knowledge on Shilling Attacks. The first Research Question aims to check whether the injection of `Linked Open Data` as a new source of knowledge can represent a 'weapon' for attackers against CF-RS. Table 3 reports the results of the $HR@10$ for each attack. For both the baseline and semantic-aware variants, we highlight in bold the best results.

Starting from the analysis of the low-informed *random attack*, experiments show that the semantic-aware attacks are remarkably effective. For instance, the semantic-attacks with ontological information at single hop (*SAShA-OS-1H*), outperforms the baselines independently of the attacked model. To support these insights, we can observe the PS resulting from random attacks. Figure 2a shows that any variant of *SAShA* has a higher prediction shift w.r.t. the baseline for `Yahoo! Movies`. In Fig. 2b, we can notice that the semantic strategy is the most effective one for each model. As an example, the PS of *Rnd-SAShA-OS-1H* increases up to 6.82% over the corresponding baseline in the case of attacks against User-kNN on `Yahoo! Movies` dataset. The full results are online available[5].

In Table 3, we observe that the injection of semantic information for *love-hate* attack is not particularly effective. This can be due to the specific

[5] https://github.com/sisinflab/papers-results/tree/master/sasha-results.

Table 3. Experimental results for *SAShA* at single and double hops.

Metric: $HR@10$		LibraryThing									Yahoo!Movies								
		User-kNN			Item-kNN			MF			User-kNN			Item-kNN			MF		
		1%	2.5%	5%	1%	2.5%	5%	1%	2.5%	5%	1%	2.5%	5%	1%	2.5%	5%	1%	2.5%	5%
Rnd	baseline	.074	.157	.230	.281	.457	.557	.767	.900	.942	.189	.366	.449	.329	.508	.598	.410	.580	.702
	CS-1H	.068*	.143*	.213*	.271*	.441*	.558	.778*	.898	.940	.202	.372	.455*	.336	.522*	.609*	.430*	.607*	.707
	OS-1H	**.081***	**.170***	**.250***	**.290***	**.467***	**.576***	**.786***	**.902**	**.944**	**.217***	**.394***	**.477***	**.345***	**.535***	**.622***	**.446***	**.635***	**.742***
	FS-1H	.072	.154	.229	.280	.455	.570*	.786*	.901	.942	.213*	.381*	.468*	.338*	.530*	.619*	.442*	.623*	.728*
L-H	baseline	.502	.518	.518	.874	.952	.978	.955	.987	**.995**	.604	.608	.605	.888	.930	**.958**	.956	.967	**.980**
	CS-1H	.502	.518	.518	**.876***	**.953**	**.979**	**.957**	.987	.994	.604	.608	.605*	**.889**	.932	.957	.956	.967	.979
	OS-1H	.502	.518	.518	.870*	.950*	.974*	.955*	.986	.994	.604	.605	.605	.887	**.933**	.955*	.956	.967	.979
	FS-1H	.502	.518	.518	.874	.951	.977	.955	.987	.993	.604*	.608	.605	.888	**.933**	.956	.956	.967	.979
Avg	baseline	.086	.197	.285	.313	**.508**	.605	.803	.915	.951	.233	**.416**	.494	**.374**	**.574**	**.654**	**.489**	**.685**	**.788**
	CS-1H	.081*	.187*	.269*	.301*	.507	.621*	.814*	.915	.950	.220*	.399*	.479*	.357*	.554*	.639*	.467*	.652*	.744*
	OS-1H	**.093***	**.202**	**.289**	.313	.507	**.610***	**.810**	.911	**.948**	**.237**	.412	.494	.371	.563*	.646*	.475	.656*	.754*
	FS-1H	.084	.190*	.272*	.305*	.504	.614*	.811	.911	.946*	.215*	.397*	.473*	.350*	.547*	.634*	.448*	.627*	.729*
Rnd	baseline	.074	.157	.230	.281	**.457**	.557	.767	.900	.942	.189	.366	.449	.329	.508	.598	.410	.580	.702
	CS-2H	.068*	.143*	.213*	.270*	.441*	.558	**.799***	.897	.940	**.234***	**.410***	**.494***	**.368***	**.564***	**.644***	**.473***	**.667***	**.772***
	OS-2H	**.075**	.157	**.231**	.252	.455	**.567***	.783*	**.901**	.941	.172	.337*	.428*	.304*	.482*	.577*	.399	.560	.652*
	FS-2H	.073	.155	.229	.281	.455	.567*	.787*	**.901**	.942	.208*	.386*	.466*	.341*	.531*	.616*	.440*	.616*	.717*
L-H	baseline	.502	.518	.518	.874	.952	.978	.955	.987	**.995**	.604	.608	.605	.888	.930	**.958**	.956	.967	**.980**
	CS-2H	.502	.518	.518	**.876**	.952	**.979**	**.956**	.987	.993	.604	.608	.605	.887	.933	.955*	.956	.967	.979
	OS-2H	.502	.518	.518	.873	.951	.976	**.956**	.986*	.994	.604	.608	.605	.888	.933	.957	.956	.967	.979
	FS-2H	.502	.518	.518	.874	.951	.976*	**.956**	.987	.994	.604	.608	.605*	.888	**.934**	.957	.956	.967	.979
Avg	baseline	**.086**	.197	**.285**	**.313**	**.508**	.605	.803	**.915**	**.951**	.233	.416	**.494**	.374	.574	.654	.489	.685	.788
	CS-2H	.081*	.188*	.269*	.301*	.507	**.621***	.815*	.914	.949	.204*	.384*	.466*	.338*	.532*	.621*	.408*	.587*	.688*
	OS-2H	.084*	**.198**	.281	.309	.506	.614*	**.816***	.914	.949	**.249***	**.429***	.493	**.400***	**.593***	**.668***	**.539***	**.720***	**.804**
	FS-2H	.084	.190*	.273*	.306	.503	.614*	.812*	.913	.948*	.227	.401*	.479*	.364	.557*	.642*	.466*	.646*	.743*

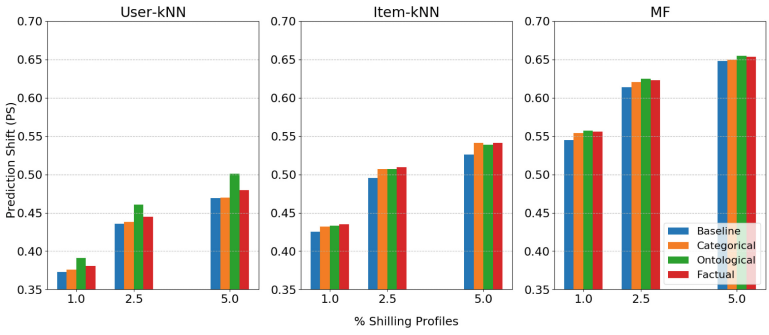
*We denote statistically significant results with * with a p-value less than* 0.05 *using a paired-t-test statistical significance test.*

attack strategy. A possible interpretation is that, since the rationale is to decrease the overall mean rating of all items but the target one, exploiting similarity does not strengthen the approach.

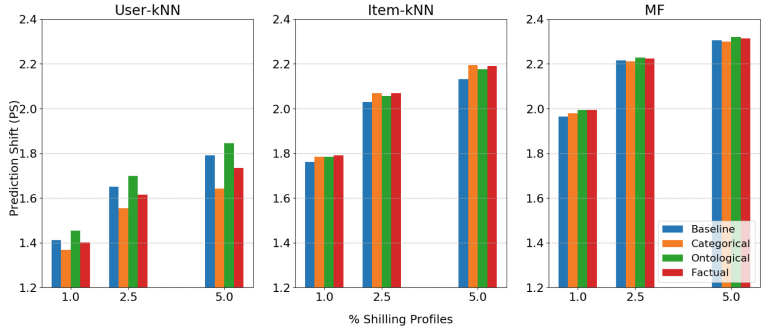
In the informed attacks (i.e., the *average* attack), results show that semantic integration can be a useful source of knowledge. For instance, *Avg-SAShA-OS-2H* improves performance on Item-kNN by 10.2% compared to the baseline.

It is noteworthy that in the semantic variant of the random attack on the movie domain, *Rnd-SAShA-CS-2H*, reaches performance that is comparable with the baseline *average* attack. This observation shows that even an attacker that is not able to access system knowledge can perform powerful attacks by exploiting public (semantic) available knowledge bases.

Analysis of the Impact of Different Semantic Information Types, and Multi-hops Influence. In the previous analysis, we have focused on the effectiveness of *SAShA* strategy irrespective of different types of semantic properties (Sect. 4.1). Table 3 shows that each attack that exploits ontological information is generally the most effective one if we consider single-hop features.



(a) `Yahoo!Movies`



(b) `LibraryThing`

Fig. 2. (a) Prediction Shift on `Yahoo!Movies` for random attacks at single hop. (b) Prediction Shift on `LibraryThing` for random attacks at single hop.

We motivate this finding with the ontological relation between the fake profiles and the target item. Exploiting ontological relations we can compute similarities without the "noisy" factual features. A possible interpretation is that a strong ontological similarity is manifest for humans, but for an autonomous agent it can be "hidden" by the presence of other features. Moreover, the exploitation of items' categorization is particularly effective to attack CF-RS since CF approaches recommend items based on similarities.

Table 3 shows the results for double-hop features. Also in this case, the previous findings are mostly confirmed but for random attacks on `Yahoo!Movies`.

Finally, we focus on the differences between the impact of single-hop and double-hops features. Experimental results show that the variants that consider the second hop have not a big influence on the effectiveness of attacks. In some cases, we observe a worsening of performance as in `LibraryThing`. For instance, the performance of random *SAShA* at double-hops considering ontological features decreases by 13.1% compared to the same configuration at single-hop (when attacking Item-kNN).

5 Conclusion and Open Challenges

In this work, we have proposed a semantic-aware method for attacking collaborative filtering (CF) recommendation models, named *SAShA*, in which we explore the impact of publicly available knowledge graph data to generate fake profiles. We have evaluated *SAShA* on two real-world datasets by extending three baseline Shilling attacks considering different semantic types of features. In detail, we have extended *random*, *love-hate* and *average* attacks by considering Ontological, Categorical and Factual $\mathcal{KG}$ features extracted from `DBpedia`. Experimental evaluation has shown that *SAShA* outperforms baseline attacks. We have performed an extensive set of experiments that show semantic information is a powerful tool to implement effective attacks also when attackers do not have any knowledge of the system under attack. Additionally, we have found that Ontological features are the most effective one, while multi-hops features do not guarantee a significant improvement. We plan to further extend the experimental evaluation of *SAShA* with different sources of knowledge like `Wikidata`. Moreover, we intent to explore the efficacy of semantic information with other state-of-the-art attacks (e.g., by considering deep learning-based techniques), with a focus on possible applications of semantic-based attacks against social networks. Finally, we plan to investigate the possibility to support defensive algorithms that take advantage of semantic knowledge.

Acknowledgments. The authors acknowledge partial support of the following projects: Innonetwork CONTACT, Innonetwork APOLLON, ARS01_00821 FLET4.0, Fincons Smart Digital Solutions for the Creative Industry.

References

1. Aggarwal, C.C.: Attack-resistant recommender systems. In: Recommender Systems, pp. 385–410. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29659-3_12
2. Alam, M., Buscaldi, D., Cochez, M., Osborne, F., Recupero, D.R., Sack, H. (eds.): Proceedings of the Workshop on Deep Learning for Knowledge Graphs (DL4KG2019) Co-located with the 16th Extended Semantic Web Conference 2019 (ESWC 2019). CEUR Workshop Proceedings, Portoroz, Slovenia, 2 June 2019, vol. 2377. CEUR-WS.org (2019)
3. Anelli, V.W., Bellini, V., Di Noia, T., Bruna, W.L., Tomeo, P., Di Sciascio, E.: An analysis on time- and session-aware diversification in recommender systems. In: UMAP, pp. 270–274. ACM (2017)
4. Anelli, V.W., Di Noia, T.: 2nd workshop on knowledge-aware and conversational recommender systems - KaRS. In: CIKM, pp. 3001–3002. ACM (2019)
5. Anelli, V.W., Di Noia, T., Di Sciascio, E., Ragone, A., Trotta, J.: How to make latent factors interpretable by feeding factorization machines with knowledge graphs. In: Ghidini, C., et al. (eds.) ISWC 2019. LNCS, vol. 11778, pp. 38–56. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30793-6_3
6. Backstrom, L., Leskovec, J.: Supervised random walks: predicting and recommending links in social networks. In: Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, 9–12 February 2011, pp. 635–644 (2011)
7. Bhatia, S., Dwivedi, P., Kaur, A.: That's interesting, tell me more! finding descriptive support passages for knowledge graph relationships. In: Vrandečić, D., et al. (eds.) ISWC 2018, Part I. LNCS, vol. 11136, pp. 250–267. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00671-6_15
8. Bhaumik, R., Williams, C., Mobasher, B., Burke, R.: Securing collaborative filtering against malicious attacks through anomaly detection. In: Proceedings of the 4th Workshop on Intelligent Techniques for Web Personalization (ITWP 2006), Boston, vol. 6, p. 10 (2006)
9. Burke, R., O'Mahony, M.P., Hurley, N.J.: Robust collaborative recommendation. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 961–995. Springer, Boston (2015). https://doi.org/10.1007/978-1-4899-7637-6_28
10. Candillier, L., Meyer, F., Boullé, M.: Comparing state-of-the-art collaborative filtering systems. In: Perner, P. (ed.) MLDM 2007. LNCS (LNAI), vol. 4571, pp. 548–562. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73499-4_41
11. Cao, J., Wu, Z., Mao, B., Zhang, Y.: Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system. World Wide Web **16**(5), 729–748 (2012). https://doi.org/10.1007/s11280-012-0164-6
12. Chen, K., Chan, P.P.K., Zhang, F., Li, Q.: Shilling attack based on item popularity and rated item correlation against collaborative filtering. Int. J. Mach. Learn. Cybernet. **10**(7), 1833–1845 (2018). https://doi.org/10.1007/s13042-018-0861-2
13. Cochez, M., et al. (eds.): Proceedings of the First Workshop on Deep Learning for Knowledge Graphs and Semantic Technologies (DL4KGS) Co-located with the 15th Extended Semantic Web Conerence (ESWC 2018). CEUR Workshop Proceedings, Heraklion, Crete, Greece, 4 June 2018, vol. 2106. CEUR-WS.org (2018)
14. Deldjoo, Y., et al.: Movie genome: alleviating new item cold start in movie recommendation. User Model. User-Adap. Inter. **29**(2), 291–343 (2019). https://doi.org/10.1007/s11257-019-09221-y

15. Deldjoo, Y., Di Noia, T., Merra, F.A.: Assessing the impact of a user-item collaborative attack on class of users. In: ImpactRS@RecSys. CEUR Workshop Proceedings, vol. 2462. CEUR-WS.org (2019)
16. Di Noia, T., Magarelli, C., Maurino, A., Palmonari, M., Rula, A.: Using ontology-based data summarization to develop semantics-aware recommender systems. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 128–144. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_9
17. Di Noia, T., Mirizzi, R., Ostuni, V.C., Romito, D., Zanker, M.: Linked open data to support content-based recommender systems. In: Proceedings of the 8th International Conference on Semantic Systems, pp. 1–8. ACM (2012)
18. Di Noia, T., Ostuni, V.C., Tomeo, P., Di Sciascio, E.: SPrank: semantic path-based ranking for top-N recommendations using linked open data. ACM TIST **8**(1), 9:1–9:34 (2016)
19. Gao, L., Yang, H., Wu, J., Zhou, C., Lu, W., Hu, Y.: Recommendation with multi-source heterogeneous information. In: IJCAI, pp. 3378–3384. ijcai.org (2018)
20. Gunes, I., Kaleli, C., Bilge, A., Polat, H.: Shilling attacks against recommender systems: a comprehensive survey. Artif. Intell. Rev. **42**(4), 767–799 (2012). https://doi.org/10.1007/s10462-012-9364-9
21. Hildebrandt, M., et al.: A recommender system for complex real-world applications with nonlinear dependencies and knowledge graph context. In: Hitzler, P., et al. (eds.) ESWC 2019. LNCS, vol. 11503, pp. 179–193. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21348-0_12
22. Hug, N.: Surprise, a Python library for recommender systems (2017). http://surpriselib.com
23. Koren, Y.: Factor in the neighbors: scalable and accurate collaborative filtering. TKDD **4**(1), 1:1–1:24 (2010)
24. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Comput. **42**(8), 30–37 (2009)
25. Lam, S.K., Riedl, J.: Shilling recommender systems for fun and profit. In: WWW, pp. 393–402. ACM (2004)
26. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. **7**(1), 76–80 (2003)
27. Mobasher, B., Burke, R., Bhaumik, R., Williams, C.: Effective attack models for shilling item-based collaborative filtering systems. In: Proceedings of the WebKDD Workshop, pp. 13–23. Citeseer (2005)
28. Mobasher, B., Burke, R.D., Bhaumik, R., Williams, C.: Toward trustworthy recommender systems: an analysis of attack models and algorithm robustness. ACM Trans. Internet Technol. **7**(4), 23 (2007)
29. Ning, X., Karypis, G.: Sparse linear methods with side information for top-n recommendations. In: Cunningham, P., Hurley, N.J., Guy, I., Anand, S.S. (eds.) Sixth ACM Conference on Recommender Systems, RecSys 2012, Dublin, Ireland, 9–13 September 2012, pp. 155–162. ACM (2012)
30. O'Mahony, M.P., Hurley, N.J., Kushmerick, N., Silvestre, G.C.M.: Collaborative recommendation: a robustness analysis. ACM Trans. Internet Technol. **4**(4), 344–377 (2004)
31. Paulheim, H., Fürnkranz, J.: Unsupervised generation of data mining features from linked open data. In: WIMS, pp. 31:1–31:12. ACM (2012)
32. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: an open architecture for collaborative filtering of netnews. In: CSCW, pp. 175–186. ACM (1994)

33. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Shen, V.Y., Saito, N., Lyu, M.R., Zurko, M.E. (eds.) Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, 1–5 May 2001, pp. 285–295. ACM (2001)
34. Shadbolt, N., et al.: Linked open government data: lessons from data.gov.uk. IEEE Intell. Syst. **27**(3), 16–24 (2012)
35. Wang, H., Zhang, F., Xie, X., Guo, M.: DKN: deep knowledge-aware network for news recommendation. In: WWW, pp. 1835–1844. ACM (2018)
36. Wang, X., Wang, D., Xu, C., He, X., Cao, Y., Chua, T.S.: Explainable reasoning over knowledge graphs for recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 5329–5336 (2019)
37. Yu, X., et al.: Personalized entity recommendation: a heterogeneous information network approach. In: WSDM, pp. 283–292. ACM (2014)
38. Zhou, W., Wen, J., Xiong, Q., Gao, M., Zeng, J.: SVM-TIA a shilling attack detection method based on SVM and target item analysis in recommender systems. Neurocomputing **210**, 197–205 (2016)