Local Model-Agnostic Explanations for Black-box Recommender Systems Using Interaction Graphs and Link Prediction Techniques

Marta Caro-Martínez, Guillermo Jiménez-Díaz, Juan A. Recio-García

Department of Software Engineering and Artificial Intelligence, ComplutenseUniversity of Madrid, Madrid (Spain)

Received 16 March 2021 | Accepted 29 September 2021 | Published 14 December 2021

Abstract

Explanations in recommender systems are a requirement to improve users' trust and experience. Traditionally, explanations in recommender systems are derived from their internal data regarding ratings, item features, and user profiles. However, this information is not available in black-box recommender systems that lack sufficient data transparency. This current work proposes a local model-agnostic, explanation-by-example method for recommender systems based on knowledge graphs to leverage this knowledge requirement. It only requires information about the interactions between users and items. Through the proper transformation of these knowledge graphs into item-based and user-based structures, link prediction techniques are applied to find similarities between the nodes and to identify explanatory items for the user's recommendation. Experimental evaluation demonstrates that these knowledge graphs are more effective than classical content-based explanation approaches but have lower information requirements, making them more suitable for black-box recommender systems.

Keywords

Black-box Recommender Systems, Explainable Artificial Intelligence, Graph Knowledge, Graph Representation, Link Prediction Techniques.

DOI: 10.9781/ijimai.2021.12.001

I. Introduction

Recommender systems are one of the essential tools on the Internet today [1]. They are set up on many platforms of e-commerce (Amazon, eBay) and entertainment (Netflix, Spotify), among others. They are necessary to help users find the most interesting products according to their interests. This task can be difficult for them due to the wide selection of products that they can access with new technologies [2]. However, recommender systems may not be as effective as we would expect. Many times, users do not trust this kind of system since they do not understand how a recommender system works and the reasons behind the recommendations [3], [4]. As a consequence, users do not put much attention on them. Because of this, explanations in recommender systems have appeared to solve this problem. Explanation systems try to clarify why a recommendation was provided for a target user [5].

In the literature, several approaches implement recommender systems based on classical techniques. Traditionally, these techniques involve collaborative filtering and content-based systems. Collaborative filtering systems use knowledge extracted from user ratings [6]. In the case of content-based systems, the recommendations are generated with the information about item descriptions and user preferences [7]. However, we must acknowledge that this useful knowledge is not always available when making explanations for a recommendation, sometimes because this information does not exist and other times because we cannot obtain it. For example, we find this problem when we want to provide explanations to black-box recommender system users [8], [9]. A black-box system is a system where users do not know how the method works, and they do not understand it. Therefore, as developers, we cannot access the recommendation process and use it as knowledge source to obtain explanations. [10]. In recent surveys [11], [12], authors focus on the wide use of knowledge representations and reasoning based on graphs to solve complex tasks. Furthermore, the authors classify recommender systems as a type of knowledge-aware application where the integration of knowledge graphs can enhance the reasoning behind the recommendation and, therefore, its interpretability and explainability. This is the reason why we have decided to focus our work on graphs. Our hypothesis is that graph-based explanations can achieve more effective explanations than other classical techniques.

Consequently, we propose a local model-agnostic surrogate explanation system for recommender systems that can be included in this knowledge-aware applications group. A local model-agnostic system is a post-hoc approach that tries to explain a black-box model's behaviour, focusing on a portion of the complete knowledge to provide explanations. In return, users can better understand how the system works because local models are more interpretable than global models, which use all the knowledge available and may be too complex [13], [14], [15], [16]. To tackle the problem of explaining recommendations

* Corresponding author.

E-mail addresses: martcaro@ucm.es (M. Caro-Martínez), gjimenez@ucm.es (G. Jiménez-Díaz), jareciog@ucm.es (J. A. Recio-García).

in black-box systems, we infer and use the knowledge within the interactions between users and items and represent them in graphs to provide explanations. We extend our previous work [17] defining two different approaches according to the entities that represent the graph nodes: the item-based approach and the user-based approach. We apply link prediction techniques on this knowledge to find similarities between the nodes [18], [19]. Among all link prediction techniques, we only consider those with which our proposal becomes a local model. These similarities are used to retrieve explanatory items to show to the user as an explanation for a recommendation. The explanatory items are a set of items with which the target user has interacted before. The user can compare these items with the recommended item and assess if the recommendation is suitable and interesting for them. Accordingly, the explanation is personalized for this user.

Moreover, our graph-based explanation method is independent of the recommender system: it does not require information about how the recommender works in obtaining the explanations, therefore it is suitable as a surrogate model for black-box recommender systems. The experimental evaluation performed in this paper compares the performance of both approaches -the user-based and item-based explanation approaches- to a content-based explanation method, which needs additional information in order to provide explanations. Although we can apply our graph-based methods to any recommendation system, including any black-box recommender, we have provided explanations for matrix factorization recommender systems in the evaluation because they are an excellent example of black-box recommender systems that achieve good results. Results demonstrate that our knowledge graph approach provides better explanations than content-based approaches, while having a remarkably lower information requirement.

The paper is structured as follows. First, Section II shows a review of the literature about explanation approaches in recommender systems and graph-based works. In Section III, we introduce our proposal: the item-based and the user-based graphs. We also present the similarity measures based on link prediction techniques that we have used in our approaches. Afterwards, in Section IV, we present the evaluation that was performed: the dataset, the experimental setup, and a discussion of the results that we obtain. Finally, we present the conclusions of this work and some future research in the last Section V.

II. Related Work

There are many state-of-the-art research studies about recommender systems. Many of them are reviews on this topic, and others are proposals of new techniques to make recommendations [1], [2], [20], [21]. In these works, we can observe two main classical techniques in recommender systems: collaborative filtering and content-based algorithms [22].

On the one hand, recommender systems based on collaborative filtering use the users' ratings to compute the recommendations [23]. On the other hand, the content-based systems take into account the item features and the user profiles to find the most interesting items to be recommended [7]. There are many research works focused on these two techniques and their effectiveness in recommendation tasks [24], [25], [26], [27], [28], [29], [30]. For example, we have some proposals related to our work in the works by Bobadilla et al. [31] and Cordobés de la Calle et al. [32] because they present recommender approaches that use knowledge about the user past interactions without considering rating values. Moreover, the work [32] uses graphs to represent the information and get the recommendations.

Explanation system research is a growing field in studies on recommender systems. When users do not understand why an item is suitable for them according to the recommendation system, they usually stop using these systems or decreasing their use because they do not feel confident with their results [1]. This issue is critical in some contexts and fields such as health care [33]. However, providing explanations in recommendations of daily activities, such as e-commerce or entertainment, is also remarkable because it increases the system's credibility and user's loyalty [34]. As a consequence, it increases users' trust and their use of recommender systems [1]. Therefore, one goal of explanation system research on recommendation systems is to increase the users' trust [35]. There are already many research reviews on and approaches to explanations in recommender systems. In the work by Zhan and Chen [36], we can see that the explanation approaches are based on users, items, and features, traditionally. Therefore, the knowledge extracted from collaborative filtering and content-based systems plays an essential role in classical techniques to provide explanations. The work by Nunes and Jannach [37] describes a taxonomy of explanations in recommender systems. To develop this model, the authors delve into a large amount of research in this field. It was also an essential reference for our previous work [38], where we built a theoretical model to classify explanation approaches and an ontology of explanation approaches in recommender systems: ExRecOnto[1]. We have found other taxonomies about explanation approaches that inspired our previous work, such as the works by Friedrich and Zanker [9] and Papadimitriou et al. [39]. In other works, we can see new proposals for explanation approaches. In the paper by Herlocker et al. [10], we find a classic work that proposes some different types of visualization of explanations in collaborative filtering recommender systems. They use the information extracted from collaborative filtering to provide explanations. Gedikli et al. [40] also present new ways to visualize explanations based on the work of Herlocker et al. In the recent work by Kouki [41], we observe different styles of explanation approaches: user-based, item-based, content-based, social-based, and item popularity.

Moreover, there are more innovative explanation approaches. For example, in the work by Quijano-Sanchez et al. [42], an explanation approach for group recommender systems is proposed based on social information. Andjelkovic et al. [43] describe a music recommender system that includes information about the recommendation through a graph. The approach allows the users to interact with the interface to change their preferences. In the work by Wang et al. [44], the authors describe a new proposal named the Tree-enhanced Embedding Method (TEM). TEM uses models based on embeddings and trees to provide explanations using knowledge extracted from collaborative filtering and latent factors. In our previous work [45], we propose a new way to explain recommendations based on matrix factorization. We use the information from latent factors to build a case-based reasoning [46] system that retrieves explanatory cases [47].

Knowledge representation in graphs and reasoning based on these structures are useful techniques to solve challenging problems [11], [48], [49], [50], [51], [52], [53]. The work by Ji et al. [11] describes a complete review on this topic. They categorize the work on knowledge graphs, and according to the classification proposed in this work, we find four types of graph-based knowledge research works: knowledge acquisition, knowledge representation learning, temporal knowledge graphs and knowledge-aware applications. The last group is the most interesting for this paper because it includes recommender systems as applications that can be enhanced using knowledge graphs.

Link prediction techniques are one of the essential bases of our work. We use the metrics from link prediction techniques to find similarities between our system's items and users. Some research works review these techniques [18], [19], [54], [55] and their application on social networks and recommender systems [56]. There are some approaches that make recommendations and that use graphs with or without link

[1] Available at: https://gaia.fdi.ucm.es/ontologies/#exreconto

prediction techniques. For example, in the work by Chiluka et al. [57], the authors describe an approach based on a user-item graph that employs link prediction techniques to collect the recommendations in User-Generated Content systems (UGCs) such as YouTube or Flickr. In the case of the work in [58], the authors do not use these link prediction techniques. However, they present a graph-based approach that combines content-based and collaborative knowledge for digital libraries. Chen et al. [59] propose a recommender system based on interaction graphs and collaborative filtering. Wang et al. [60] present a new system that uses graph representation to enrich news recommendations. We also find interesting research in the work by Shahmohammadi et al. [61]. The authors define the new concept "collaborative path", which refers to the use of collaborative filtering based on the user interaction background. They use "collaborative path" to create new proximity measures and recommendation algorithms based on link prediction techniques for online social networks, such as Facebook. Another example is the work [62], where the authors employ a bipartite network projection to provide recommendations.

On the topic of explanation approaches, there are a few research works about knowledge graphs. We encounter some recent works that review the role of knowledge graphs in the Explainable Artificial Intelligence (XAI) field, identifying the necessities that they cover [12], [63], [64]. We can also find explanation approaches based on knowledge graphs, which are different from our proposal. With our proposal, we provide explanations-by-example without considering the knowledge from the recommender system, only information about interactions. It requires a minimum amount of knowledge, while other similar systems use additional information. For example, in the work by Barbieri et al. [65], an explanation system using graphs and link prediction techniques is proposed. They provide an explanation using the reason why a link exists. To do this, the authors include latent factors that represent the user's preferences. Xian et al. [66] describe a new method called Policy-Guided Path Reasoning (PGPR), which uses a knowledge graph to generate recommendations. PGPR takes into account real paths in the graph to create explainable recommendations. Therefore, it uses additional information from the white-box recommender system to provide explanations. We also have to mention the work by Wang et al. [67], which introduces a new model Knowledge-aware Path Recurrent Network (KPRN). KPRN also uses knowledge graphs to make recommendations and to collect better results than other models such as collaborative knowledge base embedding or the neural factorization machine. One remarkable feature of the model is that it is also interpretable. Again, this method needs to use data from the recommendations to obtain explanations. However, there is much work to do in the field of explanations for recommender systems using graphs. These structures can represent a wide and varied knowledge, which is likely to be useful to justify recommendations.

III. The Knowledge Graph Explanation System

In this paper, we propose a novel knowledge-light, explanation system for black-box recommender systems. Our method only requires the knowledge inferred from the interactions between users and items within the recommender system to generate example-based explanations. Because our proposal is independent of the recommender process, it is suitable to support black-box recommender systems, where its working and knowledge is not available to obtain explanations. Thus, it can be considered a model-agnostic surrogate method, as depicted in Fig. 1.

Example-based explanations use previous items that the user liked and and similar to the recommended one. Additionally, theses explanations can also present items that users similar to the target user liked. Thus, every explanation is personalized for the target user, who will check the provided recommendation's suitability compared with the explanatory items.

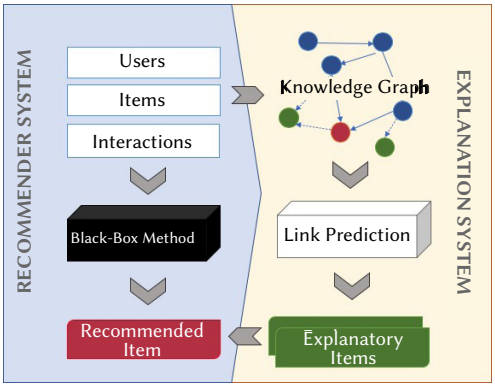

Fig. 1. General overview of the explanation approach using interaction graphs.

It is important to note that our method is designed to be knowledge-light, not requiring any other additional information such as rating values, descriptions, etc. Although additional information may increase the explanation's accuracy, it also increases the dependency on the recommendation algorithm. Our goal is to propose a model-agnostic explanation method with a minimum knowledge requirement but that achieves an acceptable performance compared with other explanation algorithms that require a similar or even higher knowledge level regarding the underlying recommendation process. It is also remarkable that our method proposed is totally independent of the recommendation system. It does not require any knowledge or reasoning that the recommender method uses. Therefore, we can apply our explanation method to any recommendation system. This is the reason why our graph-based explanation system is suitable to be applied on black-box recommender systems.

The general overview of our explanation method is presented in Fig. 1 and is as follows. First, we create knowledge graphs (described in Subsection A) using the interactions performed by the user within the recommender system. We define two different approaches to select the explanatory examples: *item-based* and *user-based* knowledge graphs.

The item-based graph represents connections between items, where links between two nodes are created when at least one user has consumed both items. The weight of the link is the number of users who have interacted with both items. Then, given a recommended item i for target user u, we compute the similarity between i and potential explanatory items (items that u liked previously), which is calculated using link prediction techniques (Subsection B). Finally, the most similar items will be presented as explanatory items. This approach is described in Subsection C.

Alternatively, the user-based graph describes connections between users who have interacted with at least one item in common, where link weights represent the number of common items consumed by the two users. Here, given a target user u, link prediction techniques are used to find other potential similar users that may consume items related to recommendation i. Next, the items already consumed by these similar users and u are aggregated and selected as explanatory items for u. This alternative approach is described in Subsection D.

We have provided the functional description of our method, and the following subsection describes the acquisition process to create the required item-based and user-based knowledge graphs.

A. Knowledge-Graph Acquisition

We can define an interaction for a recommender system as an action that a user has carried out with an item, such as watching a movie, rating a book or buying a product.

The most common interaction used in recommender systems is the rating of items. A rating action can be represented as a tuple $R = (t, u, i, x)$, where t is the timestamp when the interaction was performed, u is the user that went through with the interaction, i is the item with which u interacted, and x is the value associated with the rating. For instance, x could be the rating value (such as "5 stars") with which user u has assessed item i. However, this representation is still valid for any other kind of interaction where x is empty, and no additional information is associated with the interaction (such as "watching a movie"). This way, we assume a "minimal knowledge scenario" for our proposal, as it does not distinguish between positive, negative or neutral interactions. For example, it is positive if the user has rated a movie 4 or 5 stars. In contrast, a user has negatively interacted with a movie if they rated it less than 4 stars. A neutral interaction is defined as when x is empty. However, our model does not make this distinction and represents all of them equally: the x value is not required.

Taking negative interactions into account may seem useless or even harmful, but we think that we may lose important knowledge if we delete them. There are three main goals to include negative interactions in the model:

1. **Help users to find a correlation between the recommnedations and past interactions**. Items, which the target user did not like, have attributes that she may like. The user had at least a minimum interest in this item. Therefore, they could find recommendations similar to it helpful. For example, a user has watched a horror movie that she did not like. It does not mean that she does not like this type of movie necessarily. Maybe this particular movie was not great for her. Therefore, if the system recommends a new horror movie, then the movie that she had watched previously can be a reasonable explanation for this recommendation. She can see why the system provided this recommendation and the connections between the movies according to her preferences.
2. **Help to discard recommendations**. It could be valuable for the target user to discard recommendations that are not interesting. Although the recommender system had used the target user preferences to get a recommendation, this item might not be a good recommendation for her. Users have many different interests, and they need different products depending on their context [42]. For example, a target user wants to watch a movie on Halloween with her friends. They want to watch a horror movie. Although the target user likes Disney movies, these movies are not suitable for her in this context. With negative interactions, she can assess the recommendations better.
3. **Negative interactions are helpful to provide trust and loyalty**. Users need to trust the system, and we can only provide trust if users know how the system works [35]. The target user needs to know why the system provides a recommendation that is not interesting for her. She can understand how the system works, even if the system is mistaken. This information provides trust and loyalty, increasing user satisfaction [34].

Therefore, an interaction represents a relation between entities from *user set U* and *item set I*. This relation can be represented in an adjacency matrix $A = A_{ui}$, where A_{ui} represents if an interaction has occurred between user $u \in U$ and item $i \in I$. If the interaction has occurred, then $A_{ui} = 1$; otherwise, $A_{ui} = 0$, so the link does not exist. The graph built using this adjacency matrix is a bipartite graph: the nodes represent both user (from user set U) and item (from item set I) entities, and the relation is always from u to i. The semantic description of this relation is as simple as "user $u \in U$ has interacted with the item $i \in I$".

However, we can transform this graph by applying a bipartite network projection (Fig. 2) to create two different types of knowledge graphs that will ease our task of generating explanations: an *item graph* and a *user graph*.

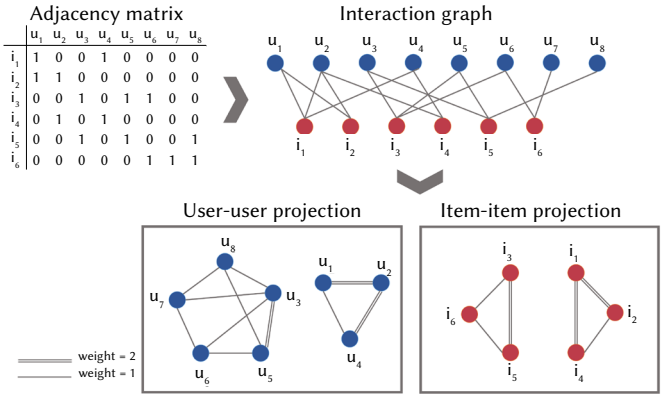

Fig. 2. Transformation of a bipartite graph into a nonbipartite graph through *bipartite network projection*.

On the one hand, in the item-based graph, nodes represent entities from the items set *I*. A link between two items i and j represents that "at least one user in common has interacted with both i and j". The link's weight is the number of common users that interacted with i and j.

On the other hand, the user-based graph only contains entities from user set U. In this graph, a link between two users u and v represents a relation between users, whose semantic description is "u and v have interacted with at least one item in common". As in the previous graph, the weight of the link represents the number of items in common.

These knowledge graphs are the primary building block of our explanation method, as they are used to find the most relevant items to show to the target user as explanatory items. To identify these items, we apply link prediction techniques to compute the similarities between the nodes in the graphs. We have considered and evaluated several similarity metrics from the link prediction literature, which are described in the following section.

B. Link Prediction Metrics

Link prediction techniques are algorithms from social network analysis that predict new links that will appear in a graph [18], [19], [54], [55]. There are several types of metrics to predict these links, but we focus on the similarity-based approaches due to our proposal's nature. The similarity-based metrics are, in turn, divided into four groups: node-based (they use the node properties), neighbour-based (they take into account the features between neighbours of the nodes), path-based (they define paths between the nodes), and walk-based (they use transition probabilities between nodes and neighbours). We choose to stress node-based and neighbour-based approaches because they are local models. These models focus on a local section of the knowledge, which is suitable for providing a concrete explanation without considering the whole knowledge represented in the graph [13], [14], [15], [16]. Local models are more interesting than global ones because they are more interpretable for target users, and we do not need to reach explanatory cases far from our target node. Therefore, the explanatory examples collected with the link prediction methods can be more useful and suitable for our proposal than other global link prediction techniques. The metrics used in the current study are the ones proposed in our previous graph-based recommender and explanation approaches [17], [68], [69], [70]. They are a variation of the classic link prediction metrics, and some of them can be divided into two versions: weighted and unweighted. Although similarity

metrics are commonly measured in the range of [0,1], our approach defines these similarity metrics more as a scoring function to rank similar items.

To clarify the description of the similarity metrics described here, we give some notation:

- $N(i)$ represents the set of neighbours of node i.
- $|N(i)|$ represents the number of neighbours (or node degree) of node i.
- L_{ij} represents the weight w of the link between nodes i and j.
- $W(i) = \sum L_{ix}: x \neq i \in I$ represents the *weighted node degree* of node i, which indicates the sum of the weights of the links directly connected with node i.

The link prediction metrics used in our explanation system are as follows:

Edge Weight (EW). This metric measures the similarity between two nodes as the weight of the link between them. $L_{ij} = 0$ represents that node i and node j are not connected. An unweighted version of this metric exists ($L_{ij} = 1$ if the link exists, 0 otherwise), but we have not used it because it is too simple.

$$EW(i,j) = L_{ij} \tag{1}$$

Common Neighbours (CN). Using this metric, the similarity between two nodes is the number of neighbours they have in common. The greater the intersection of the neighbour sets of any two nodes is, the greater the chance of a future association between them. Weighted Common Neighbours (WCN) is the weighted version of this metric.

$$CN(i,j) = |N(i) \cap N(j)| \tag{2}$$

$$WCN(i,j) = \sum_{z \in N(i) \cap N(j)} L_{iz} + L_{jz} \tag{3}$$

Jaccard Neighbours (JN). This metric is an improvement of $CN(i,j)$, as it measures the number of common neighbours of i and j compared with the number of total neighbours of both nodes. It does not have a weighted metric version.

$$JN(i,j) = \frac{|N(i) \cap N(j)|}{|N(i) \cup N(j)|} \tag{4}$$

Adar/Adamic (AA). This metric also measures the intersection of neighbour sets of two nodes in the graph, emphasizing the smaller overlap. The weighted version of this metric is Weighted Adar/Adamic (WAA).

$$AA(i,j) = \sum_{z \in N(i) \cap N(j)} \frac{1}{log|N(z)|} \tag{5}$$

$$WAA(i,j) = \sum_{z \in N(i) \cap N(j)} \frac{L_{iz} + L_{jz}}{log(1 + W(z))} \tag{6}$$

Preferential Attachment (PA). This metric is based on the consideration that nodes create links, with higher probability, with those nodes that already have a larger number of links. The probability of creating a link between nodes i and j is computed as the product of the degree of nodes i and j; therefore, the higher the degree of both nodes is, the higher is the probability of linking. This metric has the drawback of leading to high probability values for highly connected nodes to the detriment of the less connected nodes in the network. Weighted Preferential Attachment (WPA) is the weighted version of this metric. It is an improvement of PA in which the link weights are taken into account when computing the degree of nodes i and j.

$$PA(i,j) = |N(i)| \cdot |N(j)| \tag{7}$$

$$WPA(i,j) = W(i) \cdot W(j) \tag{8}$$

We have described the link prediction techniques, and the following sections present the contributions of this paper.

C. The Item-Based Explanation Method

The explanation process is defined for a target user u, who accepts the recommendation of item i. The goal of our system is to retrieve the best list of explanatory items $E = [e_1, e_2, ..., e_k]$ that helps u understand why the black-box recommender system recommended i. Therefore, this explanation-by-example method consists of displaying items similar to i that u previously interacted with.

This method uses the item-based knowledge graph to find explanatory examples. We define our item graph as $G_i = \langle I, L \rangle$, where I is the set of nodes representing the items, and L is the set of links that connect the nodes. We can define links as $L = \{(i,j,w) | i \neq j \in I\}$. Nodes i and j represent the items connected by a link, and w is the weight of the link. As described before, the weight w is the number of common different users that have interacted with both items. Due to the graph's high density, we decided to apply a preliminary filter to remove low representative links. Therefore, we define a threshold parameter δ_w to remove all links whose weight w is lower than its value.

The process for creating the list of explanatory items E is as follows (Fig. 3):

- **Step 1**. We build a similarity matrix S with the similarity scores between all nodes in the graph using the link prediction metric lp. Thus, the $S(i,j) = lp(i,j)$ value corresponds to the similarity between items i and j computed by the link prediction metric lp, as defined in Section B.
- **Step 2.** We build a set of candidate explanatory items $E' = \{(e_1, s_1), (e_2, s_2), ..., (e_n, s_n)\}$ that includes the items most similar to i using the similarity values in S. Value $s_x = S(i, e_x)$ represents the similarity between i and the explanatory item e_x.
- **Step 3**. We filter the candidate explanatory items already consumed by the target user by removing from E' all items in this set with which u has not interacted yet.
- **Step 4**. We rank E' in decreasing order using the similarity scores of the items (s_x). Finally, the top k items in this sorted list are returned as the explanatory items $E = [e_1, e_2, ..., e_k]$ for recommendation i and target user u.

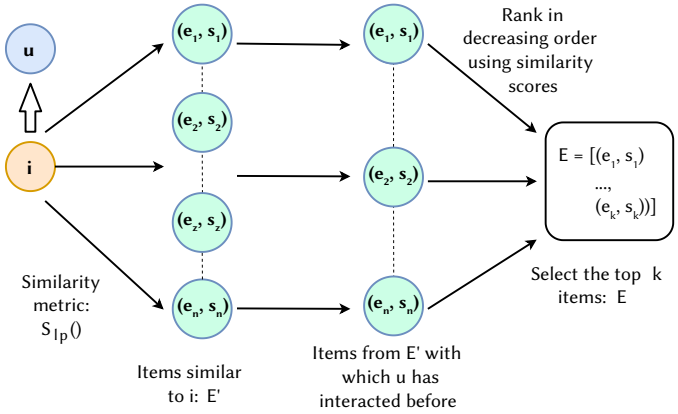

Fig. 3. Process of selecting the explanatory items with our item-based approach.

D. The User-Based Explanation Method

This alternative method is based on the user-to-user graph. The graph $G_u = \langle U, L \rangle$ represents the set of the user entities U as nodes, and the set of the links L is noted as $L = \{(u,v,w) | u \neq v \in U\}$. Analogously to the previous section, a link connects two nodes u and v when they

have interacted with at least one item in common. The weight of link w is the number of items with which both have interacted. Again, as it is a high-density graph, we apply a threshold parameter δ_w to remove all links whose weight w is lower than its value.

The process for creating the list of explanatory items E for target user u and a recommended item i is as follows (Fig. 4):

- **Step 1**. We build a similarity matrix S that stores all similarity scores between every pair of nodes of the graph using the link prediction metrics lp. Therefore, the $S(u,v) = lp(u,v)$ value corresponds to the similarity value between users u and v using the similarity metric lp. Again, lp is one of the link prediction metrics proposed in Section B.
- **Step 2**. From S, we build the set $V = \{v_1, v_2, ..., v_n)\}$ containing the n most similar users to u.
- **Step 3**. For every related user v, we obtain the set of items that has interacted with: $I_v = \{(e_1, s), (e_2, s), ..., (e_m, s\}$. Here, the similarity s associated with each item is the similarity between the target user u and the user v, that is, $s = S(u, v)$. Therefore, all of the items in I_v have the same similarity value.
- **Step 4**. Next, we build the set of candidate items for the target user by joining the items that similar users have interacted with: $E' = \bigcup_{v \in V} I_v$. Duplicated items are stored only once, and their associated similarity is the highest value found among all repetitions in the set.
- **Step 5**. We filter E' by removing all of the items that the target user has not interacted with yet.
- **Step 6**. Finally, the list of explanatory items $E = [e_1, e_2, ..., e_k]$ is created by sorting E' in decreasing order according to the similarity values associated with each item, and selecting the first k elements.

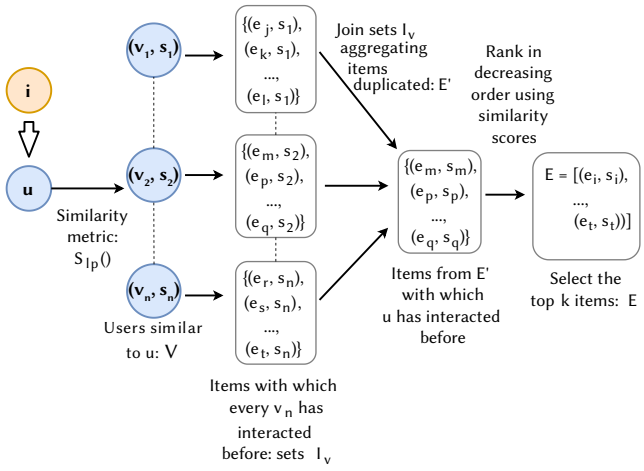

Fig. 4. Process of getting explanation items with our user-based approach.

We have presented our two surrogate methods, and the following section presents their experimental evaluation.

IV. Evaluation

Our experimental evaluation goal is to demonstrate that our method achieves similar performance as other explanatory approaches while having a lower knowledge requirement. The code implementation of the evaluation carried out with the graph-based methods is on a GitHub repository[2] to make more accessible the reproducibility of our proposal.

[2] https://github.com/martcaro/GraphBasedExplanations

In the literature, we can encounter that the items descriptions and item features are one of the knowledge used the most to generate explanations in recommender systems [36]. Therefore, we designed a comparative evaluation against a content-based system [46] that uses the item descriptions to find explanatory examples. Similar to our methods, this system will provide explanatory cases for a target user and a recommended item. However, it requires information about the features of the items in order to retrieve the explanatory examples. In addition, it is a global model because it uses all knowledge available. The evaluation hypothesis is that our proposal can find explanatory items that are as useful as the ones retrieved by the content-based system while using less information about items, only a portion of the knowledge represented by the interaction graphs.

We also need to use a recommender system to evaluate the models. Although we do not use knowledge about the recommendation process, we need to have a list of recommendations based on actual data. The models evaluated will try to explain this list. As we are dealing with model-agnostic surrogate models, we do not need information from the recommendation process and we can provide explanations to any recommender system. However, we consider that evaluating our model with a black-box recommender is more interesting. Because we do not need information about the recommendation process, we can validate our hypothesis that the graph-based explanation methods can retrieve useful explanatory examples without using rating values, descriptions or other additional information. Taking this into account, we decided to use a matrix factorization recommender system in our evaluation. Matrix factorization is one of the most e˙ective algorithms to make recommendations nowadays [47]. Nevertheless, it is not transparent for target users, and it is not easy to understand why an item has been recommended to them. Hence, matrix factorization recommendations are a good example that fits the need for an explanation system. In the evaluation process described next, we use the recommender as a black-box system, and our graph-based methods do not use any other information from the recommender to obtain the explanatory examples. Furthermore, we have used a dataset from the movie domain, as it is one of the most widely used to evaluate recommender systems. Next, we explain the evaluation process. In Section A, we describe the dataset that we have used to perform the evaluation. In the following Section B, we relate the experimental process itself. Finally, Section C discusses the results of the evaluation.

A. Dataset

In the experiment, we combine two different datasets for building our evaluation test set. On the one hand, we use the 100K MovieLens dataset[3] because it is a common choice to evaluate recommender systems. The MovieLens dataset contains 100K ratings in a set of tuples $R = (t, u, i, x)$, where u is the user who has watched movie i, t is the timestamp when u has rated i and x is the rating provided by u for i. Only the $\langle u, i \rangle$ pair is the information required by our explanation method. However, this evaluation aims to demonstrate that the quality of the explanations is similar to that for a content-based approach with a higher knowledge requirement. Therefore, we require an additional dataset with extra information about the recommended items. The chosen dataset is the IMDB dataset[4]. This dataset contains feature information about 5,000 movies such as genres, actors, and directors. This information will complement the MovieLens dataset to implement the content-based explanation system. However, not all movies in the MovieLens dataset appear in the IMDB one. Hence, we filter the MovieLens dataset to retain the movies that also appear in the IMDB dataset. We denote the resulting dataset as D. We divide D into the training set D_t with 90% of the interactions and the test set D_e with the remaining 10%.

[3] https://grouplens.org/datasets/movielens/100k/

[4] https://www.imdb.com/

Before performing the evaluation, we run an exploratory analysis on the dataset, following the model proposed by Dooms [71] that we use in previous works [17], [68]. Table I shows the results of this analysis.

TABLE I. Analysis of the Datasets Used in the Evaluation. *ML* is the Original MovieLens Dataset With 100K Interactions

Metric	ML	D	D_t	D_e	B_e
# Ratings	100,000	11,477	10,330	1,147	280
# Items	1,682	164	164	145	109
# Users	943	587	584	394	134
Density	0.06	0.12	0.11	0.02	0.02
Items					
Maximum # ratings per item	583	329	305	30	10
Median # ratings per item	27	43.5	39	5	2
Average # ratings per item	59.45	69.98	62.99	7.91	2.57
Minimum # ratings per item	1	1	1	1	1
Users					
Maximum # ratings per user	737	128	113	15	11
Median # ratings per user	65	12	11	2	1
Average # ratings per user	106.05	19.55	17.69	2.91	2.09
Minimum # ratings per user	20	1	1	1	1
Ratings					
% Ratings ≥4	55.38	52.54	52.66	51.44	37.50
% Ratings <4	44.62	47.46	47.34	48.56	62.50

From this analysis, we found a significant bias in dataset D_e: it is unbalanced regarding the number of items associated with each rating value. As a consequence, we decided to create a stratified test set, B_e. B_e avoids this bias because it contains the same number of items (35) for each rating value. We have selected this amount of items because it is the minimal amount for a rating value (2.5 stars).

B. Experimental Setup

The experimental process starts by building the graph-based and content-based explanation methods on the training set. We implemented several versions of the graph-based methods regarding the similarity metrics described in Section B: AA, CN, EW, JN, PA, WAA, WCN, and WPA. We also configured the threshold $\delta_w = 5$ since we considered this value sufficient to reduce the density without removing essential knowledge.

The content-based explanation system retrieves the items most similar to the recommended item i_r taking into account the movie features in the IMDB dataset. In this case, we evaluate Euclidean, Cosine, and Jaccard methods as similarity metrics. Both explanation methods generate the list E of explanatory items sorted in decreasing order. The list size is adjusted by the k parameter, which has been evaluated within the range $k \in [1,10]$.

The experimental process continues by measuring the effectiveness of both explanation systems against two test sets: D_e and B_e. Each evaluation is repeated 100 times, where we randomly obtain B_e from D_e for each iteration. The explanation methods' effectiveness is evaluated from the similarity between the ratings of recommended item i_r and the explanatory items in E_k. To do this, we employ the Root Mean Square Error (RMSE) metric to compare the rating for i_r predicted by the matrix factorization recommender system and the actual ratings of the explanatory items in E_k retrieved by each method. It is important to note that, for each k, we have removed the users who do not interact with at least k movies in the test set. Thus, if a target user u_t has only rated four movies, then she is not suitable to be evaluated when the list of explanatory items has $k \geq 5$.

C. Discussion

Table II and Table III report the results obtained with the original dataset and the stratified dataset, respectively. For each table, we show the performance of the graph-based and content-based methods. It is remarkable that there are similar scores for both datasets, but not among the methods being evaluated.

TABLE II. Results of the Evaluation With the Original Test Set (D_e). The Column Values Correspond to k From 1 to 10. The Best Results Among Similarity Metrics Are in Bold. The Best Evaluation Metric Values Are Underlined

	1	2	3	4	5	6	7	8	9	10
I - AA	1.095	0.865	0.792	0.769	0.747	0.723	0.702	0.686	0.678	0.664
I - CN	1.035	0.821	0.750	0.713	0.683	0.666	0.649	0.640	0.636	0.635
I - EW	1.087	0.878	0.797	0.754	0.734	0.706	0.679	0.666	0.651	0.639
I - JN	**0.961**	**0.734**	**0.658**	**0.624**	**0.599**	**0.573**	**0.562**	**0.549**	**0.540**	**0.534**
I - PA	1.126	0.897	0.806	0.789	0.782	0.762	0.741	0.728	0.718	0.706
I - WAA	1.113	0.968	0.908	0.855	0.822	0.794	0.775	0.747	0.723	0.705
I - WCN	1.113	0.968	0.907	0.855	0.821	0.796	0.778	0.749	0.723	0.704
I - WPA	1.115	0.970	0.910	0.852	0.821	0.793	0.779	0.747	0.723	0.703
U - AA	0.865	0.798	0.759	0.741	0.720	0.705	0.696	0.690	0.688	0.685
U - CN	0.877	0.788	0.758	**0.734**	**0.715**	0.701	**0.692**	0.688	0.685	0.681
U - EW	0.871	**0.778**	**0.744**	0.736	0.718	**0.699**	0.695	**0.684**	0.688	0.684
U - JN	0.874	0.805	0.777	0.745	0.728	0.711	0.704	0.703	0.695	0.686
U - PA	0.865	0.791	0.752	0.742	0.720	0.706	0.695	0.690	0.689	0.685
U - WAA	**0.864**	0.793	0.753	0.743	0.720	0.706	0.695	0.690	0.690	0.684
U - WCN	**0.864**	0.793	0.753	0.743	0.720	0.706	0.695	0.690	0.690	0.684
U - WPA	0.865	0.791	0.752	0.743	0.720	0.706	0.695	0.690	0.690	0.684
Cosine	0.973	1.036	1.064	**1.078**	**1.087**	1.100	1.101	1.104	1.108	1.111
Euclidean	**0.966**	**1.032**	**1.063**	1.079	1.092	**1.092**	**1.099**	**1.100**	**1.102**	**1.105**
Jaccard	0.974	1.037	1.064	**1.078**	**1.087**	1.099	1.101	1.104	1.109	1.111

TABLE III. Results of the Evaluation With the Stratified Test Set (B_e). The Column Values Correspond to k From 1 to 10. The Best Results Among Similarity Metrics Are in Bold. The Best Evaluation Metric Values Are Underlined

	1	2	3	4	5	6	7	8	9	10
I - AA	1.164	0.947	0.861	0.839	0.817	0.796	0.774	0.764	0.754	0.738
I - CN	1.084	0.893	0.811	0.761	0.739	0.721	0.696	0.695	0.695	0.698
I - EW	1.142	0.938	0.863	0.827	0.802	0.766	0.743	0.736	0.714	0.702
I - JN	**1.004**	**0.748**	**0.673**	**0.643**	**0.629**	**0.597**	**0.584**	**0.573**	**0.566**	**0.566**
I - PA	1.182	1.000	0.902	0.883	0.873	0.852	0.835	0.819	0.809	0.793
I - WAA	1.156	1.061	1.002	0.944	0.922	0.894	0.874	0.838	0.810	0.790
I - WCN	1.149	1.054	0.994	0.937	0.917	0.890	0.873	0.837	0.807	0.787
I - WPA	1.152	1.065	1.011	0.940	0.921	0.889	0.873	0.837	0.812	0.789
U - AA	0.883	0.830	0.794	0.765	0.741	0.731	0.717	0.716	0.715	0.713
U - CN	0.898	0.820	0.787	0.758	0.740	0.727	0.715	0.715	0.713	0.710
U - EW	0.888	**0.804**	**0.758**	**0.749**	**0.733**	**0.710**	**0.707**	**0.698**	**0.706**	**0.704**
U - JN	0.881	0.833	0.811	0.771	0.755	0.738	0.733	0.733	0.725	0.716
U - PA	0.883	0.820	0.778	0.765	0.740	0.727	0.709	0.708	0.711	0.708
U - WAA	0.879	0.821	0.779	0.766	0.740	0.728	0.710	0.709	0.712	0.708
U - WCN	0.881	0.823	0.779	0.767	0.741	0.729	0.711	0.710	0.713	0.709
U - WPA	0.881	0.820	0.778	0.767	0.742	0.730	0.712	0.712	0.715	0.711
Cosine	1.117	1.130	1.125	1.125	1.125	1.120	1.129	1.125	1.121	1.121
Euclidean	1.090	1.105	1.102	1.110	1.105	1.106	1.106	1.111	1.112	1.112
Jaccard	**1.052**	**1.052**	**1.054**	**1.067**	**1.084**	**1.097**	**1.100**	**1.099**	**1.102**	**1.100**

In the Table II, we can observe the differences among the similarity metrics used in the evaluation and their performance when applied to the original dataset D_e.

For the item-based method, the scores are similar regarding the k parameter. However, outlier values correspond to the JN similarity

metric. With this similarity metric, we come across a better result for the RMSE for all values of k. JN always improves the performance of the other similarity metrics with a difference of approximately 10% to 20%. Our explanation for this behaviour is that JN considers the number of common neighbours compared with the number of total neighbours to obtain the items most similar to the recommended one. This indicates that this metric considers the similarity and the diversity of the sets. It may also indicate that the knowledge from negative interactions is useful for explaining recommendations.

However, we do not find this pattern in the user-based approach. The results of all the metrics are very similar. Moreover, there is no obvious best similarity metric. When $k = 1$, WAA and WCN are the metrics that perform better. They take into account the weight of the links. Therefore, they achieve higher performance when $k = 1$ because they exploit that information. For the rest of the values of k, the best similarity metric varies between CN and EW. With EW, we achieve the best results four times, while with CN, we reach the lower scores for five setups. Therefore, we can conclude that CN may be the best similarity metric for the user graph-based method. We can conclude that EW provides better results because it considers the links' weight to obtain explanatory items.

Regarding the content-based results, the best similarity metric is also clear. It is not as obvious as in the item-based method, but we can conclude that we obtain the best results with the Euclidean distance. Moreover, we can observe that the difference between the results is even lower than the difference found for the previous methods. These di˙erences are not significant.

We can obtain additional conclusions by comparing the best scores. This analysis is shown in Fig. 5. In this figure, we report the results of each approach with its best similarity metric: JN for the item-based approach, CN for the user-based approach, and the Euclidean distance for the content-based approach. On the one hand, we can see a heterogeneous behaviour regarding the k parameter. In the case of the content-based system, the RMSE value becomes higher with increasing k. However, in the case of the graph-based approaches, the performance improves. The error value stabilizes because the algorithm retrieves a larger amount of explanatory items; therefore, it is more diÿcult to make a significant mistake. On the other hand, the best performance values are always achieved by graph-based methods. When $k = 1$, the best results are achieved with the user-based proposal. In the rest of the cases, we have achieved the best results with the item-based proposal. We can also conclude that the item-based approach performs better because the recovery of similar items is straightforward and target users are familiarised with items with which they have interacted before. Moreover, explanatory examples using a justification based on similar users who they do not know can be less helpful.

Table III reports the results of the evaluation with the stratified dataset B_e. The trend in the results is similar to the trend that we come across in Table II. With the stratified dataset, we obtain worse results, but the difference is not remarkable: the bias does not have a relevant impact. However, the results are slightly worse with the stratified dataset because we remove the bias that we found in the original dataset.

In the item-based approach, we can see that JN is the similarity metric that performs better. Therefore, the bias of the dataset does not change the comparison among the imilarity metrics. For the user-based method, the discussion is also very similar to the previous one. We can observe that EW is the best similarity metric for all values of k except for $k = 1$. With $k = 1$, WCN, WPA, and JN achieve the best performance with the same value (0.881). Here, we can conclude that WCN acts as the best similarity metric with both the original and stratified datasets. For the rest of the values of k, EW is still the best metric, but CN worsens, although it achieves suÿcient results. Therefore, we can say that the similarity metrics that work with knowledge about the weight provide better results with the user-based approach. In the case of the content-based system, the conclusions change. We observe that the best result is achieved with the Jaccard similarity metric for all values of k.

To compare the performance of the three approaches when applied to the stratified dataset B_e, we have created Fig. 6. We have also decided to represent in this chart the best similarity metric for each method: the JN similarity metric for the item-based method, EW for the user-based method, and Jaccard for the content-based method. Again, we can see the same trend that we saw with the original dataset. The chart shapes are almost the same, although we do not achieve the best results with the same similarity metrics for the user-based approach and content-based system.

Considering the results analysed, we can also discuss the parameters chosen in the evaluation. On the one hand, we split the dataset into the training set and test set with the 90% and 10% of the interactions, respectively. It can a˙ect the results in terms of performance. If we decrease the number of interactions in the training set and increment the interactions in the test set, we may get lower RMSE values. The explanation models' performance depends on the amount of knowledge that we use to build them. It can a˙ects to both graph models and content-based methods. In the graph-based models, we will have a graph with a lower amount of nodes and links. Therefore it is more challenging to get a correct answer if we have a less amount of candidates. Equally, the content-based method

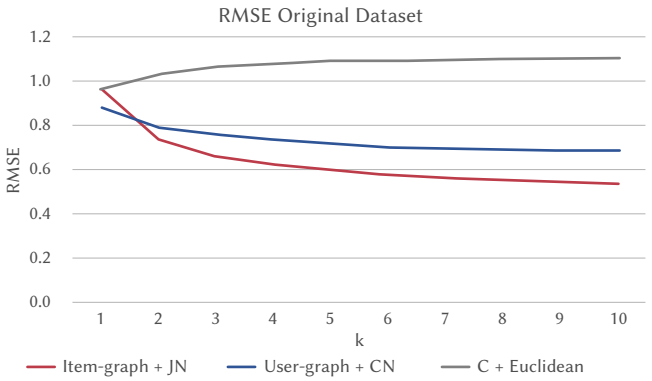

Fig. 5. Chart which represents the results got with the original dataset (D_e). For each approach studied, we have chosen the similarity metrics which performed better. In the axis Y, we represent the RMSE value. We consider the number of explanatory items retrieved in the axis X.

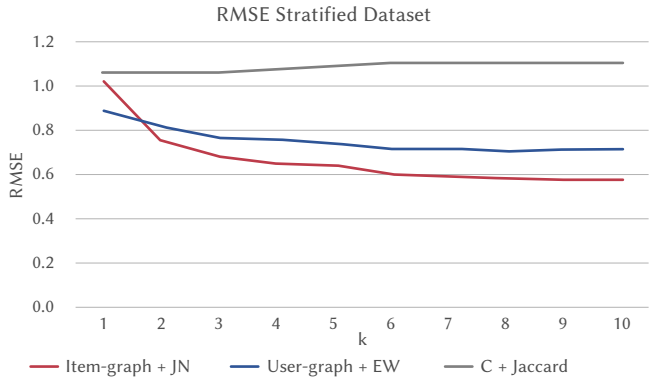

Fig. 6. The results got with the stratified dataset (B_e). For each approach, we have chosen the similarity metrics which performed better. In the axis Y, we represent the RMSE value. We consider the number of explanatory items retrieved in the axis X.

would have a smaller list of explanation example candidates. Then, the probabilities of finding similar items according to their attributes are lower. On the other hand, we delete the dataset's bias removing all the ratings whose value is lower than 2.5 stars. We have shown in the results above that, removing the bias, we can get better results. However, we did not analyse how the models' behaviour is if we had chosen other minimal value. When choosing a lower value, our theory is that we will get similar values to the evaluation carried out with the original dataset because the bias would not be removed. However, using higher values than 2.5, we would obtain worse results because we would be deleting knowledge.

Finally, we have decided to show the percentage of improvement of the graphs concerning the content-based system in Table IV. The two first rows correspond to the comparison between the item-based model using JN and the user-based model using CN with the content-based system using Euclidean distance, which are the models that performed the best in Table II. The last two rows show the comparison between the item-based model using JN and the user-based model using EW with the content-based system using Jaccard, which are the models that performed the best in Table III. We can see that the graph models enhance the performance of the explanations, becoming 50% better than the content-based ones.

TABLE IV. Percentage of Improvement of the Graph-based Methods Regarding the Content-based Model. The Bestpercentage of Improvement in Every Row Is Marked in Bold

	1	2	3	4	5	6	7	8	9	10
I-JN (D_e)	0.49	28.83	38.09	42.18	45.17	47.50	48.83	50.10	50.99	**51.64**
U-CN (D_e)	9.26	23.65	28.67	31.94	34.50	35.83	36.99	37.46	37.84	**38.35**
I-JN (B_e)	4.56	28.90	36.15	39.74	41.97	45.58	46.91	47.86	**48.64**	48.55
U-EW (B_e)	15.56	23.54	28.10	29.82	32.34	35.29	35.72	**36.45**	35.91	36.03

As a conclusion of this evaluation, the previous discussion validates our hypothesis. We achieve better results with the graph-based methods with both the original and stratified datasets than with the content-based method, regardless of the type of graph or the similarity metric used. Thus, we can consider that our graph-based proposals perform better than content-based approaches to provide explanations using a less amount of knowledge. Furthermore, we can conclude that the item-based approach performs better because it finds the explanatory items in a straightforward way.

V. Conclusions and Future Work

The current work proposes a novel local, model-agnostic, surrogate method to provide explanations for black-box recommender systems using knowledge graphs. This proposal is an alternative solution for when classical explanation techniques cannot be applied due to their requirements regarding the recommender system's input data or internal behaviour. Traditionally, these techniques involve collaborative filtering that requires ratings as the input knowledge or content-based methods that take into account item features or user profiles. However, in many scenarios, the knowledge required by these techniques is not available.

The minimum knowledge that can be obtained from a recommender system is the previous interactions between users and items. In this work, we propose only to use this knowledge to implement a surrogate explanation-by-example method for recommender systems. This proposal does not need information about ratings or descriptions or any other additional knowledge from the recommender system. Therefore, our proposal is suitable to support any type of recommender system, including black-box recommenders whose information is not available to obtain explanations. We hypothesize that we can provide explanations for black-box recommender systems using a minimum amount of knowledge while achieving the same or even better performance than the classical techniques.

We represent the interaction knowledge as a graph, where nodes are users and items and the links represent that the user has interacted with the item. Then, we apply a bipartite network projection obtaining two di˙erent knowledge graphs: an item-based graph and a user-based graph. The item-based graph has items as nodes, and their connections represent the number of users that have interacted with both items. Alternatively, the user-based graph represents users as nodes, and the weight of the link is the number of items with which the users have both interacted. Thus, we have two di˙erent graph structures to provide explanations.

In our method, we apply link prediction techniques to find similar nodes that lead to the discovery of explanatory examples. It is important to note that these link prediction techniques turn our approach into a local model. This implies that our approach is easier to interpret for target users.

One of this work's major novelties is that we consider all interactions performed by users (positive, negative and neutral). As a consequence, the explanation examples provided to the target users can be items that they did not like; therefore, they can decide better if the recommended item is of interest to them or not.

From the item-based graph, the identification of the explanatory examples is very straightforward, directly applying link prediction metrics in order to find the items most similar to the recommended one. Then, we filter these items, removing the ones with which the target user has not interacted yet. In the case of the user-based approach, the process is slightly more complicated. We apply the similarity metrics to find the users most similar to the target user. Then, we compute the items with which this set of users has interacted, removing those the target user has already interacted with. These items will be the explanatory examples to show to the target user.

Therefore, the explanation is personalized for each target user. However, it is important to note that our approach has the cold start problem, similar to many recommender and explanation systems. If there are not su˙cient interactions, then we cannot provide personalized explanations. Solving this problem could be a future line of research.

To validate our method, we performed an experimental evaluation. Its goal was to compare our method's performance with that of a global classical explanation-by-example technique with a higher knowledge requirement, that is, a content-based explanation system.

The evaluation dataset was created from the MovieLens and IMDB datasets. We used the RMSE metric to compare the performance achieved with the three approaches: item-based graph, user-based graph, and content-based. After a complete analysis of the evaluation results, we conclude that the hypothesis is correct, as the graph-based approaches achieve a higher performance than the content-based approach while requiring a lower level of knowledge. Furthermore, the more the list of explanatory items grows, the better the graph-based system's performance, in contrast with the behaviour of the content-based system. Globally, the item-based graph seems to be the most effective method when configured with the Jaccard Neighbours similarity metric.

For future work, we can outline some research areas, apart from solving the cold start problem. We want to validate our hypothesis with different datasets from other domains. For example, we believe that recommenders for the music domain can take advantage of our approach because their datasets usually lack ratings. We could compare our approaches with additional techniques, such as collaborative filtering or machine learning techniques, to confirm the hypothesis

that we verified in this work. We can also evaluate our models applying our explanation models to another kind of black-box recommender systems, apart from matrix factorization. Another research area is to apply new aggregation methods to the graph-based approaches, which we have already performed in previous recommender system proposals [68]. We can also use global link prediction techniques and compare their performances in our graph-based methods with this paper's results. Another essential area for future work is to evaluate with real users because they can provide a more accurate opinion about the graph-based approaches' effectiveness. Moreover, they can provide an analysis regarding their explanation goals, such as user trust or user satisfaction. To perform this evaluation, we would need to develop a visualization method for the explanations. For example, we could provide explanations based on textual justifications or more innovative and visual interfaces that use graph representation.

Founding

Supported by the UCM (Research Group 921330), the Spanish Committee of Economy and Competitiveness (TIN2017-87330-R) and the funding provided by Banco Santander at UCM (CT42/18-CT43/18).

References

[1] C. C. Aggarwal, *et al.*, *Recommender systems*. Springer, 2016.

[2] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, "Recommender systems survey," *Knowledge-based systems*, vol. 46, pp. 109–132, 2013.

[3] D. Jannach, M. Jugovac, I. Nunes, "Explanations and user control in recommender systems," in *Personalized Human-Computer Interaction*, De Gruyter Oldenbourg, 2019, pp. 133–156.

[4] N. Tintarev, J. Masthoff, "A survey of explanations in recommender systems," in *2007 IEEE 23rd international conference on data engineering workshop*, 2007, pp. 801– 810, IEEE.

[5] R. Sharma, S. Ray, "Explanations in recommender systems: an overview," *International Journal of Business Information Systems*, vol. 23, no. 2, pp. 248–262, 2016, doi: 10.1504/IJBIS.2016.078909.

[6] M. D. Ekstrand, J. T. Riedl, J. A. Konstan, *et al.*, "Collaborative filtering recommender systems," *Foundations and Trends® in Human–Computer Interaction*, vol. 4, no. 2, pp. 81–173, 2011, doi: 10.1561/1100000009.

[7] P. Lops, M. De Gemmis, G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender systems handbook*, Springer, 2011, pp. 73– 105.

[8] R. Sinha, K. Swearingen, "The role of transparency in recommender systems," in *CHI'02 extended abstracts on Human factors in computing systems*, 2002, pp. 830–831, ACM.

[9] G. Friedrich, M. Zanker, "A taxonomy for generating explanations in recommender systems," *AI Magazine*, vol. 32, no. 3, pp. 90–98, 2011.

[10] J. L. Herlocker, J. A. Konstan, J. Riedl, "Explaining collaborative filtering recommendations," in *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, 2000, pp. 241–250, ACM.

[11] S. Ji, S. Pan, E. Cambria, P. Marttinen, S. Y. Philip, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE Transactions on Neural Networks and Learning Systems*, 2021, doi: 10.1109/TNNLS.2021.3070843.

[12] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, Q. He, "A survey on knowledge graph- based recommender systems," *IEEE Transactions on Knowledge and Data Engineering*, 2020, doi: 10.1109/TKDE.2020.3028705.

[13] M. T. Ribeiro, S. Singh, C. Guestrin, ""why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[14] J. Singh, A. Anand, "Exs: Explainable search using local model agnostic interpretability," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019, pp. 770–773.

[15] V. Arya, R. K. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S. C. Hoffman, S. Houde, Q. V. Liao, R. Luss, A. Mojsilovic, *et al.*, "One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques," 2019.

[16] M. T. Ribeiro, S. Singh, C. Guestrin, "Anchors: High- precision model-agnostic explanations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[17] M. Caro-Martinez, J. A. Recio-Garcia, G. Jimenez-Diaz, "An algorithm independent case-based explanation approach for recommender systems using interaction graphs," in *International Conference on Case-Based Reasoning*, 2019, pp. 17–32, Springer.

[18] P. Wang, B. Xu, Y. Wu, X. Zhou, "Link prediction in social networks: the state-of-the-art," *Science China Information Sciences*, vol. 58, no. 1, pp. 1–38, 2015.

[19] D. Liben-Nowell, J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[20] F. Ricci, L. Rokach, B. Shapira, "Introduction to recommender systems handbook," in *Recommender systems handbook*, Springer, 2011, pp. 1–35.

[21] F. Isinkaye, Y. Folajimi, B. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261–273, 2015, doi: 10.1016/j.eij.2015.06.005.

[22] L. Sharma, A. Gera, "A survey of recommendation system: Research challenges," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 4, no. 5, pp. 1989–1992, 2013.

[23] J. B. Schafer, D. Frankowski, J. Herlocker, S. Sen, "Collaborative filtering recommender systems," in *The adaptive web*, Springer, 2007, pp. 291–324.

[24] J. Bobadilla, A. Hernando, F. Ortega, J. Bernal, "A framework for collaborative filtering recommender systems," *Expert Systems with Applications*, vol. 38, no. 12, pp. 14609–14623, 2011, doi: https://doi.org/10.1016/j.eswa.2011.05.021.

[25] B. M. Sarwar, G. Karypis, J. A. Konstan, J. Riedl, *et al.*, "Item-based collaborative filtering recommendation algorithms.," *WWW*, vol. 1, pp. 285–295, 2001.

[26] J. Bobadilla, F. Ortega, A. Hernando, J. Alcalá, "Improving collaborative filtering recommender system results and performance using genetic algorithms," *Knowledge-based systems*, vol. 24, no. 8, pp. 1310–1316, 2011.

[27] J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.

[28] M. De Gemmis, P. Lops, C. Musto, F. Narducci, G. Semeraro, "Semantics-aware content-based recommender systems," in *Recommender Systems Handbook*, Springer, 2015, pp. 119–159.

[29] M. De Gemmis, P. Lops, G. Semeraro, P. Basile, "Integrating tags in a semantic content-based recommender," in *Proceedings of the 2008 ACM conference on Recommender systems*, 2008, pp. 163–170, ACM.

[30] C. Musto, G. Semeraro, M. de Gemmis, P. Lops, "Learning word embeddings from wikipedia for content-based recommender systems," in *European Conference on Information Retrieval*, 2016, pp. 729–734, Springer.

[31] J. Bobadilla, F. Ortega, A. Gutiérrez, S. Alonso, "Classification-based deep neural network architecture for collaborative filtering recommender systems.," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 6, no. 1, 2020, doi: 10.9781/ijimai.2020.02.006.

[32] H. Cordobés de la Calle, L. F. Chiroque, A. Fernández Anta, R. García, P. Morere, L. Ornella, F. Pérez, A. Santos, "Empirical comparison of graph-based recommendation engines for an apps ecosystem," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 3, no. 2, pp. 33–39, 2015.

[33] K. W. Darlington, "Designing for explanation in health care applications of expert systems," *Sage Open*, vol. 1, no. 1, p. 2158244011408618, 2011.

[34] B. Walek, V. Fojtik, "A hybrid recommender system for recommending relevant movies using an expert system," *Expert Systems with Applications*, vol. 158, p. 113452, 2020.

[35] N. Tintarev, "Explanations of recommendations," in *Proceedings of the 2007 ACM conference on Recommender systems*, 2007, pp. 203–206, ACM.

[36] Y. Zhang, X. Chen, "Explainable recommendation: A survey and new perspectives," *Foundations and Trends in Information Retrieval*, vol. 14, no. 1, pp. 1–101, 2020, doi: 10.1561/1500000066.

[37] I. Nunes, D. Jannach, "A systematic review and taxonomy of explanations in decision support and recommender systems," *User Modeling and User-Adapted Interaction*, vol. 27, no. 3-5, pp. 393–444, 2017, doi: 10.1007/s11257-017-9195-0.

[38] M. Caro-Martinez, G. Jimenez-Diaz, J. A. Recio-Garcia, "A theoretical model of explanations in recommender systems," *ICCBR 2018*, p. 52, 2018.

[39] A. Papadimitriou, P. Symeonidis, Y. Manolopoulos, "A generalized taxonomy of explanations styles for traditional and social recommender systems," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 555–583, 2012, doi: 10.1007/s10618-011-0215-0.

[40] F. Gedikli, D. Jannach, M. Ge, "How should I explain? a comparison of different explanation types for recommender systems," *International Journal of Human-Computer Studies*, vol. 72, no. 4, pp. 367–382, 2014.

[41] P. Kouki, J. Schaffer, J. Pujara, J. O'Donovan, L. Getoor, "Personalized explanations for hybrid recommender systems," in *Proceedings of the 24th International Conference on Intelligent User Interfaces*, 2019, pp. 379– 390, ACM.

[42] L. Quijano-Sanchez, C. Sauer, J. A. Recio-Garcia, B. Diaz-Agudo, "Make it personal: a social explanation system applied to group recommendations," *Expert Systems with Applications*, vol. 76, pp. 36–48, 2017.

[43] I. Andjelkovic, D. Parra, J. O'Donovan, "Moodplay: Interactive music recommendation based on artists' mood similarity," *International Journal of Human- Computer Studies*, vol. 121, pp. 142–159, 2019.

[44] X. Wang, X. He, F. Feng, L. Nie, T.-S. Chua, "Tem: Tree-enhanced embedding model for explainable recommendation," in *Proceedings of the 2018 World Wide Web Conference*, WWW '18, Republic and Canton of Geneva, Switzerland, 2018, pp. 1543–1552, International World Wide Web Conferences Steering Committee.

[45] J. Jorro-Aragoneses, M. Caro-Martinez, J. A. Recio-Garcia, B. Diaz-Agudo, G. Jimenez-Diaz, "Personalized case-based explanation of matrix factorization recommendations," in *International Conference on Case-Based Reasoning*, 2019, pp. 140–154, Springer.

[46] F. Sørmo, J. Cassens, A. Aamodt, "Explanation in case- based reasoning–perspectives and goals," *Artificial Intelligence Review*, vol. 24, no. 2, pp. 109–143, 2005, doi: 10.1007/s10462-005-4607-7.

[47] J. Bennett, S. Lanning, *et al.*, "The netflix prize," in *Proceedings of KDD cup and workshop*, vol. 2007, 2007, p. 35, New York, NY, USA.

[48] M. Nickel, K. Murphy, V. Tresp, E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 11– 33, 2015, doi: 10.1109/JPROC.2015.2483592.

[49] L. Qiao, L. Zhang, S. Chen, D. Shen, "Data-driven graph construction and graph learning: A review," *Neurocomputing*, vol. 312, pp. 336–351, 2018.

[50] Q. Wang, Z. Mao, B. Wang, L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724–2743, 2017.

[51] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," *Semantic web*, vol. 8, no. 3, pp. 489–508, 2017.

[52] Y. Lin, X. Han, R. Xie, Z. Liu, M. Sun, "Knowledge representation learning: A quantitative review," 2018, https://arxiv.org/abs/1812.10901.

[53] Y. Chong, Y. Ding, Q. Yan, S. Pan, "Graph-based semi- supervised learning: A review," *Neurocomputing*, 2020.

[54] B. Furht, *Handbook of social network technologies and applications*. Springer Science & Business Media, 2010.

[55] L. Lü, T. Zhou, "Link prediction in complex networks: A survey," *Physica A: statistical mechanics and its applications*, vol. 390, no. 6, pp. 1150–1170, 2011, doi: 10.1016/j.physa.2010.11.027.

[56] N. N. Daud, S. H. Ab Hamid, M. Saadoon, F. Sahran, N. B. Anuar, "Applications of link prediction in social networks: A review," *Journal of Network and Computer Applications*, p. 102716, 2020.

[57] N. Chiluka, N. Andrade, J. Pouwelse, "A link prediction approach to recommendations in large-scale user- generated content systems," in *European Conference on Information Retrieval*, 2011, pp. 189–200, Springer.

[58] Z. Huang, W. Chung, T.-H. Ong, H. Chen, "A graph- based recommender system for digital library," in *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, 2002, pp. 65–73, ACM.

[59] H. Chen, X. Li, Z. Huang, "Link prediction approach to collaborative filtering," in *Digital Libraries, 2005. JCDL'05. Proceedings of the 5th ACM/IEEE-CS Joint Conference on*, 2005, pp. 141–142, IEEE.

[60] H. Wang, F. Zhang, X. Xie, M. Guo, "Dkn: Deep knowledge-aware network for news recommendation," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1835–1844.

[61] A. Shahmohammadi, E. Khadangi, A. Bagheri, "Presenting new collaborative link prediction methods for activity recommendation in facebook," *Neurocomputing*, vol. 210, pp. 217–226, 2016.

[62] T. Zhou, J. Ren, M. Medo, Y.-C. Zhang, "Bipartite network projection and personal recommendation," *Physical Review E*, vol. 76, no. 4, p. 046115, 2007, doi: 10.1103/PhysRevE.76.046115.

[63] I. Tiddi, *et al.*, "Foundations of explainable knowledge- enabled systems," *Knowledge Graphs for eXplainable Artificial Intelligence: Foundations, Applications and Challenges*, vol. 47, p. 23, 2020.

[64] F. Lecue, "On the role of knowledge graphs in explainable ai," *Semantic Web*, no. Preprint, pp. 1–11, 2019.

[65] N. Barbieri, F. Bonchi, G. Manco, "Who to follow and why: link prediction with explanations," in *20th ACM SIGKDD International Conference on Knowledge discovery and data mining*, 2014, pp. 1266–1275, ACM.

[66] Y. Xian, Z. Fu, S. Muthukrishnan, G. De Melo, Y. Zhang, "Reinforcement knowledge graph reasoning for explainable recommendation," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 285–294.

[67] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, T.-S. Chua, "Explainable reasoning over knowledge graphs for recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5329–5336.

[68] M. Caro-Martinez, G. Jimenez-Diaz, "Similar users or similar items? comparing similarity-based approaches for recommender systems in online judges," in *International Conference on Case-Based Reasoning*, 2017, pp. 92–107, Springer.

[69] G. Jimenez-Diaz, P. P. Gómez-Martín, M. A. Gómez- Martín, A. A. Sánchez-Ruiz, "Similarity metrics from social network analysis for content recommender systems," *AI Communications*, vol. 30, no. 3-4, pp. 223– 234, 2017.

[70] G. Jimenez-Diaz, P. P. G. Martín, M. A. G. Martín, A. A. Sánchez-Ruiz, "Similarity metrics from social network analysis for content recommender systems," in *International Conference on Case-Based Reasoning*, 2016, pp. 203–217, Springer.

[71] S. Dooms, A. Bellogín, T. D. Pessemier, L. Martens, "A framework for dataset benchmarking and its application to a new movie rating dataset," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 7, no. 3, p. 41, 2016.

Marta Caro-Martínez

Marta Caro-Martínez is a Ph.D. Student at the Complutense University of Madrid. She got a degree in Computer Science in 2015 and a Master's degree in the same field in 2017. Her research work focuses on recommender systems and Explainable Artificial Intelligence. She uses graphs and Social Network Analysis techniques to implement recommendations and explanations. She has also researched interactive visualization and augmented reality applications for museums. Her publications have appeared at several International Conferences.

Guillermo Jiménez Díaz

Guillermo Jiménez Díaz is Computer Research Scientist and Associate Professor at Universidad Complutense Madrid. He received his Ph.D. Universidad Complutense Madrid in Computer Science. His Thesis focused on using virtual environments and active-based learning to teach Object-Oriented Programming. His research is concerned to recommender systems and its combination with techniques from social network analysis. Our research is applied in two different domains: tourism and e-learning. He is also interested in the application of augmented reality technologies in Museums.

Juan A. Recio García

Juan A. Recio García is Associate Professor at the Department of Software Engineering and Artificial Intelligence at the Computer Science Faculty at the Complutense University of Madrid, where he held the position of Head of Department from 2015 to 2019. Currently he holds the BOSCH-UCM Honorary Chair on Artificial Intelligence applied to Internet of Things and is Board Member of the IAAA Student Chapter in Mexico for the promotion of Artificial Intelligence. He is lead investigator of several national-founded projects and he has conducted several contracts with companies in the area of Artificial Intelligence. His research has focused on the confluence of Software Engineering and Case-Based Reasoning, developing the COLIBRI platform for building CBR systems. He has also worked in the areas of Context-aware and social Recommender Systems. Currently his research is focused on eXplainable Artificial Intelligence (XAI).