Interactive Recommendations by Combining User-Item Preferences with Linked Open Data

Surya Kallumadi
Kansas State University
surya@ksu.edu

William H. Hsu
Kansas State University
bhsu@ksu.edu

ABSTRACT

Recent advances in graph and network embeddings have been utilized for the purpose of providing recommendations. Hybrid recommender systems have shown the efficacy of using side information associated with entities. In this work we show how domain specific knowledge can be used to define meta paths within these heterogeneous domains and how these path constrained random walks can be used to embed user preferences in heterogeneous domains. The semantic embeddings generated from heterogeneous knowledge sources combined with user preferences can be used to refine a user's information needs. This representation modeling of users, entities and their associated properties opens up new modalities of interactions for the users to gravitate towards their requirements. In this work we propose the use of semantic embeddings for two kinds of interactive recommendation modalities: 1) exemplar based recommendations 2) *"less like this/ more like this"* style recommendations. In our opinion providing these modalities would boost the expressive power of exploratory search and recommender systems.

KEYWORDS

Graph embeddings, Heterogeneous Networks, Distributed Representations, Interactive recommender systems

ACM Reference format:
Surya Kallumadi and William H. Hsu. 2018. Interactive Recommendations by Combining User-Item Preferences with Linked Open Data. In *Proceedings of 26th Conference on User Modeling, Adaptation and Personalization Adjunct, Singapore, Singapore, July 8–11, 2018 (UMAP'18 Adjunct),* 5 pages.
https://doi.org/10.1145/3213586.3226222

1 INTRODUCTION

Linked Open Data sources such as DBPedia and WikiData have become great sources for generating knowledge graphs for online entities. Knowledge graphs with typed entities and relationships can be viewed as instances of a Heterogeneous information network. Heterogeneous information networks are graphs wherein vertices and edges belong to one or more types $t \in \mathcal{T}$. Although most information networks can be viewed as heterogeneous, researchers and practitioners typically ignore the type designations when performing various analytical tasks. One such task is the *recommendation of entities in heterogeneous information networks*, which we define as follows: given a heterogeneous information network $\{V, E\} = G$ and a preference matrix $\mathcal{W}$, we aim to predict the top k vertices $< v_1, v_2, \ldots, v_k >$ of type v associated with some input vertex u.

Personalized PageRank [10], SimRank [2], Deepwalk [7], LINE [9] and other models have been applied for the recommendations task in homogeneous information networks with good results; however, the type signal provided by heterogeneous information networks allows the user to specify what *type* of information to return. Recent work in Hybrid recommender systems have demonstrated the efficacy of using knowledge graphs to improve recommendations [4]. Musto et. al. in [5][4] use linked open data and PageRank style features to improve the performance of Graph oriented recommender systems. Path based approaches that are aware of node type in a graph was proposed in [11] for personalized recommendations. Graph embedding on random walks over graphs and using embedding similarities from *node2vec*[1] was proposed in *Entity2rec* [6], this approach learns a user, item relatedness in the context of a intermediate property.

1.1 User driven Interactive Recommendations

In addition to typed recommendations, the presence of multiple types of entities within a heterogeneous information network, opens up new modalities for specifying user intent and retrieving recommendations to the users. Query-based information retrieval is one of the primary ways in which meaningful nuggets of information is retrieved from large amounts of data. Here the query is represented as a user's information need. In a homogeneous network, in the absence of type and contextual side information, the retrieval context for a user reduces to the user's preferences over observed items. In a heterogeneous setting, information regarding entity types and preference context is available. Thus, query-based contextual recommendations are possible in a heterogeneous network. The contextual query could be type-based (e.g. directors, actors, movies, books etc.) or value-based (e.g. based on tag values, genre values such as "Comedy", "Romance") or a combination of Types and Values. Thus a user can express not only queries such as *"Generate movies that I like"*, but also queries such as ***"Generate movies that I like" + "similar to the movie 'Saving Private Ryan'" + "war movie" + "drama movie" - "comedy movie"***.

Existing Meta Path based approaches such as PathSim [8] have explored the task of Top-K similarity search in HIN. One limitation of these approaches is that the approaches only allows for obtaining pair wise similarity of entities and cannot support combination queries that involve multiple types of entities. In our proposed approach as all the typed entities within the HIN are represented

UMAP'18 Adjunct, July 8–11, 2018, Singapore, Singapore

ACM ISBN 978-1-4503-5784-5/18/07…$15.00
https://doi.org/10.1145/3213586.3226222

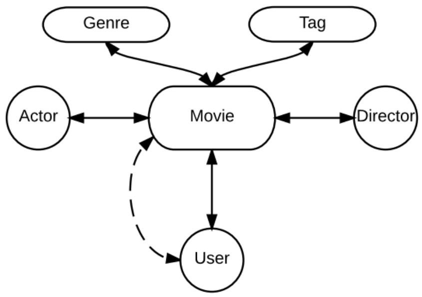

Figure 1: Movie - Knowledge Graph Schema

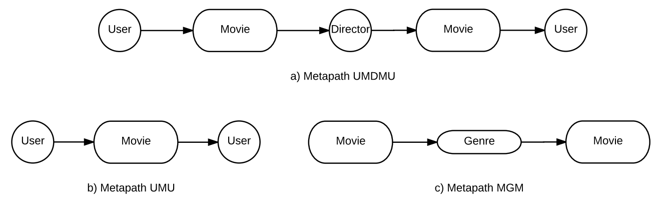

Figure 2: Movie Network Meta Paths

Table 1: Metapaths captured from IMDB schema

IMDB Metapaths
user - movie
user - movie - director - movie
user - movie - actor - movie
user - movie - genre - movie
user - movie - language - movie
user - movie - keyword - movie
movie - genre - movie - director
director - movie - actor - movie
director - movie - genre - movie
language - movie
keyword - movie

in the same space, we have the ability to retrieve recommendations for complex contextual queries.

We evaluate the effectiveness of query-based recommendations using network embeddings by applying them to the list completion task. Experiments conducted on the user curated movie lists in the IMDB data set for set completion problem demonstrates the usefulness of distributed representations. We use Precision@K metric to compare the user lists generated with the ground truth user lists.

Following is a summary of the contributions of this paper:

- We propose a meta path based path constrained random walk framework to capture the interaction patterns between entities in the HIN and generate distributed representations of these entities.
- We propose a query-based user driven interactive recommendation approach using the Meta Path based HIN embeddings. This allows the user to define the context for the recommendations. The query context is expressed as a combination of entities present in the HIN. We demonstrate the efficacy of this approach by using the HIN embeddings for the List completion task. We propose 2 kinds of user interface modalities for User driven interactive recommendations 1) exemplar based recommendations 2) *"less like this/ more like this"* style recommendations

2 PROBLEM DEFINITION

In this section we provide an overview of of Heterogeneous Information Networks and introduce MetaWalk a meta path embedding approach for recommendations and retrieval in HIN. Figure 1 illustrates the various kinds of interactions that can exist with the Movie domain. Traditional recommender systems were restricted to a bipartite network between Movie and User entities with rating edges connecting them.

Definition 2.1. **Heterogeneous Information Network (HIN):** A weighted HIN is defined as a directed graph $G = \{V, E\}$ with an entity mapping function $\phi : \mathbb{V} \rightarrow A$ and a edge type mapping function $\psi : \mathbb{E} \rightarrow R$ where each node v $\in \mathbb{V}$ belongs to one particular entity type ϕ(v) $\in \mathcal{A}$ and each edge e $\in \mathbb{E}$ belongs to a relationship type ψ(e) $\in \mathcal{R}$. The edge weights associated between vertices with the relationship context ψ(c) $\in \mathcal{R}$ is captured as a preference matrix $\mathcal{W}_c$.

Definition 2.2. **Meta Path**: Meta path is a sequence of edges specified over a HIN schema $\mathcal{S}_G = (\mathcal{A}, \mathcal{R})$. The Meta Path defines a composite relationship that consists of ordered sequence of edge types specified in the HIN schema.

Figure 2 illustrates some of the metapaths derived from the Movie network schema specified in Figure 1 (b). The UMU meta path specified in Figure 2(b) illustrates the rating semantic relationship between the entities User and Movie. This relationship is captured in the User-Rating-Movie bipartite network, where the edges exist as rating relationship between users and movies. Similarly the MGM meta path specified in Figure 2(c) illustrates the Movie attribute semantic relationship between the set of movies and the genres associated with these movies.

Table 1 enumerates the various metapaths that we specify over IMDB schema. These domain aware metapaths are used as blueprints for performing random walks. We use the meta path based random walk algorithm described in Algorithm 1 to perform weighted meta walks over a HIN.

3 DISTRIBUTED REPRESENTATIONS AND HIN EMBEDDINGS

Distributed representations have become a popular and effective way to represent higher-dimensional information. Distributed Representations have found a wide degree of applications in the fields of machine learning, natural language processing and information retrieval. In the MetaWalk framework we aim to represent HIN into a low dimensional representation while ensuring that the structural and semantic properties of the HIN are preserved. To obtain information network embeddings we first specify the set of meta paths with the network that capture semantic and structural patterns and then generate meta path constrained random walks over the network. Once these random walks are generated, we use the Skip-gram approach specified in [3] to generate information network embeddings that capture semantic relationships captured by user curated lists.

3.1 SkipGram

The skip gram approach is a neural network approach for generating distributed representations over contextual windows. Given an entity e, the skip gram model tries to predict the surrounding context entities that are present within the random walk. The context for an entity in a path constrained random walk is defined by a window around the entity. For example in the User - Movie meta path, the context for users would be movies that they prefer and similar Users that have shown preference to the Movie. This meta path essentially captures the collaborative filtering hypothesis of similar users show similar preferences over items, for recommendations generation. For each entity and the context available a Softmax function acts on the output layer activations for each input context Entity. This approach tries to maximize the co-occurrence probability of entities that appear within the same context.

3.2 SkipGram for meta paths

In the case of User-Item metapath, this relationship spans the bipartite graph of Users and items. We then generate a randomwalk by traversing this bipartite graph. The semantic meaning of these random walks can be expressed as similar users will have semantically similar movies. For a sequence of items $\{s_i\}_{i=1}^{K} \subseteq S$ we maximize the following term:

$$\frac{1}{K}\sum_{i=1}^{K}\sum_{j\neq i}^{K}\log\left(\sigma(u_i v_j)\prod_{k=1}^{N}\sigma(-u_i v_k)\right)$$

where $u_i \in U$ is the latent vectors of target entity and $v_i \in V$ is the latent vectors of other entities in the context window around target entity.

$$\sigma(x) = 1/1 + exp(-x)$$

N is the number of negative samples drawn that would be used to minimize the similarity between the target entity and the item not present in the item context. The negative items are sampled from the item distribution. The latent embeddings of items are estimated by using stochastic gradient descent that maximizes the above criteria. We use *Cosine similarity* to calculate similarity between two latent representations. We use additive vector composition(AVC) to come up with a single representation from multiple representations if we want get a single representation for a set of entities.

Skip-Gram for Homogeneous Paths: When all the nodes in a metapath are of the same *type* we can model the conditional probability $\Pr(v_j|\Phi(v_i))$ to be independent of the *type* of the node v_j, and can be derived by softmax as:

$$\Pr(v_j|\Phi(v_i)) = \frac{\exp(\Psi(v_j)\cdot\Phi(v_i))}{\sum_{u\in V}\exp(\Psi(u)\cdot\Phi(v_i))}. \tag{1}$$

Skip-Gram For Metapaths that assumes the probability $\Pr(v_j|\Phi(v_i))$ is related to the type of node v_j:

$$\Pr(v_j|\Phi(v_i)) = \Pr(v_j|\Phi(v_i), \phi(v_j))\Pr(\phi(v_j)|\Phi(v_i)), \tag{2}$$

Table 2: IMDb dataset overview

sparsity	5	10	50	100
user	18414	18414	18414	18414
movie	30408	40837	73571	91070
rating	323979	648847	3249149	6494480
actor	35911	46475	79633	96910
director	13042	16843	27922	33493
genre	26	27	28	28
language	216	220	256	259

and can be derived by softmax as:

$$\Pr(v_j|\Phi(v_i), \phi(v_j)) = \frac{\exp(\Psi(v_j)\cdot\Phi(v_i))}{\sum_{u\in V,\phi(u)=\phi(v_j)}\exp(\Psi(u)\cdot\Phi(v_i))}. \tag{3}$$

4 EXPERIMENTS AND RESULTS

In this section, we present the empirical analysis of the proposed recommendation framework that leverages network schema of the Heterogeneous domain to generate recommendations. We performed a series of experiments to demonstrate the effectiveness of our MetaWalk model on IMDb.com domain. In this movie domain, we apply the proposed algorithm to recommend a personalised Top-N movie-list for the user.

4.1 Dataset

To demonstrate the effectiveness of the proposed recommendation framework, we choose the movies domain. We represent the movies domain dataset as heterogeneous network. In the IMDb dataset, we use user ratings and meta-data of movies that captures side information in order to build a heterogeneous information network. The different entities in our graph are users, movies, actors, directors and genre. The HIN graph consists of both weighted and unweighted edges over different typed entities. The edges between user and movie entities are weighted. The edges about the meta-data of movies are unweighted. The weighted edges represent the degree of affinity between two nodes. For example, if a user rated a movie, we assign a weight to the user-movie edge as a function of user's rating. On the other hand, unweighted edges represent meta-data about the movies e.g. a movie-genre edge is created if movie belongs to that particular genre. We filter out all the users who have rated less than 20 movies while building the dataset. The dataset is divided into train and test set as follows. For each user, we retain 50% user-movie edges in train dataset, and rest is treated as test dataset. In order to show the effectiveness of our proposed approach, we further divide training set into sparse sets based on number of edges. We create four sets with 5%, 10%, 50% and 100% randomly sampled edges conditioned on user from the train dataset. The four sparse sets are described in Table 2.

4.2 Evaluation metrics

To measure the efficacy of top-k item recommendations using various proposed approaches, we use precision@K. In the case of Imdb

Table 3: Results

	10% Data		50% Data		100% Data	
	P@10	P@20	P@10	P@20	P@10	P@20
LINE	0.082	0.132	0.083	0.133	0.089	0.143
DeepWalk	0.118	0.187	0.141	0.218	0.169	0.246
Node2Vec	0.081	0.132	0.082	0.133	0.083	0.134
MetaWalkU	0.113	0.182	0.116	0.183	0.122	0.191
MetaWalkW	0.159	0.238	0.18	0.262	0.185	0.271

Table 4: Exemplar based user list generation for "Bond Movies"(Left) and "Film Noir"(Right) Movie titles in bold constitute the query set

Dr. No	**Maltese Falcon**
Goldfinger	Double Indemnity
You Only Live Twice	The Big Sleep
Thunderball	Key Largo
On Her Majesty's Secret Service	The Third Man
From Russia with Love	Laura
Moonraker	Out of the Past
Diamonds Are Forever	To Have and Have Not
The Man with the Golden Gun	White Heat
For Your Eyes Only	Dark Passage
The Living Daylights	The Lady from Shanghai

data set we treat the recommendation task as recommending unseen movies to the users. We treat this as a Top-K recommendation task and not as rating completion task.

4.3 Results

Table 3 shows the empirical results for Precision@K in the Imdb.com domain. MetaWalk approach that leverages domain information in the HIN schema outperforms our baseline approach. MetaWalkU - Unweighted MetaWalk is the approach that generates random walks without considering the user-item ratings. MetaWalkW - Weighted MetaWalk is the approach that generates random walks while taking into account the user-item ratings.

5 INTERACTIVE RECOMMENDATIONS

The semantic embeddings can be used to refine a user information need by providing exemplars. This opens up new modalities of interactions for the users to gravitate towards their requirements.

5.1 Exemplar based recommendations

Table 4 shows the result of exemplar based list completion for the user information need "Bond" movies and "Film Noir" movies, as the community structure of these movies is tight knit, our list generation approach generates a good set of movies with query set of size 1. This interactive modality can be viewed as a List refinement task, where the user is providing examples and refining the result set of items based on an intent that they do not know how to express. Table 5 shows the result of exemplar based list completion for the user information need "new romantic comedy" movies, by adding more examples into the query set the user can tweak the membership of the list generated. The rank location of the candidate items changes as the query set membership is varied.

Table 5: Exemplar based user refinement for the intent "newer romantic comedies", Movie titles in bold constitute the query set

The 40-Year-Old Virgin	**The 40-Year-Old Virgin**
Wedding Crashers	**Knocked Up**
Waiting...	Superbad
Anchorman...	Wedding Crashers
Step Brothers	The Heartbreak Kid
EuroTrip	Anchorman...
Harold & Kumar...	EuroTrip
The Ringer	Sex Drive
The 40-Year-Old Virgin	**The 40-Year-Old Virgin**
Knocked Up	**Knocked Up**
Anchorman...	**Anchorman...**
Superbad	**Step Brothers**
Wedding Crashers	Superbad
EuroTrip	Wedding Crashers
Harold & Kumar...	Role Models
Step Brothers	Harold & Kumar...

5.2 *Less Like this and More Like this*

As all the entities(users, items, item attributes) are represented in the same space and as the user embeddings are personalized, we can retrieve the resultant entities that match the query expressed by the user. Notice here that entities need not be associated with the attributes such as "war", "comedy" etc. to qualify for this as they are represented semantically. We can also express queries such as *"more like this"* and *"less like this". e.g.* ***"generate movies more like 'Casino Royale' and less like 'Golden Eye' "***. The presence of positive and negative interactions between the users and the entities can also be leveraged to provide *Less Like this and More Like this* interactive interfaces (where High ratings/ "thumbs up" can be considered as a positive interaction). We believe that interactive search and recommendations interfaces can be highly effective where user's are in an *exploratory* phase and are not familiar with the domain. In these cases, these novel modalities of interaction can prove to be highly effective. This is a less researched area and more work needs to be done in this field, especially with respect to metrics to identify effectiveness of such interface modalities.

6 CONCLUSION AND FUTURE WORK

In this work we propose approaches for providing recommendation in a heterogeneous information network. We propose a meta path based path constrained random walk framework to capture the interaction patterns between entities in the HIN and generate distributed representations of these entities. We show that using ratings as strength of association between the entities and by using non-linear associations we can perform recommendations in a HIN with weighted edges. We show how the representations obtained by a metawalk based graph embeddings approach can be used for interactive recommendations. Future work will investigate: (1) combining MetaWalk based approaches with collaborative filtering algorithms to overcome cold start, (2) domain transfer for various tasks between related domains (e.g. books and movies), (3) Using metawalk similarities for explaining recommendations generated by collaborative filtering algorithms.

REFERENCES

[1] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. *CoRR* abs/1607.00653 (2016). arXiv:1607.00653 http://arxiv.org/abs/1607.00653

[2] Glen Jeh and Jennifer Widom. 2002. SimRank: A Measure of Structural-context Similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, 538–543. DOI: http://dx.doi.org/10.1145/775047.775126

[3] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013). http://arxiv.org/abs/1301.3781

[4] Cataldo Musto, Pierpaolo Basile, Marco Degemmis, Pasquale Lops, Giovanni Semeraro, and Simone Rutigliano. 2015. Automatic Selection of Linked Open Data Features in Graph-based Recommender Systems. In *CBRecSys@RecSys*.

[5] Cataldo Musto, Pierpaolo Basile, Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2017. Introducing Linked Open Data in Graph-based Recommender Systems. *Inf. Process. Manage.* 53, 2 (March 2017), 405–435. DOI: http://dx.doi.org/10.1016/j.ipm.2016.12.003

[6] Enrico Palumbo, Giuseppe Rizzo, and Raphaël Troncy. 2017. Entity2Rec: Learning User-Item Relatedness from Knowledge Graphs for Top-N Item Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. ACM, New York, NY, USA, 32–36. DOI: http://dx.doi.org/10.1145/3109859.3109889

[7] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*. ACM, New York, NY, USA, 701–710. DOI: http://dx.doi.org/10.1145/2623330.2623732

[8] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *In VLDBâĂŽ 11.*

[9] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15)*. ACM, New York, NY, USA, 1067–1077. DOI: http://dx.doi.org/10.1145/2736277.2741093

[10] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast Random Walk with Restart and Its Applications. In *Proceedings of the Sixth International Conference on Data Mining*. IEEE, Washington, DC, USA, 613–622. DOI: http://dx.doi.org/10.1109/ICDM.2006.70

[11] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized Entity Recommendation: A Heterogeneous Information Network Approach. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM '14)*. ACM, New York, NY, USA, 283–292. DOI: http://dx.doi.org/10.1145/2556195.2556259

Algorithm 1 Meta Path based Random Walk generation for weighted graphs

```
Data: initial node n of type T, meta path P, length of randomwalk L
/* meta path P is a Circular Queue data structure */
Result: weighted random walk w
w ={}
/* random walk w is a Queue data structure */
Enqueue(w,n)
while iter ≤ L do
    nextType=Next(P)
    /* getting the next edge type in meta path */
    if isWeighted(nextType) then
        /* if next Edge in the metapath is a weighted edge */
        WNeighborhood= GetNeighbors(n,nextType)
        /* Get the Typed neighbors From the current Node */
        n = WeightedSampleNeighborhood(WNeighborhood)
        /* Perform weighted sampling in the Typed Neighborhood */
        Enqueue(w,n)
    else
        Neighborhood= GetNeighbors(n,nextType)
        /* Get the Typed neighbors From the current Node */
        n = UniformSampleNeighborhood(WNeighborhood)
        /* Perform uniform sampling in the Typed Neighborhood */
        Enqueue(w,n)
    end
end
```