On How Networks Stabilize User Interest Based Methods and Vice Versa

László Grad-Gyenge

University Eötvös Loránd
Budapest, Hungary
laszlo.grad-gyenge@inf.elte.hu

Peter Filzmoser

TU Vienna
Vienna, Austria
peter.filzmoser@tuwien.ac.at

***Abstract*—This paper presents the evaluation of two graph-based recommendation methods compared to collaborative filtering as the baseline. The evaluation is primarily based on the investigation of the Average Receiver operating Characteristic curve on the MovieLens dataset. The presented methods operate on the knowledge graph, which information representation technique is also discussed in this paper. The evaluation results show that combining a network based and a user interest based method leads to a more stable performance and an increase in the recommendation quality.**

***Keywords*–recommender system; graph based; knowledge graph; explicit feedback; receiver operating characteristic**

I. Introduction

Collaborative filtering and content-based filtering are two prominent classes of the recommender systems. The essence of collaborative filtering is that the recommendations are derived only from user-item interactions. Content-based techniques primarily focus on item attributes. Thinking in general, utilizing the user attributes can also be treated as a content-based technique. Our work is based on a graph based information representation technique, which is capable to represent user-item interactions, item attributes and user attributes at the same abstraction level. We call this representation technique the knowledge graph. This technique can be treated as a hybridization method at the information representation level.

Graph based recommender systems provide an alternative aspect to the widely used, matrix or tensor oriented methods. An advantage of the graph based representation is the potential to develop recommendation methods operating on networks. Referring to the results of network science, utilizing networks, the calculation methods can be improved regarding to their robustness and stability. In addition, the graph based representation has the capability to represent heterogeneous information sources and to provide a general information representation method. Working with heterogeneous information can be helpful to eliminate the cold start problem, as the more information is available, the higher is the chance to connect the current user to the items in the graph.

In our work, we focus on separating the information representation method from the calculation methods. We do this in order to have a clearer methodological approach. As the representation method provides the hybridization technique, two calculation methods, spreading activation and recommendation spreading is described and compared to the performance of collaborative filtering. Recommendation spreading basically alloys spreading activation and collaborative filtering. As recommendation spreading does not use any representation technique to compress the adjacency matrix of the graph, we compare the performance of recommendation spreading to the performance of collaborative filtering.

Grad-Gyenge et al. [1] evaluate recommendation spreading and collaborative filtering regarding to the mean absolute error (MAE) and the coverage of the mentioned methods. As list based recommendations are more in the focus of interest of the research on recommendation techniques, a Receiver operating Characteristic (RoC) based evaluation of the methods is presented. In order to adapt the RoC measure to the field of recommender systems, the Average Receiver operating Characteristic (ARoC) evaluation method is introduced. Providing an overview of the performance of the evaluated method, ARoC interprets the RoC in the case of recommender systems, as the RoC graphs are averaged over all users in the knowledge graph. The result is a more robust measure and a smoother graph. An additional advantage of the evaluation method is that it also provides information about the completeness of the list of retrieved items.

The contribution of the paper can be summarized as follows. The evaluation results show that regarding to the ARoC, recommendation spreading is capable to incorporate the advantages of spreading activation and collaborative filtering, thus we show that the information found in the network can stabilize rating value based methods and also vice versa. The information found in rating values can improve the network based calculation. We also show that in the information sparse case, the rating estimation based methods show better performance than ranking based methods.

Section II presents related research conducted. Section III discusses the graph based information representation technique. Section IV describes the recommendation methods evaluated in the paper. Section V introduces ARoC, the evaluation method. Section VI presents the results of the evaluation. Section VII concludes the paper and gives insight into our plans for the future.

II. Related Work

To discuss related research conducted, we focus on graph based information representation techniques, spreading activation based methods and RoC evaluation methods. Regarding to graph based information representation and spreading activation based techniques, the improvement presented in this paper can be found in the performance of recommendation spreading.

Although not widely used, the graph based information representation technique presents in the field of recommender

 ISBN: 978-1-61208-542-5

systems. State of the art research is also conducted on graph based representation. Tiroshi et al. [2] involve graph representation to work with social data. Lee et al. [3] represents correlations between the entities in a graph. Similarly to the representation presented in this paper, Lee at al. [4] represents content-based and collaborative filtering information in a heterogeneous graph.

Next to ontology representation, graphs are typically involved to model the social relationships. To mention asymmetric networks, Ziegler et al. [5], Guha et al. [6], Jsang et al. [7], Massa et al. [8] calculate the recommendations with the help of the trust network. Symmetric networks are also involved, as He et al. [9], Konstas et al. [10] and Guy et al. [11] calculate recommendations with the help of the social network. Layered graphs, as less generalized approaches also can be found in the literature. Cantador et al. [12] apply a clustering technique on a multi-layered graph. Kazienko et al. [13] calculate recommendations on a layered graph.

Representing heterogeneous information in the knowledge graph is also in the focus of intense research. Burke et al. [14] define a heterogeneous network in order to be able to model various recommendation cases as user-based k-Nearest Neighbors algorithm (k-NN) with the user-tag matrix, user-based k-NN with the user-resource matrix, item-based k-NN with the resource-tag matrix and item-based k-NN with the resource-user matrix. Yu et al. [15] introduce the PathSim measure to compare paths in the knowledge graph to measure the similarity between the observed and the potential paths. Catherine et al. [16] derive recommendations with a probabilistic logic approach on the knowledge graph. Hu et al. [17] present label propagation for lead generation. Kouki et al. [18] define a probabilistic framework as a hybridization technique.

Spreading activation is widely used in different domains to derive recommendations. Alvarez et al. define ONTO-SPREAD, a well-elaborated, spreading activation based method for medical systems [19]. Troussov et al. present the investigation of different decay configurations of spreading activation in a tag aware recommendation scenario [20]. Gao et al. argue that the domain knowledge and user interests on items are to be represented in the same ontology [21]. Blanco-Fernandez et al. utilize spreading activation to conduct content based reasoning [22]. In their work, they model the semantics of the preferences of the users. They stress out that spreading activation is a potential method to avoid overspecialisation. Jiang et al. utilize spreading activation to calculate recommendations on an ontology based user model [23]. The primary goal of Hussein et al. is to close the gap between context-awareness and self-adaptation [24]. To perform this task, SPREADR, a spreading activation based recommendation method is defined. Codina et al. present a semantic recommendation method to estimate user ratings on items with a reasoning technique [25]. In their work, the item score is defined as the weighted average of the related concepts.

Herlocker et al. describe a method to prepare the RoC curve [26], which is a known evaluation technique in the field of recommender systems. They leave the definition of the relevance of an item for the specific user open, thus the relevance is to be defined for the actual evaluation case. For example, in the case of rating estimation methods, the relevance can be defined based on a threshold value. To draw an RoC graph, the curve is started from the origin and an iteration is conducted on the recommendation result list. For each item, the relevance is determined. If the item is relevant, then the curve is drawn one step vertically. If the item is not relevant, then the curve is drawn one step horizontally. Herlocker et al. define the RoC curve for the specified user. As typically there are several users utilizing a recommender system, a possible enhancement of the RoC should examine the performance of the recommendation method regarding to all users or a well defined subset of users.

Cremonesi et al. also utilize the RoC curve to evaluate their recommendation methods [27]. In their work, Cremonesi et al. define two variants of the RoC curve and denote them as ROC1 and ROC2. ROC1 uses a threshold based technique to identify true positives, false positives, true negatives and false negatives. ROC2 is suitable for ranked lists and is defined for both the binary and the non-binary case. An important aspect of their work is that Cremonesi et al. use a user sampling technique. To focus on users with relatively sparse on item preferences, they evaluate their methods on the subset of users containing users issued at most 99 ratings.

Improvements to the RoC curve can also be found in the literature. Schein et al. introduce CROC, the Customer RoC curve [28]. In their work, Schein et al. stress out the divergence in the lengths of the recommendation lists of different users. To solve this problem, they propose a technique to unify the lengths of the recommendation lists and calculate the measures necessary to produce the RoC curve based on the unified lists. Also mentioning the problem of different lengths of item lists, Schröder et al. focuses on the first n items of the recommendation lists [29].

III. Representation Technique

The advantage of the graph based knowledge base is the capability to represent heterogeneous information sources in the same structure. In this section, a modelling technique is discussed, which is capable to store the information necessary for both collaborative and content-based filtering. In special cases, this technique can also act as the background of rule based systems. A similar representation technique is used by Lee et al. [4], Burke et al. [14], Yu et al. [15], Kouki et al. [18] and Grad-Gyenge et al. [1].

This section presents the definition of the information representation method and also provides theoretical insights. In order to clarify the approach, the concrete dataset and its representation is described in this section.

A. *Definition*

The information is represented in a labelled multigraph. We refer to it as the knowledge graph or the knowledge base and define it in Equation (1).

$$\mathcal{K} = (N, E, T_N, T_E, t_N, t_E, r), \tag{1}$$

where N represents the set of nodes in the graph, $E \subseteq \{\{u, v\} | u \in N \wedge v \in N \wedge u \neq v\}$ represents the set of undirected edges between the nodes. T_N denotes the set of node types and T_E denotes the set of edge types. The function $t_N \subset N \times T_N$ assigns a node type to each node, the function $t_E \subset E \times T_E$ assigns an edge type to each edge. The partial function $r \subset E \times \mathbb{R}$ assigns a rating value to specific edges. The function is partial, as in most cases not all the edges

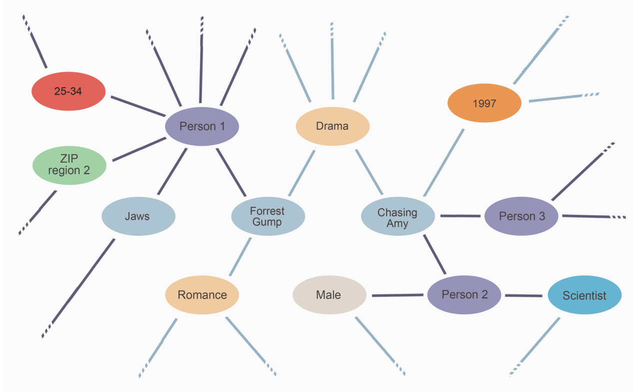


Figure 1. A detailed view of the MovieLens dataset represented in the knowledge graph.

represent a rating. Although it is formally not defined, in the implementation, due to performance reasons, we avoid parallel edges of the same type between the same pair of nodes. We would also like to mention here that type assignments do not influence the final recommendation result and are introduced for completeness.

B. MovieLens

The numerical experiment is conducted on the MovieLens dataset [30]. Analysing the available MovieLens versions, we decided to use the MovieLens 1M dataset, as in addition to containing true rating values, this dataset is also rich in user and item attributes. The user attributes are gender, age category, occupation and ZIP code. The item attributes are year of release and list of genres.

As illustrated in Fig. 1, the user and item attributes are modelled in the knowledge graph similarly how semantic networks represent the information. Light blue nodes as `Jaws`, `Forrest Gump` and `Chasing Amy` represent the items (movies). Lilac nodes as `Person 1`, `Person2` and `Person3` represent the persons. Drab nodes as `Romance` and `Drama` represent the genres. The gray node represents the gender `Male`. The blue node represents the occupation `Scientist`. The light brown node represents the release year `1997`. The green node represents the `ZIP region 2`. The red node represents the age category `25-34`.

Each user is represented with a node of type `Person`. A node type is introduced to represent each kind of user attributes. To represent the user attributes, a node of the appropriate type is created for each attribute value. Nodes of type `Gender` represent the genders. Nodes of type `AgeCategory` represent the age categories. Nodes of type `Occupation` represent the occupations. Nodes of type `ZipCodeRegion` represent the zip code regions. In this case the first digit of the zip code is used as it determines the U.S. region. To model that a user has a specific attribute value, the node representing the user is connected to the node representing the attribute value with an edge of the appropriate type. To model this information, edge types `PersonGender`, `PersonAgeCategory`, `PersonOccupation` and `PersonZipCodeRegion` are introduced, respectively.

Each item is represented with a node of type `Item`. To represent the item attributes, a node of the appropriate type is created for each attribute value. Nodes of type `Genre` represent the genres. An item can have multiple genres. In this case, the item node is connected to multiple genre nodes. Nodes of type `YearOfRelease` represent the years of release. To model that an item has a specific attribute value, the node representing the item is connected to the node representing the attribute value with an edge of the appropriate type. To model this information, edge types `PersonGender`, `PersonAgeCategory`, `PersonOccupation` and `PersonZipCodeRegion` are introduced, respectively.

The MovieLens 1M dataset contains 1 000 209 true ratings. Each true rating consists of an item, an user, a rating value and a time-stamp the rating event has been recorded at. The rating values are integer numbers and are in the interval $[1, 5]$. In our experiment, the rating values are normalized and are transformed linearly into the interval $[0.2, 1]$ by a division by 5. We denote the set of known true rating events with T and an element of the set with t. To access the attributes of true rating t, $t.u$, $t.i$, $t.v$ and $t.t$ stands for the user, item, value and time-stamp of rating t, respectively.

In the case a rating is added to the knowledge base, a new edge of type `ItemRating` is created between the node representing the user and the node representing the rated item. The rating value is assigned to the edge using the function r.

C. The Limes of the Hybridization

To present the amount of information the methods operate on in this experiment, Table I summarizes the count of nodes and edges in the knowledge graph. Subtable Ia contains the number of nodes of each node type. Subtable Ib presents the number of edges of each edge type. The total number of nodes is 10 062. The total number of edges not counting edges of type `ItemRating` is 34 451.

TABLE I. COUNT OF NODE AND EDGE TYPES IN THE MOVIELENS DATASET.

(a) Count of node types.

Node type	Count
Person	6 040
AgeCategory	7
Gender	2
Occupation	21
ZipCodeRegion	10
Item	3 883
Genre	18
YearOfRelease	81

(b) Count of edge types.

Edge type	Count
PersonAgeCategory	6 040
PersonGender	6 040
PersonOccupation	6 040
PersonZipCodeRegion	6 040
ItemGenre	6 408
ItemYearOfRelease	3 883
ItemRating	1 000 209

The representation technique models the information necessary to conduct both collaborative and content-based filtering methods. A properly defined calculation method should treat these information sources as general. It means that deriving recommendations, the calculation method should process the edges of different type at the same algorithmic abstraction level.

In the cold start case, when the knowledge base is sparse on `ItemRating` edges and is relatively dense on edges representing content-based information, the recommendation method can be treated as content-based. Thinking about the magnitude of the number of edges of type `ItemRating` (1 000 209) and other, content-based edges (34 451), as during the operation, the knowledge base is filled with user interaction, the recommendations are to be become more collaborative. In other words, the hybridization technique inherently ensures content-based recommendations in the cold start case and inherently transforms the methods operating on the top of it to be collaborative as it is populated with edges representing user-item interaction.

IV. Recommendation Methods

In our experiment, collaborative filtering, spreading activation, recommendation spreading and random recommendations are evaluated. The methods are defined in the following subsections.

A. Collaborative Filtering

We utilize a representation technique, which gives a different aspect to the more or less traditional, matrix based methods. Another problem of the matrix based representation is the restricted ability to represent heterogeneous information sources. A well researched direction to solve this issue is to involve tensors and to conduct tensor factorization [31].

Collaborative filtering [30] calculates rating estimations basically by averaging the known ratings on the item in question. The weight of a rating is the similarity of the user issuing the rating to the user the recommendations are generated for. To be more exact, instead of aggregating the known ratings, the differences from the mean ratings are averaged and then addded to the mean rating of the user. The rating estimation formula for user u on item i on our knowledge base is defined in Equation (2).

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{e \in E_r, \{v,i\}=e, v \neq i, u \neq v} (r(e) - \bar{r_v}) s_{u,v}}{\sum_{e \in E_r, \{v,i\}=e, v \neq i, u \neq v} s_{u,v}}, \qquad (2)$$

where $\hat{r}_{u,i}$ denotes the estimated rating value for user u on item i. The average of the known ratings is denoted by $\bar{r}_u$. To calculate the similarity between u and v, the Pearson correlation formula is utilized on the rating values of the common rated items.

B. Spreading Activation

Spreading activation [32] is widely used on ontologies, semantic networks, associative networks and RDF knowledge bases [33]. To utilize the method on the knowledge graph, spreading activation is used to calculate ranking values for the items. In this case no rating estimation is calculated.

Spreading activation is an iterative method with a `step limit` (c) based termination criteria. To generate item rankings, the method maintains the activation of the nodes. The function $a_{(i)} \subset N \times \mathbf{R}$ assigns an activation value to each node in the graph in each iteration step. In the initial step the activation of the nodes is set to 0 except for n_s, as $a_{(0)}(n_s) = 1$. The notation n_s stands for the source node, the node representing the user.

During the iteration, the activation of the nodes is propagated in the network. In each step, (i) a part of the activation of the nodes is distributed to the neighbour nodes equally and (ii) the activation of the nodes is relaxed. The parameter `spreading relax` (r_s) controls the amount of activation to distribute. The parameter `activation relax` (r_a) determines the ratio the activation of the nodes are to be relaxed. The update of the activation is conducted according to the rule defined in Equation (3).

$$a_{(i+1)}(n) = r_a a_{(i)}(n) + r_s \sum_{m \in M_n} \frac{a_{(i)}(m)}{|M_n|}, \qquad (3)$$

where $n \in N$, $i \geq 0$. The set containing the neighbour nodes of n is denoted with M_n, as $M_n = \{m | \{m,n\} \in E\}$.

The iteration is performed until the `step limit` (c) is reached. The rank of each node is defined as its activation after the iteration has been stopped.

C. Recommendation Spreading

Recommendation spreading introduced by Grad-Gyenge et al. can be treated as the generalization of collaborative filtering for the graph based knowledge base [1]. The method is based on spreading activation but focuses on rating estimation. As already discussed, collaborative filtering defines a weighted average of the known rating values. In the case of collaborative filtering, the weights are determined by the similarity of the users. In the case of recommendation spreading, a distance like measure is defined between the user to generate the recommendations for and the edges representing the known rating values. To calculate the distance, an iteration is conducted with a `step limit` (c) based termination criteria. The activations are calculated using the same formula as in the case of the spreading activation. In each iteration step, the amount of flown through activation is summarized for the edges, as defined in Equation (4).

$$A_e = \sum_{i \in [0,s-1], m \in e, t_N(m) = Person} r_s \frac{a_{(i)}(m)}{|M_n|}, \qquad (4)$$

where $e \in E$. The set containing the neighbour nodes of n is denoted with M_n, as $M_n = \{m | \{m,n\} \in E\}$.

To estimate the rating of an item, recommendation spreading calculates a weighted average. The weight of a rating is the flown through activation in the spreading iteration, as defined in Equation (5).

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{e \in E_r, \{v,i\}=e, v \neq i, u \neq v} (r(e) - \bar{r_v}) A_e}{\sum_{e \in E_r, \{v,i\}=e, v \neq i, u \neq v} A_e}. \qquad (5)$$

D. The Random Method

The random method is involved in the experiment for the following reasons. Thinking about the no free lunch theorem [34], the method can act as the theoretical baseline for all the methods. The other reason to involve random recommendations into the experiment is to control the mathematical correctness of our evaluation measure. A more or less trivial consequence of the definition of the RoC curve is that the performance of random recommendations should show a minor diagonal on the RoC graph.

 ISBN: 978-1-61208-542-5

E. List Recommendations

Collaborative filtering and recommendation spreading calculates a rating estimation for each item. Based on the rating estimation of the items, the list of recommended items is assembled by sorting the items in descending order by their rating estimation. As spreading activation calculates a ranking value for each item, the list of recommended items in this case is assembled by sorting the items in descending order by their rank.

We introduce function m as the notation for calculating the list of the recommended items for user u using method m over knowledge base $\mathcal{K}$, as defined in Equation (6).

$$m_{\mathcal{K}}(u). \tag{6}$$

V. Evaluation Method

The evaluation of the recommendation methods is conducted with ARoC, a RoC based evaluation technique, which is to be defined in this section. An important parameter of the evaluation method is the amount of rating edges to be inserted in the knowledge base in addition to the content-based information.

A. Initial Information

The evaluation of the recommendation techniques analysed in this paper is strongly connected to the information content contained in the knowledge base. Each evaluation starts with a knowledge graph containing only the user and the item attribute edges. In this case, there is no user preference stored in the knowledge graph. To incorporate also user preference information, a specified number of ratings is added to the knowledge base by creating edges of type `ItemRating`. As our intention is to model real-world applications, in this case, the first n ratings are selected from the true ratings in ascending order by their time-stamp. The first n ratings are called training set and are denoted with T_i.

B. Definitions

The evaluation of the methods is based on the RoC curve. Its essence is to plot the true positive rate (TPR) of the method in question against its false positive rate (FPR) on a single plot. RoC is typically used in the case of binary classification. In our experiment an item is defined to be positive for a particular user if the training data contains a true rating value higher than a pre-defined threshold. An item is defined to be negative if the value of the true rating is lower than the threshold. To formalize it, the predicate p stands for the positive case and the predicate n stands for the negative case as defined in Equations (7) and (8), respectively.

$$p_\Theta(u,i) = \exists t \in T : u = t.u \wedge i = t.i \wedge t.v \geq \Theta, \tag{7}$$

$$n_\Theta(u,i) = \exists t \in T : u = t.u \wedge i = t.i \wedge t.v < \Theta. \tag{8}$$

Based on the predicates, the true positive and related measures are to be defined. The functions TP, FP, TN, FN calculate the number of true positive, false positive, true negative and false negative items, respectively. The functions are defined in Equations (9), (10), (11) and (12), recpectively.

$$TP_{\Theta,k}(u,l) = |\{i \in I \mid p_\Theta(u,i) \wedge \exists j \leq k : i = l_j\}|, \tag{9}$$

$$FP_{\Theta,k}(u,l) = |\{i \in I \mid n_\Theta(u,i) \wedge \exists j \leq k : i = l_j\}|, \tag{10}$$

$$TN_{\Theta,k}(u,l) = |\{i \in I \mid n_\Theta(u,i) \wedge \nexists j \leq k : i = l_j\}|, \tag{11}$$

$$FN_{\Theta,k}(u,l) = |\{i \in I \mid p_\Theta(u,i) \wedge \nexists j \leq k : i = l_j\}|. \tag{12}$$

Functions TP, FP, TN and FN count the items for user u on the list of items l. The function attribute Θ specifies the threshold value. The function attribute k specifies the length the item list should be analized for.

The RoC curve is produced by plotting the TPR against the FPR in a graph. The TPR is the ratio of positive items retrieved compared to all the positive items. The FPR is the ratio of negative items retrieved compared to all the negative items. The functions are defined in Equations (13) and (14), recpectively.

$$TPR_{\Theta,k}(u,l) = \frac{TP_{\Theta,k}(u,l)}{TP_{\Theta,k}(u,l) + FN_{\Theta,k}(u,l)}, \tag{13}$$

$$FPR_{\Theta,k}(u,l) = \frac{FP_{\Theta,k}(u,l)}{FP_{\Theta,k}(u,l) + TN_{\Theta,k}(u,l)}. \tag{14}$$

The function TPR and FPR deliver the appropriate ratio values for user u on the list of items l. The function attribute Θ specifies the threshold value. The function attribute k specifies the length the item list should be analized for.

The functions TPR and FPR are to be calculated for a given user and list of items. In order to be able to plot the RoC curve, a distinguished user has to be selected from the knowledge base. As this selection procedure is not a straightforward task, instead of calculating TPR and FPR for a specific user, the average of these measures is calculated for all the users in the dataset. For each user, the list of items is delivered by the evaluated recommendation method, thus l is to be substituted to $m_{\mathcal{K}}(u)$ as presented in Equations (15) and (16).

$$ATPR_{\Theta,k}(m) = \frac{\sum_{u \in U} TPR_{\Theta,k}(u, m_{\mathcal{K}}(u))}{|U|}, \tag{15}$$

$$AFPR_{\Theta,k}(m) = \frac{\sum_{u \in U} FPR_{\Theta,k}(u, m_{\mathcal{K}}(u))}{|U|}, \tag{16}$$

where m denotes the evaluated method. The method operates on the knowledge base $\mathcal{K}$. The set U denotes the set of users ($U = \{u \in N \mid t_N(u) = Person\}$).

 ISBN: 978-1-61208-542-5

C. ARoC

Having the underlying measures defined, an RoC based evaluation method is to be introduced, the Average Receiver operating Characteristic, the ARoC. The definition of the ARoC is based on $ATPR$ and $AFPR$. To draw the RoC curve of method m, k is iterated from zero to the length of the longest list of recommended items. For each value of k, a mark is plotted onto the graph. The coordinates of the mark are calculated as the value of $ATPR_{\Theta,k}(m)$ and $AFPR_{\Theta,k}(m)$.

As its name indicates, ARoC averages the RoC graphs over all the users into a single graph. Thanks to the aggregation, ARoC provides a more robust measure and also a smoother graph. The difference between RoC and ARoC can also be found in the drawing method. While the drawing of the RoC curve is based on vertical and horizontal steps of the same unit, the coordinates of the ARoC graph is defined by the $ATPR$ and $AFPR$ function. This is also the reason why the ARoC graph is not necessarily a continuous curve.

The $ATPR_{\Theta,k}$ and the $AFPR_{\Theta,k}$ measures are calculated as the averages on the lists of recommended items for the specified list length k. As mentioned in Section II, the recommendation lists typically differ in their length, as the reachable item nodes differ for each user. This is the reason why the higher is value of k, the lower is the amount of the averaged TPR and FPR measures.

Unlike random item selection, most recommendation methods do not retrieve the whole set of recommendable items. To illustrate this phenomenon in the graph oriented aspect, it is not ensured that all the items are linked to the users with the appropriate path. Looking at the ARoC graphs presented in Section VI, especially in the case of collaborative filtering and recommendation spreading, the graph of the methods do not reach the upper-right corner because of the aforementioned reason. We think about this property of the ARoC method as a useful feature, as next to illustrating the TPR and FPR of the methods, it also provides information about the completeness of the retrieved items.

VI. EVALUATION RESULTS

The methods described in Section IV are evaluated on the MovieLens dataset represented in the knowledge graph defined in Section III. The evaluation is based on the ARoC curve as defined in Section V.

To evaluate the methods, various the following parameter settings of Θ are evaluated 0.4, 0.6, 0.8 and 1.0. Due to space limitations, the 0.8 case is presented. This is also the most representative case. To interpret the 0.8 value, items with rating 4 or 5 are treated positive. Table II contains the presented method configurations. Column `Name` contains the short name of the method. Column `Method` holds the type of the method. Column `Configuration` defines the configuration parameters of the methods if there is any. Our past results [1] show that the examined methods are not sensitive to the different r_a and r_s settings. These results are not presented in this paper due to space limitations. Regarding to the setting of c, those configurations are presented, which at most represent the evaluation properties of the specific method.

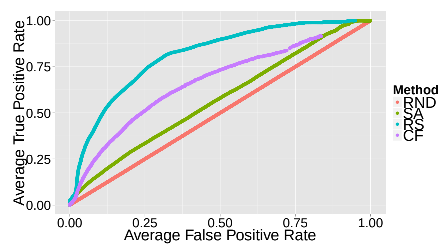
A. ARoC

As we are interested in how the amount of rating edges in the knowledge base influences the performance of the methods,

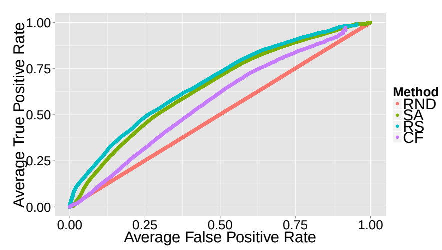
TABLE II. METHOD CONFIGURATIONS

Name	Method	Configuration
CF	Collaborative Filtering	–
SA	Spreading Activation	$c = 5$, $r_a = 0.5$, $r_s = 0.5$
RS	Recommendation Spreading	$c = 5$, $r_a = 0.5$, $r_s = 0.5$
RND	Random Method	–

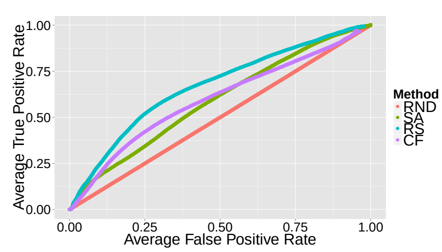
the evaluation is organized into 4 scenarios. The scenarios differ in the size of the training set (T_i). To refer to it later, the `10k`, `20k`, `40k` and `200k` shorthands are introduced for the case with 10 000, 20 000, 40 000 and 200 000 rating values, respectively. Fig. 2 contains the ARoC graph of the discussed methods with different $|T_i|$ settings in its subfigures.



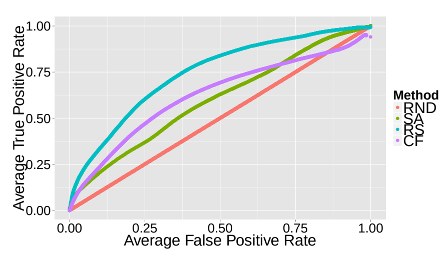
(a) 10 000 rating values.



(b) 20 000 rating values.



(c) 40 000 rating values.



(d) 200 000 rating values.

Figure 2. The ARoC curve of the evaluated methods on the knowledge containing different number of rating values.

 ISBN: 978-1-61208-542-5

The primary result of the evaluation is that RS outperforms the other methods in all the 4 scenarios. The advantage of the method stands out more in the information sparse (`10k`) and in the information dense (`200k`) cases.

Comparing the CF and the SA, the performane of the methods vary with the different amount of rating values in the knowledge graph. While in the `10k` scenario, the CF is dominant, in the `20k` case, the SA performs better. In the `40k` and the `200k` case, the CF performs better on the lower domain of k than SA then the graph of the CF and the SA are crossing each other. In these cases, the performance of the methods is ambigously comparable.

Analysing the performance of the CF on high k values, Fig. 2d shows that the performance of the CF falls below the performance of the RND. Referring to the no free lunch theorem [34], this is an important theoretical result. In addition, the curve of the CF is not monotonic and is also not continous. We explain it as follows. The ARoC measure is defined to show the average performance of the precision and recall measures over all users of the dataset. As the length of the recommendation lists grows, the amount of the averaged measures decreases causing non-monotonicity.

As mentioned in Section V-C, an important feature of the ARoC curve is that the coverage of the evaluated methods can be read from its graph. For example, the curve of CF on Fig. 2a does not reach the upper-right corner meaning that the CF retrieves only a subset of relevant items for the users. Analysing the methods from this aspect, it can be seen that the SA has the highest coverage, the RS is the second highest and the CF has the lowest performance. This result can be explained with the restrictions on the path between the users and the items. For example, in the case of the CF the path must contain exactly 3 edges and the type of the edges should be `ItemRating`. Regarding the current configuration, the length of the path of the RS is 5 and the type of the only last edge is restricted. The restriction for the SA is the path length of 5. A more trivial result is that the coverage of the methods grow as the more rating values are contained in the knowledge base.

A minor result of the evaluation is that the graph of the RND shows the minor diagonal.

B. *Time need*

Spreading activation based methods are computation intense. This is the reason why we also summarize the time need of the examined methods. Table III presents the time need of the methods. Column `Method` contains the method configuration. Columns `10000`, `20000`, `40000` and `200000` contains the time necessary to generate the recommendations to draw the ARoC curves in the `10k`, the `20k`, the `40k` and the `200k` case, respectively. The execution times are the total times of calculating 6 040 recommendations, as the ARoC curve averages the performance of the method among the users in the dataset.

TABLE III. THE TIME NEED OF THE EVALUATION OF THE METHODS IN THE INVOLVED SCENARIOS.

Method	10 000	20 000	40 000	200 000
CF	00:00:28	00:00:50	00:01:22	00:13:46
SA	00:29:07	00:52:38	00:34:20	00:58:05
RS	01:07:44	03:34:08	06:05:55	04:14:30

The numerical experiments have been conducted in a virtualised environment on a single computation core. The virtual hardware configuration is Intel(R) Xeon(R) CPU E5-2650 @ 2.00GHz, 11GB of memory. Regarding the computational resource need, the CF has the highest performance, SA is the next and RS involves the most resources.

VII. CONCLUSION

The performance of collaborative filtering, spreading activation and recommendation spreading is compared on the MovieLens dataset. The methods operate on the knowledge graph presented in Section III. The evaluation is based on ARoC, which evaluation method is introduced in this paper. Its essence is to average the RoC curves over all the users in the dataset. The evaluation results present the ARoC graphs of the methods in three different cases. The evaluation cases are distinguished by the amount of rating information inserted into the knowledge graph.

The SA calculates recommendations based on the structure of the knowledge graph. The CF derives its recommendations from user preferences on items. As its definition shows, recommendation spreading alloys spreading activation and collaborative filtering. On one hand, the RS can be treated as the generalization of the CF for the graph based case. On the other hand, the RS can be treated as the extension of the SA with the ability to incorporate rating values into the recommendations process. The method has the capability to both utilize the structure of the network to stabilize its performance and to involve the explicit ratings as a sophisticated declaration of the user affinity to the recommendable items. To draw a conclusion, the evaluation results show that regarding to the ARoC, while the CF and the SA show a varying performance, the RS successfully alloys the information found in the structure of the network and the information found in the user ratings. The price for the higher recommendation quality is the higher computational resource need.

Thinking about the cold start problem and the information sparse case, we would like to emphasize the evaluation case `1k`. This is the case with the lowest amount of information about user preferences on items. Also, this is the case, when the methods involving the user ratings as an information source provide a better performance than the ranking based spreading activation. To draw a conclusion based on the results, the rating values hold an important source of information, especially in the information sparse environment.

Analysing the ARoC curves over the evaluation cases, the graphs show that the performance of the methods decrease as the more training data is added to the knowledge base. This result lets us draw the same conclusion as described by Blanco-Fernandez et al. [22], as spreading activation based methods have the potential to avoid overspecialization.

In our future work, first of all, we would like to extend the evaluation scenarios to additional datasets. In addition, at the moment, no representation learning techniques are involved in the experiment. In order to further investigate the methods, our plan is to apply SVD or other matrix factorization technique to the adjacency matrix of the network and to involve additional, matrix factorization recommendation techniques to the evaluation.

REFERENCES

[1] L. Grad-Gyenge, P. Filzmoser, and H. Werthner, "Recommendations on a Knowledge Graph," in *MLRec 2015 : 1st International Workshop on Machine Learning Methods for Recommender Systems*, 2015, pp. 13–20.

[2] A. Tiroshi, S. Berkovsky, M. A. Kâafar, D. Vallet, and T. Kuflik, "Graph-Based Recommendations: Make the Most Out of Social Data." in *UMAP*, ser. Lecture Notes in Computer Science, V. Dimitrova, T. Kuflik, D. Chin, F. Ricci, P. Dolog, and G.-J. Houben, Eds. Springer, pp. 447–458.

[3] K. Lee and K. Lee, "Escaping your comfort zone: A graph-based recommender system for finding novel recommendations among relevant items." *Expert Syst. Appl.*, no. 10, pp. 4851–4858.

[4] S. Lee, S. Park, M. Kahng, and S. goo Lee, "PathRank: Ranking nodes on a heterogeneous graph for flexible hybrid recommender systems." *Expert Syst. Appl.*, no. 2, pp. 684–697.

[5] C.-N. Ziegler and G. Lausen, "Propagation Models for Trust and Distrust in Social Networks," *Information Systems Frontiers*, vol. 7, no. 4-5, pp. 337–358, Dec. 2005.

[6] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust," in *WWW '04: Proceedings of the 13th international conference on World Wide Web*. New York, NY, USA: ACM, 2004, pp. 403–412.

[7] A. Jøsang, S. Marsh, and S. Pope, "Exploring Different Types of Trust Propagation." in *iTrust*, ser. Lecture Notes in Computer Science, K. Stølen, W. H. Winsborough, F. Martinelli, and F. Massacci, Eds., vol. 3986. Springer, 2006, pp. 179–192.

[8] P. Massa and P. Avesani, "Trust-Aware Collaborative Filtering for Recommender Systems." in *CoopIS/DOA/ODBASE (1)*, ser. Lecture Notes in Computer Science, R. Meersman and Z. Tari, Eds., vol. 3290. Springer, 2004, pp. 492–508.

[9] J. He, "A Social Network-based Recommender System," Ph.D. dissertation, Los Angeles, CA, USA, 2010, aAI3437557.

[10] I. Konstas, V. Stathopoulos, and J. M. Jose, "On Social Networks and Collaborative Recommendation," in *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '09. New York, NY, USA: ACM, 2009, pp. 195–202.

[11] I. Guy *et al.*, "Personalized recommendation of social software items based on social relations." in *RecSys*, L. D. Bergman, A. Tuzhilin, R. D. Burke, A. Felfernig, and L. Schmidt-Thieme, Eds. ACM, pp. 53–60.

[12] I. Cantador and P. Castells, *Multilayered Semantic Social Network Modeling by Ontology-Based User Profiles Clustering: Application to Collaborative Filtering*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 334–349.

[13] P. Kazienko, K. Musial, and T. Kajdanowicz, "Multidimensional Social Network in the Social Recommender System," *Trans. Sys. Man Cyber. Part A*, vol. 41, no. 4, pp. 746–759, Jul. 2011.

[14] R. Burke, "The Adaptive Web," P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Heidelberg: Springer-Verlag, ch. Hybrid Web Recommender Systems, pp. 377–408.

[15] X. Yu *et al.*, "Personalized Entity Recommendation: A Heterogeneous Information Network Approach," in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, ser. WSDM '14. New York, NY, USA: ACM, pp. 283–292.

[16] R. Catherine and W. Cohen, "Personalized Recommendations Using Knowledge Graphs: A Probabilistic Logic Programming Approach," in *Proceedings of the 10th ACM Conference on Recommender Systems*, ser. RecSys '16. New York, NY, USA: ACM, pp. 325–332.

[17] Q. Hu *et al.*, "HeteroSales: Utilizing Heterogeneous Social Networks to Identify the Next Enterprise Customer," in *Proceedings of the 25th International Conference on World Wide Web*, ser. WWW '16. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, pp. 41–50.

[18] P. Kouki, S. Fakhraei, J. Foulds, M. Eirinaki, and L. Getoor, "HyPER: A Flexible and Extensible Probabilistic Framework for Hybrid Recommender Systems," in *Proceedings of the 9th ACM Conference on Recommender Systems*, ser. RecSys '15. New York, NY, USA: ACM, pp. 99–106.

[19] J. M. Alvarez, L. Polo, P. Abella, W. Jimenez, and J. E. Labra, "Application of the Spreading Activation Technique for Recommending Concepts of well-known ontologies in Medical Systems," 2011.

[20] A. Troussov, D. Parra, and P. Brusilovsky, "Spreading Activation Approach to Tag-aware Recommenders: Modeling Similarity on Multidimensional Networks," D. Jannach, W. Geyer, J. Freyne, S. S. Anand, C. Dugan, B. Mobasher, and A. Kobsa, Eds., 2009, pp. 57–62.

[21] Q. Gao, J. Yan, and M. Liu, "A Semantic Approach to Recommendation System Based on User Ontology and Spreading Activation Model." in *NPC Workshops*, J. Cao, M. Li, C. Weng, Y. Xiang, X. Wang, H. Tang, F. Hong, H. Liu, and Y. Wang, Eds. IEEE Computer Society, 2008, pp. 488–492.

[22] Y. Blanco-Fernández, M. L. Nores, A. Gil-Solla, M. R. Cabrer, and J. J. P. Arias, "Exploring synergies between content-based filtering and Spreading Activation techniques in knowledge-based recommender systems." *Inf. Sci.*, vol. 181, no. 21, pp. 4823–4846, 2011.

[23] X. Jiang and A.-H. Tan, "Learning and inferencing in user ontology for personalized Semantic Web search." *Inf. Sci.*, vol. 179, no. 16, pp. 2794–2808, 2009.

[24] T. Hussein, D. Westheide, and J. Ziegler, "Context-adaptation based on Ontologies and Spreading Activation," in *LWA 2007: Lernen - Wissen - Adaption, Halle, September 2007, Workshop Proceedings*, 2007, pp. 361–366.

[25] V. Codina and L. Ceccaroni, "Taking Advantage of Semantics in Recommendation Systems." in *CCIA*, ser. Frontiers in Artificial Intelligence and Applications, R. Alquézar, A. Moreno, and J. Aguilar-Martin, Eds., vol. 210. IOS Press, 2010, pp. 163–172.

[26] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," *ACM Trans. Inf. Syst.*, no. 1, pp. 5–53, Jan.

[27] P. Cremonesi, R. Turrin, E. Lentini, and M. Matteucci, "An Evaluation Methodology for Collaborative Recommender Systems," in *Automated solutions for Cross Media Content and Multi-channel Distribution, 2008. AXMEDIS '08. International Conference on*, Nov 2008, pp. 224–231.

[28] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "CROC: A New Evaluation Criterion for Recommender Systems," *Electronic Commerce Research*, no. 1, pp. 51–74.

[29] G. Schröder, M. Thiele, and W. Lehner, "Setting goals and choosing metrics for recommender system evaluations," in *In CEUR Workshop Proc.*, vol. 811, 2011, pp. 78–85.

[30] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," in *Proc. of ACM 1994 Conference on Computer Supported Cooperative Work*. Chapel Hill, North Carolina: ACM, 1994, pp. 175–186.

[31] B. Hidasi and D. Tikk, "Speeding up ALS learning via approximate methods for context-aware recommendations," *Knowledge and Information Systems*, no. 1, pp. 131–155.

[32] M. R. Quillian, "Semantic memory," in *Semantic Information Processing*, M. Minsky, Ed. Cambridge, MA: MIT Press, 1968, pp. 227–270.

[33] M. Hochmeister, "Spreading Expertise Scores in Overlay Learner Models." in *CSEDU (1)*, M. Helfert, M. J. Martins, and J. Cordeiro, Eds. SciTePress, pp. 175–180.

[34] D. H. Wolpert and W. G. Macready, "No Free Lunch Theorems for Optimization," *Trans. Evol. Comp*, no. 1, pp. 67–82, Apr.