Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference

PathMining: A Path-Based User Profiling Algorithm for Heterogeneous Graph-Based Recommender Systems

Sangkeun Lee
Computational Sciences and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831

Sanghyeb Lee
Department of EECS
University of Tennessee, Knoxville
Knoxville, TN 37996

Byoung-Hoon Park
Computational Sciences and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831

Abstract

In a heterogeneous graph-based recommender system, relationships among various entities are used to predict the rating or preference that a user would give to an item. By modeling user's selection behavior as a path in the heterogeneous graph, we can capture each user's unique selection behaviors as a set of paths with specific types of nodes and edges. Since these paths capture selection behaviors unique to each user, these can then be used to perform personalized profiling and recommendation for each user. In this paper, we introduce *PathMining*, an algorithm which constructs personalized user profile and make recommendations based on Monte Carlo sampling of graph traversal. *PathMining* predicts preferable items by emulating user's selection processes. The performance and potential value of our method is validated by using HetRec 2011 dataset.

Traditionally, recommendation has been considered as a problem of finding an utility function that best predicts a user's unknown ratings given the user's known ratings as found in User×Item rating matrix (Adomavicius and Tuzhilin 2005). In the past, most recommender systems, based on this problem formulation, focused on increasing recommendation accuracy by minimizing differences between the predicted recommendations and the known ratings in the matrix as exemplified by the Netflix prize (Bennett and Lanning 2007). However, with the observation that decrease in the errors in the matrix alone does not necessarily increase the quality of the recommendations, the current trend is to incorporate other external types of information into the problem.

A heterogeneous graph-based recommender system is an approach to integrate user ratings with other types of information following their semantic relationships, and represent them as a single graph. The types of entities in a graph are denoted as a node or an edge. Examples of nodes and edges for a movie rating system may be (*e.g.*, USER, MOVIE, ACTOR, DIRECTOR, TAG, ...) and edges (*e.g.*, USER $\xrightarrow{liked}$ MOVIE, MOVIE $\xrightarrow{directedBy}$ DIRECTOR, MOVIE $\xrightarrow{starring}$ ACTOR, ...). The advantage of the heterogeneous graph model is that a recommendation can be inferred by considering these heterogeneous entities and their relationships together, thus offering rooms for applying different types of recommendation approaches such as content-based filtering, collaborative filtering, or hybrid approaches that combine the two.

Generally, a user's process of finding his or her preferable items can be represented as a set of paths in a heterogeneous graph. For instance, let us consider a case that a user finds next favorable movies by examining the directors who directed some movies the user already likes, and see how this would be represented in a heterogeneous graph. Let us assume M_a and M_b, and D_a are of type MOVIE and DIRECTOR, respectively, and $\xrightarrow{liked}$ and $\xrightarrow{directedBy}$ are of edge types between the USER and the MOVIE, and the MOVIE and DIRECTOR. Then starting from the user U, a path ($U \xrightarrow{liked} M_a \xrightarrow{directedBy} D_a \xrightarrow{directedBy^-} M_b$) would denote how the user comes to find a next favorable movie, if M_b was found attractive to the user. Correspondingly, such a selection behavior of the user can be represented as USER $\xrightarrow{liked}$ MOVIE $\xrightarrow{directedBy}$ DIRECTOR $\xrightarrow{directedBy^-}$MOVIE, which we call a *path type*. In practice, a user's selection process should involve more node and edge types resulting in longer path types.

We postulate that a user's unique selection behaviors can be probabilistically extracted in terms of path types from a heterogeneous graph. More formally put, between a user U and a set of his favorable movies $\{M_1, M_2, \ldots, M_n\}$ in a graph, path types that are dominantly found in paths between U and $\{M_1, M_2, \ldots, M_n\}$ will faithfully represent the user's selection behaviors. Note, however, a path type found to represent a behavior of a user may not necessarily represents that of another user. For instance, a user may have many favorite actors, so he tends to prefer movies starring those actors. In this case, following a path whose path type is USER $\xrightarrow{liked}$ MOVIE $\xrightarrow{actedIn^{-1}}$ ACTOR $\xrightarrow{actedIn}$ MOVIE, starting from the node representing the user in the graph, has more potential of reaching at the user's preferable movies than following the other types of paths. On the other hand, another user may have preference for movies attached by the same specific tags like 'surreal' or 'artistic', then following USER $\xrightarrow{liked}$ MOVIE $\xrightarrow{hasTag}$ TAG $\xrightarrow{hasTag^-}$ MOVIE will tend to reach more at the user's preferable movies. In this paper, we introduce an algorithm named *PathMining* that produces the top-k most plausible path types for a user by probabilistically simulating the user's selection behavior. We particularly apply a Monte-Carlo method to draw a sufficient number of sample paths between a user and the favorite item set. This step is essentially to generate a large number of random-walk sequences over the graph. The sample paths are then subsequently analyzed to produce the resulting top k path types.

To test our algorithm, we constructed a heterogeneous graph using the HetRec 2011 Dataset[1] (Cantador, Brusilovsky, and Kuflik 2011) which is an extension of MovieLens10M dataset merged with information in corresponding web pages atInternet Movie Database (IMDb)[2] and Rotten Tomatoes movie review systems[3]. We selected several users and generated their path-based profiles. Our findings and insights from the analysis on the generated path-based profiles can be summarized as follows:

- Validity of a path type varies depending on users, and most users are profiled by a different sets of top-k effective path types.
- Path types which emulate well-known recommendation concepts such as content-based and collaborative filtering can be detected by our algorithm, and they are effective in general.
- Exploiting a personalized set of path types for recommending items has much potential of improving the performance as it will produce more tailored recommendation results for different users.

We organized the rest of this paper as follows. Section 2 presents research background. In section 3, we formally define our graph model and introduce the *PathMining* algorithm. In section 4, we show some results generated by the algorithm and discuss the insights and findings that we have achieved so far. Section 6 concludes the paper and discusses future work.

Related Work

In general, recommendation algorithms can be categorized into two main streams: Content-based Filtering (CBF) and Collaborative Filtering (CF). Whereas the CBF recommends items that have similar contents with a user's previously preferred items, the CF utilizes other users preferences on items to predict the given user's unknown preferences. For years, majority of researchers have focused on improving the recommendation accuracies. Some earlier graph-based recommendation methods (Fouss et al. 2007; Gori and Pucci 2007) also tried to improve accuracy by exploiting graph structures for recommendation. Later, it was shown that graph-based systems have advantages in incorporating heterogeneous types of information (Bogers 2010; Lee et al. 2011; 2013). Bogers et al. (Bogers 2010) introduced an algorithm named *ContextWalk* and showed how heterogeneous entities and their relationships can be integrated into the item ranking process by performing random-walks over a heterogeneous graph. Lee et al. (Lee et al. 2011) introduced how to build a heterogeneous bipartite graph with context information and utilize it for generating context-aware recommendation results.

Recently, several recommendation and node ranking approaches emphasized the importance of utilizing different *paths* in a heterogeneous graph as *paths* can represent rich semantics of relationships between entities. Lee et al. (Lee et al. 2013) introduced a node ranking algorithm named *PathRank*, which utilizes different types of paths in a heterogeneous graph, and showed that it can emulate collaborative filtering, content-based filtering, context-aware recommendation, and combinations of any of these recommendation semantics. Sun et al. (Sun et al. 2011), although in a slight different context, introduced *PathSim* which is a path-based similarity measure between two nodes in a heterogeneous graph. Both approaches addressed the importance of types of paths found, but neither approach proposed a systematic way of finding such paths.

Data Model

There can be many variations of heterogeneous graphs (*e.g.*, directed, undirected, property, weighted, ...). In this section, we formally define a heterogeneous graph and related concepts used in this paper.

Heterogeneous Graph: A heterogeneous graph is a directed graph $G = (\mathcal{V}, \mathcal{E}, T_{\mathcal{V}}, T_{\mathcal{E}})$, where $\mathcal{V}$ is a finite set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a finite multi-set of edges, $T_{\mathcal{V}}$ is a finite set of node types, and $T_{\mathcal{E}}$ is a finite set of edge types. Each node is mapped to a node type by a node type mapping function $\phi_{\mathcal{V}} : \mathcal{V} \rightarrow T_{\mathcal{V}}$, and each edge is mapped to an edge type by an edge type mapping function $\phi_{\mathcal{E}} : \mathcal{E} \rightarrow T_{\mathcal{E}}$. Each node $v_i \in \mathcal{V}$ or edge $e_k(v_i,v_j) \in \mathcal{E}$ has a set of ¡*attribute, value*¿ pairs which describe the properties of the node. Also, if $e_k(v_i,v_j)$ then, we say $head(e_k) = v_i$ and $tail(e_k) = v_j$.

Path and Path Type: A path p in a heterogeneous graph is defined as a sequence $<pe_1, pe_2, \ldots, pe_i, \ldots, pe_l>$, where $pe_i \in \mathcal{E} \bigcup \mathcal{E}^-$, and $\mathcal{E}^-$ is a set of reverse edges[4] of all $e \in \mathcal{E}$. For readability, we denote a path p by $pn_1 \xrightarrow{\phi_{\mathcal{E}}(\mathrm{pe}_1)^{\mathrm{D}_{\mathcal{E}}(\mathrm{pe}_1)}} pn_2 \xrightarrow{\phi_{\mathcal{E}}(\mathrm{pe}_2)^{\mathrm{D}_{\mathcal{E}}(\mathrm{pe}_2)}} \ldots \xrightarrow{\phi_{\mathcal{E}}(\mathrm{pe}_l)^{\mathrm{D}_{\mathcal{E}}(\mathrm{pe}_l)}} pn_{l+1}$, where $tail(pe_i) = pn_i$ and $head(pe_i) = pn_{i+1}$ for all $0<i\leq l$, and an edge direction mapping function $D_{\mathcal{E}}$: $\mathcal{E} \bigcup \mathcal{E}^- \rightarrow \{+,-\}$ is defined as $D_{\mathcal{E}}(pe_i)$ ='$+$' if $pe_i \in \mathcal{E}$ and $D_{\mathcal{E}}(pe_i)$ ='$-$' if $pe_i \in \mathcal{E}^-$. Each path p is mapped to its path type that is a sequence of node and edge types in the path by a path type mapping function $\phi_{\mathcal{P}}$, where it is defined as $\phi_{\mathcal{P}}(p) = \phi_{\mathcal{V}}(pn_1) \xrightarrow{\phi_{\mathcal{E}}(\mathrm{pe}_1)^{\mathrm{D}_{\mathcal{E}}(\mathrm{pe}_1)}} \phi_{\mathcal{V}}(pn_2) \xrightarrow{\phi_{\mathcal{E}}(\mathrm{pe}_2)^{\mathrm{D}_{\mathcal{E}}(\mathrm{pe}_2)}} \ldots \xrightarrow{\phi_{\mathcal{E}}(\mathrm{pe}_l)^{\mathrm{D}_{\mathcal{E}}(\mathrm{pe}_l)}} \phi_{\mathcal{V}}(pn_{l+1})$.

[1] http://grouplens.org/datasets/hetrec-2011/

[2] http://www.imdb.com

[3] http://www.rottentomatoes.com

[4] For an edge e from v_i to v_j, $tail(e) = v_i$ and $head(e) = v_j$. If $head(a) = tail(b)$ and $tail(a) = head(b)$, then both a and b are reverse edges to each other

Path-based User Profiling

In this section, we introduce the *PathMining* algorithm which quantifies effectiveness of path types by performing Monte-Carlo random-walk sequence samplings.

Algorithm 1 PathMining Algorithm

INPUT: A heterogeneous graph $G = (\mathcal{V}, \mathcal{E}, T_{\mathcal{V}}, T_{\mathcal{E}})$, a node $n_u \in \mathcal{V}$ where n_u represents a user u, a validation rating set R_u=$\{(u, i, r_i)$— user u's rating r_i on item $i\}$, l_{max}, n, item node type $nt_{item} \in T_{\mathcal{V}}$, cnt_{sample} and k.
OUTPUT: A path-based profile for the user u which is a set of top-k most effective path types $\mathcal{P}$=$\{pt_1$, pt_2,..., $pt_k\}$

$\mathcal{P} \leftarrow \{\}$; $cnt \leftarrow 0$; $seq \leftarrow \phi$; $history \leftarrow \{\}$
while $cnt{<}cnt_{sample}$ **do**
  if seq is ϕ **then**
    $seq \leftarrow n_u$; $n_{last} \leftarrow n_u$; $l_{path} \leftarrow 1$;
  end if
  while $l_{path}{<}l_{max}$ **do**
    $T_{adj} \leftarrow \{\}$;
    for all edge e_{adj} linked with n_{last} **do**
      Add $\phi_{\mathcal{E}}(e_{adj})$ to T_{adj};
    end for
    $et_{adj} \leftarrow$ Pick an edge type from T_{adj};
    /*move to an adjacent node by following an either forward or reverse et_{adj} type edge*/
    seq.join($n_{last} \xrightarrow{\mathrm{et_{adj}}^{+,-}} n_{adj}$);
    //join an edge representing a step with seq
    $n_{last} \leftarrow n_{adj}$; l_{path}++;
    if $\phi_{\mathcal{V}}(n_{last}) = nt_{item}$ and $(u, i, r_i) \in R_u$, where n_{last} represents an item i **then**
      $pt \leftarrow \phi_{\mathcal{P}}(seq)$;
      if $pt \notin history$ **then**
        $cnt[pt] \leftarrow 0$; $sum[pt] \leftarrow r_i$;
      else
        $cnt[pt]$+=1; $sum[pt]$+=r_i;
      end if
      Add seq to $history$; break;
    end if
  end while
  seq = ϕ; cnt++;
end while
for all pt_n in $history$ **do**
  if $cnt[pt]{>}avg(cnt[pt])$ **then**
    $score[pt] \leftarrow sum[pt]/cnt[pt]$;
  end if
end for
Add k path types with highest $score[pt]$ to $\mathcal{P}$;
return $\mathcal{P}$

Our main idea is to generate a large number of random walk sequences to simulate user's selection processes over the graph. Each random walk sequence starts from the given user node and continues following either forward or reverse edges in the graph. If the random-walker encounters an item node whose rating is in the validation rating set, the rating of the item is assigned as a score for the current random walk sequence. The validation rating set is a set of ratings that were not used to create the graph. If the validation rating set does not contain the corresponding rating, random walker continues the walk. The algorithm aggregates the scores of large number of random walk sequences grouping by the same path types and returns the top-k path types as a result. Algorithm 1 details the process.

Evaluation and Discussion

Evaluation Setup

To test our algorithm, we created a heterogeneous graph the HetRec 2011 dataset (Cantador, Brusilovsky, and Kuflik 2011) by using 70% of randomly selected ratings per each user, and we constructed a validation rating set for each user using the rest of the ratings. We assume that tags attached to a movie by a user without the user's rating on the movie are not valid, so we excluded such tags together as we excluded ratings for graph construction. The constructed graph is composed of nine node types USER, MOVIE, ACTOR, GENRE, DIRECTOR, TAG, LOCATION, COUNTRY, TAG_ASSIGN, and eleven edge types MOVIE $\xrightarrow{starring}$ ACTOR, MOVIE $\xrightarrow{directedBy}$ DIRECTOR, MOVIE $\xrightarrow{hasGenre}$ GENRE, MOVIE $\xrightarrow{filmedIn}$ LOCATION, MOVIE $\xrightarrow{hasOriginCountry}$ COUNTRY, USER $\xrightarrow{liked}$ MOVIE, USER $\xrightarrow{disliked}$ MOVIE, USER $\xrightarrow{neutral}$ MOVIE, TAG_ASSIGN $\xrightarrow{assignedTo}$ MOVIE, MOVIE $\xrightarrow{hasTag}$ TAG, TAG_ASSIGN $\xrightarrow{assignedTag}$ TAG, TAG_ASSIGN $\xrightarrow{assignedBy}$ USER We defined three edge types $\xrightarrow{liked}$, $\xrightarrow{disliked}$, $\xrightarrow{neutral}$ based on users' ratings on movies. For every rating r on a movie m by a user u, we created a $\xrightarrow{liked}$ type edge between nodes representing u and m, if $r{>}\bar{r} \times 1.10$, where $\bar{r}$ is the average rating on movies of all users. Similarly, a $\xrightarrow{disliked}$ type edge is created, if $r{<}\bar{r} \times 0.70$, otherwise $\xrightarrow{neutral}$ is created between the nodes. For the algorithm parameters, we set cnt_{sample}=100000, and l_{max}=3.

Results

Table 1. shows path-based profile results for three randomly selected users from our dataset. The result indicates that each user is found to have different profiles, where each user's profile is represented by top-k path types. For example, the path type USER $\xrightarrow{liked}$ MOVIE $\xrightarrow{liked^-}$ USER $\xrightarrow{liked}$ MOVIE is ranked higher than the path type USER $\xrightarrow{liked}$ MOVIE $\xrightarrow{hasTag}$ TAG $\xrightarrow{hasTag^-}$ MOVIE for the user 30500, but it is opposite for user 24695. The path type USER $\xrightarrow{liked}$ MOVIE $\xrightarrow{hasOriginCountry}$ COUNTRY $\xrightarrow{hasOriginCountry^-}$ MOVIE is ranked 5 for user 24695, but it is not ranked high for the other users. This variations in meaningful path types show much room for performance improvement as more tailored recommendation results will be expected for users if a personalized set of path types is accommodated.

It is notable that path types USER $\xrightarrow{liked}$ MOVIE $\xrightarrow{liked^-}$

Table 1: Path-based profiles constructed by PathMining

User	RANK	Path Type	Validity Score (Count)	Semantics
41391	1	$\text{USER} \xrightarrow{assignedBy^-} \text{TAG\_ASSIGN} \xrightarrow{assignedTag} \text{TAG} \xrightarrow{hasTag^-} \text{MOVIE}$	4.183 (1606)	Recommendation of movies with tags ever assigned by the user
	2	$\text{USER} \xrightarrow{liked} \text{MOVIE} \xrightarrow{liked^-} \text{USER} \xrightarrow{liked} \text{MOVIE}$	4.138 (127)	Collaborative Filtering
	3	$\text{USER} \xrightarrow{liked} \text{MOVIE} \xrightarrow{hasTag} \text{TAG} \xrightarrow{hasTag^-} \text{MOVIE}$	4.102 (340)	Content(Tag)-based Filtering
	4	$\text{USER} \xrightarrow{liked} \text{MOVIE} \xrightarrow{hasGenre} \text{GENRE} \xrightarrow{hasGenre^-} \text{MOVIE}$	4.036 (402)	Content(Genre)-based Filtering
	5	$\text{USER} \xrightarrow{liked} \text{MOVIE} \xrightarrow{directedBy} \text{DIRECTOR} \xrightarrow{directedBy^-} \text{MOVIE}$	4.011 (402)	Content(Director)-based Filtering
30500	1	$\text{USER} \xrightarrow{liked} \text{MOVIE} \xrightarrow{liked^-} \text{USER} \xrightarrow{liked} \text{MOVIE}$	3.758 (93)	Collaborative Filtering
	2	$\text{USER} \xrightarrow{disliked} \text{MOVIE} \xrightarrow{liked^-} \text{USER} \xrightarrow{hasTag^-} \text{MOVIE}$	3.682 (77)	-
	3	$\text{USER} \xrightarrow{liked} \text{MOVIE} \xrightarrow{hasTag} \text{TAG} \xrightarrow{hasTag^-} \text{MOVIE}$	3.679 (207)	Content(Tag)-based Filtering
	4	$\text{USER} \xrightarrow{neutral} \text{MOVIE} \xrightarrow{directedBy} \text{DIRECTOR} \xrightarrow{directedBy^-} \text{MOVIE}$	3.63 (98)	Content(Director)-based Filtering
	5	$\text{USER} \xrightarrow{disliked} \text{MOVIE} \xrightarrow{disliked^-} \text{USER} \xrightarrow{liked} \text{MOVIE}$	3.620 (116)	Collaborative Filtering
24695	1	$\text{USER} \xrightarrow{liked} \text{MOVIE} \xrightarrow{disliked^-} \text{USER} \xrightarrow{liked} \text{MOVIE}$	4.403 (88)	-
	2	$\text{USER} \xrightarrow{liked} \text{MOVIE} \xrightarrow{hasTag} \text{TAG} \xrightarrow{hasTag^-} \text{MOVIE}$	4.286 (354)	Content(Tag)-based Filtering
	3	$\text{USER} \xrightarrow{liked} \text{MOVIE} \xrightarrow{liked^-} \text{USER} \xrightarrow{liked} \text{MOVIE}$	4.273 (137)	Collaborative Filtering
	4	$\text{USER} \xrightarrow{disliked} \text{MOVIE} \xrightarrow{hasTag} \text{TAG} \xrightarrow{hasTag^-} \text{MOVIE}$	4.127 (114)	-
	5	$\text{USER} \xrightarrow{liked} \text{MOVIE} \xrightarrow{hasOriginCountry} \text{Country} \xrightarrow{hasOriginCountry^-} \text{MOVIE}$	4.099 (373)	Content(Country)-based Filtering

$\text{USER} \xrightarrow{liked} \text{MOVIE}$ and $\text{USER} \xrightarrow{liked} \text{MOVIE} \xrightarrow{hasTag} \text{TAG} \xrightarrow{hasTag^-} \text{MOVIE}$ are ranked high for all three users. The semantics of these path types can be interpreted as results of collaborative filtering and content(tag)-based filtering, respectively. It shows that our algorithm detected path types that might have been produced by two well-known ranking methods.

Some interesting path types such as $\text{USER} \xrightarrow{disliked} \text{MOVIE} \xrightarrow{disliked^-} \text{USER} \xrightarrow{liked} \text{MOVIE}$ (ranked 5 for user 30500) and $\text{USER} \xrightarrow{assignedBy^-} \text{TAG\_ASSIGN} \xrightarrow{assignedTag} \text{TAG} \xrightarrow{hasTag^-} \text{MOVIE}$ (ranked 1 for user 41391) are also detected. The first one shows that the user tends to like movies that are liked by other users who, by the way, disliked movies that the user dislikes. The semantics of such tendency can also be disclosed by collaborative filtering method, i.e. finding similar users using shared disliked movies. The second case shows that the user tends to like movies which has tags ever assigned by the user. It is interesting because it shows that tags assigned by users can be strongly related to user's preference, although the tags were assigned to movies, not users. In some cases (rank 2, rank 4 for user 30500 and rank 4 for 24695), edge type $\xrightarrow{disliked}$ or $\xrightarrow{neutral}$ appeared in the positions where $\xrightarrow{liked}$ is expected. It may be because we used the global average rating instead of each user's average rating for creating $\xrightarrow{liked}$, $\xrightarrow{disliked}$, and $\xrightarrow{liked}$ edges.

Conclusions and Future Work

In this paper, we introduced *PathMining*, a Monte Carlo method that constructs path-based user rating profiles, and, although preliminary, demonstrated the results indeed capture and explain a user's rating behaviors in terms of heterogeneous information types and their relations. For future works, we plan to design a recommendation method which can fully utilize profiles constructed by the *PathMining* algorithm. We will also extend our algorithm in many directions for further improvement. In particular, we will investigate incorporation of additional criteria for random-walk sequences such as diversity and novelty.

References

Adomavicius, G., and Tuzhilin, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on* 17(6):734–749.

Bennett, J., and Lanning, S. 2007. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, 35.

Bogers, T. 2010. Movie recommendation using random walks over the contextual graph. In *Proc. of the 2nd Intl. Workshop on Context-Aware Recommender Systems*.

Cantador, I.; Brusilovsky, P.; and Kuflik, T. 2011. 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In *Proceedings of the 5th ACM conference on Recommender systems*, RecSys 2011. New York, NY, USA: ACM.

Fouss, F.; Pirotte, A.; Renders, J.; and Saerens, M. 2007. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering* 19(3):355–369.

Gori, M., and Pucci, A. 2007. Itemrank: a random-walk based scoring algorithm for recommender engines. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, 2766–2771. Morgan Kaufmann Publishers Inc.

Lee, S.; Song, S.-i.; Kahng, M.; Lee, D.; and Lee, S.-g. 2011. Random walk based entity ranking on graph for multidimensional recommendation. In *Proceedings of the fifth ACM conference on Recommender systems*, 93–100. ACM.

Lee, S.; Park, S.; Kahng, M.; and Lee, S.-g. 2013. Pathrank: Ranking nodes on a heterogeneous graph for flexible hybrid recommender systems. *Expert Systems with Applications* 40(2):684–697.

Sun, Y.; Han, J.; Yan, X.; Yu, P. S.; and Wu, T. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *PVLDB* 4(11):992–1003.