Social Web Recommendation using Metapaths

Robin Burke and Fatemeh Vahedian
Center for Web Intelligence
DePaul University
243 S. Wabash Ave
Chicago,IL
(rburke, fvahedia)@cs.depaul.edu

ABSTRACT

The social web is characterized by a wide variety of connections between individuals and entities. A challenge for social web recommendation is make the most effective use of a diverse set of relations. Typically, researchers focus on a limited set of relations (for example, person to person ties for user recommendation or annotations in social tagging recommendation). In this paper, we propose a general approach to recommendation in social networks that can incorporate multiple relations in combination. A key feature of this approach is the use of the *metapath*, an abstraction of a large class of paths in the network in which edges of different types are traversed in a particular order. As a preliminary demonstration, we show that our approach yields improvements over a state-of-the-art technique on several social tagging datasets.

General Terms

Hybrid recommender system, heterogeneous network, social tagging system

Keywords

1. INTRODUCTION

The social web is characterized by a diversity of data types and relations. For example, the employment-oriented website LinkedIn contains information about individuals, companies, jobs and skills, and connections can be drawn among any of these entities. There are also discussion forums and user groups. Diversity of information means that there are many kinds of recommendation that can be made to users: other users with whom to connect, groups to join, skills to acquire, companies to follow, jobs to apply for, etc. At the same time, the complexity of information means that there are many more types of information that can be integrated into making recommendations: should the system recommend companies that have hired your friends, those that have many (or few) employees with your skills? Often building recommenders for such sites involves devising individual ad-hoc solutions for each recommendation problem.

To illustrate this problem, consider a user Alice who is a member of the Last.fm web site for music lovers, looking for a song to add to her current playlist:

Track	Song	Artist
1	Bad Girls	Blood Orange
2	Under the Gun	Supreme Beings of Leisure
3	The Sea	Morcheeba
4	Paris Train	Beth Orton

We might expect that a suitable song would also be mellow electronica featuring a female vocalist but there will be a very large number of tracks with these characteristics. We can discriminate among these tracks using data from the Last.fm social network, as summarized in the schema in Figure 1.

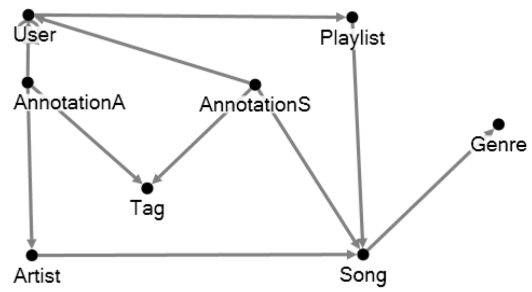


Figure 1: Network schema for Last.fm

As the schema shows, a given song may have many possible associations. It may appear on multiple playlists; it may have been tagged by one or more users (AnnotationS); it may be associated with one or more artists. We can select any of these data sources, and build a recommender system with that basis. For example, using a user-based collaborative approach we could look at similarities across playlists or across tagging histories. Any such choice inevitably excludes a great deal of possibly-relevant knowledge.

Ideally, we would like a recommendation method that is integrative – bringing all of the available data to bear. In this paper, we describe one such technique: the Weighted Hybrid of Low-Dimensional Recommenders (WHyLDR). The WHyLDR technique was originally developed for social tag-

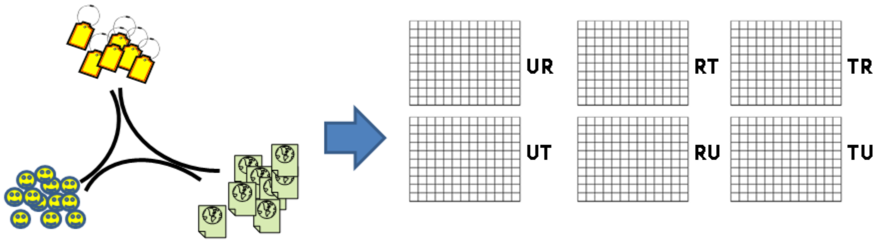


Figure 2: Two-dimensional projections for a social tagging network

ging systems [7]; here we show how the concept can be extended to more complex networks.

The key insight of the WHyLDR design is that a complex network structure can be viewed as a set of two-dimensional projections from nodes of one type to nodes of another. Figure 2 illustrates this idea in the case of social tagging systems. The tagging system on the left has annotations consisting of user, tags and web resources the users have tagged. One projection (the UT projection) maps each user to the set of tags that user has applied. Another projection (UR) maps the user to the resources he or she has tagged. Other projections link resources to tags and to users: six such projections in total.

Given a two-dimensional representation, such as user represented by tags, it is quite straightforward to apply standard collaborative recommendation methodology: find neighborhoods of similar users and make recommendations on that basis. with a hybrid recommendation approach, it is not necessary to choose just one of these projections as the source of data: a recommendation can be made by combining the results of recommendation components built from these low-dimensional projections. Our previous work has shown that a linear weighted hybrid build of such components is more flexible and more accurate than integrative techniques that attempt to model all of the dimensions at once [7].

We extend this idea to more complex networks through the concept of the *metapath* [15]. A path in a network is a sequence of edges that can be traversed to move from one node to another. A metapath is an abstraction of a network path into a sequence of edge types. Navigating a metapath from a node reaches all destination nodes reachable by following edges with the appropriate type. For example, in the music recommendation scenario, we might have the SPU metapath $\langle song \rightarrow playlist \rightarrow user \rangle$. This path goes from a song to all playlists into which it is a part and then to all users contributing those playlists. A different metapath would go from a song to all annotations in which it appears to all users creating such annotations: $\langle song \rightarrow annotationS \rightarrow user \rangle$, denoted SAsU. Note that both the SPU and SAsU metapaths map songs to users, but they follow different routes through the network.

A metapath can be used to generate a two-dimension projection where each originating node is mapped to all of the terminating nodes reachable by following the path. For example, the SPU metapath can be used to generate an item-based matrix where each song is represented in terms of the users that have incorporated it into a playlist.

A metapath can be arbitrarily long although we anticipate very long paths may not be very useful. Metapaths may also contain multiple occurrences of the same object type. For example, the songs on the playlists of the user's friends of friends can be expressed via the UUPS metapath $\langle user \rightarrow user \rightarrow playlist \rightarrow song \rangle$.

2. RELATED WORK

The integration of social network data into recommender systems has been studied extensively in recent years [17], [4], [13],[14]. Most of this work has been focused on system-specific solutions. For example, [10] shows a LastFM music recommendation based on combination of social data and annotation system. A similar system incorporating social data and tags has been used to recommend publications in the Bibsonomy dataset [3]. A more general technique is the multi-relational approach of [2] in which the heterogeneous network in Epinions is separated into multiple homogeneous networks and then an optimization approach is used to find the best combination of recommendations coming from the different networks. Kazienko and his colleagues [9] take a similar approach, treating the different kinds of relations in Flickr as "layers."

Our domain-independent approach for recommendation with social network data draws heavily on recent research in the area of complex heterogeneous information networks. According to Han[8], heterogeneous networks are "information systems which consist of a large number of interacting, multi-typed components". In particular, heterogeneous information networks involve multiple types of objects and multiple types of links denoting different relations [16].

Sun and Han [15] argue that information propagation across heterogeneous nodes and links can be very different from that across homogeneous nodes and links. To capture this diversity, the authors defined the concept of the "metapath". On top of the metapath abstraction, they were able to build algorithms operating on heterogeneous networks such as metapath-based similarity search.

As discussed above, this work is an extension of research applying linear weighted hybrids to recommendation problems in social tagging systems. This work employed a collection of recommendation components including the two-dimensional projection components built as described above and used random-restart hill climbing to optimize the contribution of each component. This technique is both simple and general. Our results showed that it was at least as effective as other, more computationally-sophisticated techniques for the well-studied problem of tag recommendation, with the added advantages that it could be applied to a wider variety of recommendation problems and could be more easily updated. See [5, 6, 7] for more detail on this line of research.

There is a close relationship between recommendation in a network setting and link prediction, which is a standard problem in the computational study of networks [11]. For example, in our playlist example, if the system recommends a track and Alice adds it to her playlist, this will become a new $\langle playlist \rightarrow song \rangle$ link in the network. However, there are some important distinctions between link prediction, as it is customarily studied, and the problem of recommendation. Foremost is the difference of emphasis demonstrated in the output of the system. In link prediction, the output of the system is a set of links likely to appear in the network. In recommendation, we are suggesting items for an individual and personalization is therefore a key element. We could filter the link predictions just to those that apply to the current user, but it is important to recognize that link prediction techniques are not really designed or evaluated with personalized presentation in mind. Secondly, recommendation often involves a host of additional considerations (serendipity, diversity, etc.) that are not typically factors in link prediction analysises.

3. LINEAR-WEIGHTED HYBRID

For the present discussion, we assume that a recommender is a function that takes a user as input and returns a ranked list of recommended items. One common way to implement such a recommendation function is to build it on top of a scoring function $s(u, i)$, where u is a user and i is an item. If we can calculate a score for each item available for recommendation, we can sort the items and present the best items to the user.

A weighted hybrid recommender is therefore a scoring function that forms a weighted sum of the results of its constituent components [1]

$$s'(u, i) = \sum_{s_i} \alpha_i * s_i(u, i) \tag{1}$$

where the s_i's are the recommendation components and α_i's are the associated weights. To define a weighted hybrid, we need to specify its components and their weights.

3.1 Recommendation Components

The components needed for a hybrid recommender are a function of the recommendation task and the data available to support recommendation. In our work on social tagging systems, we identified a number of recommendation tasks appropriate to that context, including tag recommendation, resource recommendation, resource recommendation by example, user recommendation, and others. Resource recommendation is the task of identifying items of interest for a user in social tagging system based on tagging behavior. Note that these items may or may not be items that the user "likes" – a user may frequently tag disliked items with deprecatory tags, for example.

In the experiments reported in [7], the system used the following recommendation components:

- Popular: A non-personalized recommender that scores resources based on their overall popularity.
- User-based kNN, user-tag matrix (kNN_{UT}): A user-based collaborative recommendation component in which users are compared by their usage of tags. The entries in this matrix are normalized counts – the fraction of annotations in which a user has employed a given tag. Pearson correlation is used to compare users and Resnick's algorithm is used to compare predictions.
- User-based kNN, user-resource matrix (kNN_{UR}): As above, but where users are compared on the basis of which resources they have tagged. In this matrix, we did not find any benefit to make use of the count information: the number of tags that a user applied to a given resource. The matrix is therefore binary, reflecting whether or not the user tagged a particular resource. Predictions are computed as with kNN_{UT}.
- Item-based kNN, resource-tag matrix (kNN_{RT}): Item-based collaborative recommendation in which resources are compared on the basis of the tags that have been associated with them. This matrix is similar to kNN_{UT}, but instead of users, we are profiling resources. To make predictions, we use the adjusted cosine method from [12]. The predicted relevance of a resource is a function of the normalized tag counts of similar resources. Note that this component is not personalized: it will give the same predictions for all users.
- Item-based kNN, resource-user matrix (kNN_{RU}): Item-based collaborative recommendation in which resources are compared on the basis of the users who have tagged them. This matrix is the transpose of the UR matrix, and is also binary. Adjusted cosine is used here as well.
- Cosine: In this component, the user is represented as the vector of tags they have applied, normalized as in kNN_{UT} and each resource is represented as a vector of tags that have been applied to it as in kNN_{RT}. The scoring of a resource for a user is done by computing the cosine between the two vectors.

3.2 Heterogeneous Networks

Following Sun and Han [15], we define a heterogeneous information network as a directed graph $G = (\nu, \varepsilon)$ with an object type mapping function $\gamma : \nu \rightarrow A$ and a edge type mapping function $\phi : \varepsilon \rightarrow R$ where each object belongs to particular object type $a \in A$ and each edge belongs to a particular relation type $r \in R$. Two edges of the same type by definition share the same object types at their originating and terminating points.

A heterogeneous network is one where there are multiple object types and/or multiple edge types – typically both. For example, the music example above is clearly a hetereogeneous network. There are multiple types of nodes (artists, users, songs, etc.) and multiple types of relations (user-user, user-playlist, artist-song, etc.). A network schema, such as that shown in Figure 1, gives an overview of a heterogeneous network by indicating the different object types and the relations that exist between them. A metapath in a heterogeneous network is a path over the network schema, a sequenced composition of relations between two object types.

3.3 Metapath-Based Recommendation Components

A social tagging system can be viewed as a heterogeneous network with four different types of nodes (users, tag, resources, and annotations). See Figure 3. With this in mind, consider the UR projection on which the kNN_{UR} component is built. This is a matrix in which the rows correspond to users and the columns correspond to resources, and the entries reflect the whether or not the user has tagged that particular resource. We can generate the same matrix using the schema shown in Figure 3 by following the metapath $\langle user \rightarrow annotation \rightarrow resource \rangle$. Since the schema has a simple star structure, we will omit the reference to the central annotation node (all navigation must go through it) and refer to this as the UR metapath.

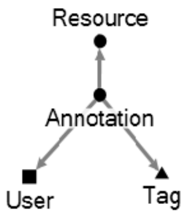


Figure 3: Network schema for Social Tagging Sytems

Adopting the metapath formalism allows us to express a much wider set of possible projections. We can expand the set of resources by which a user is represented by following an extended metapath: $\langle user \rightarrow annotation \rightarrow tag \rightarrow annotation \rightarrow resource \rangle$ or UTR for short. This path finds all tags a user has employed and then all annotations including those tags (even those not created by the user) and then the resources for that larger set of annotations. This can be seen as a kind of "query expansion" of the resource space by considering other users' annotations of the same resources.

To see how this process works, consider Figure 4, which shows a simplified network with 3 users having tagged three music tracks.

The UR matrix for this network is as follows:

	Track1	Track2	Track3
Carol	1	1	0
Bob	0	1	0
Alice	0	0	1

There are four annotations (A1...A4) because Carol has created two. Note that Bob and Alice have no tagged tracks in common.

However, if we follow the UTR metapath, the fact that Bob and Alice have both used the tag "Relaxing" means that the UTR metapath yields both Track2 and Track3 for these users. The resulting matrix shows a similarity between these two users that was not present in the UR version.

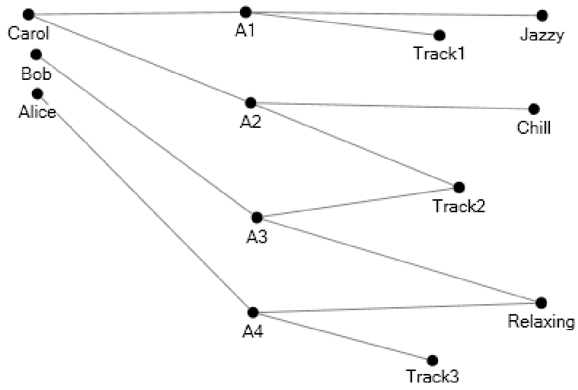


Figure 4: Simple Tagging Network

	Track1	Track2	Track3
Carol	1	1	0
Bob	0	1	1
Alice	0	1	1

Of course, this process can be extended indefinitely: UTTR, UTTTR, etc. We can envision in addition a wide variety of other metapaths: for example, UTUR would be all resources tagged by users who share tags with the target user. For our preliminary investigation of this style of recommendation, we opted to explore only a few possible components using short metapaths. We created three additional components to augment the six already incorporated in the system described in [7].

- User-based kNN with the user-tag matrix formed by following the URT metapath: kNN_{URT}.
- User-based kNN with the user-resource matrix formed by following the UTR metapath: kNN_{UTR}.
- A version of the Cosine metric above in which the vector of tags for a user is formed using the URT metapath: *Cosine-M*.

4. DATASETS

For our experiments, we used three social tagging networks from the data sets studied in [7]. All of the data sets were filtered as described in [7] to eliminate rare and idiosyncratic tags and resources.

- **Bibsonomy** which enables users to tag URL bookmark and journal articles. This dataset contains 357 users, 1.783 resources and 1,573 tags.
- **Amazon** includes 4817 users, 5801 resources (products on the Amazon.com web site) and 3201 tags. (Note that this is a subset of the full network from [7].)
- **LastFM** users have music profiles and can create playlists. User may tag songs, albums and artists. This dataset contains 2,368 users, 2,350 resources and 1,141 tags.

5. METHODOLOGY

For each data set, we divided the data randomly into five partitions each having equal numbers of annotations. The first partition is used to learn the α weights for each component. The other partitions are used for cross validation: three partitions are used as training data and the fourth is used to test the system's predictions.

5.1 Weight Learning

The α values in Equation 1 are learned empirically from the first data partition. We choose a random set of α values and calculate the F-measure for the hybrid using these weights. Then we adjust one weight at random and re-compute the F-measure over the same data partition. If it increases, the change is accepted and the process repeats. Otherwise, the change is rejected and another random modification is proposed. When the values stabilize, another random starting point is chosen. The weights leading to the highest F-measure are then chosen for the rest of the experiment and the data is discarded.

5.2 Evaluation

To measure the quality of recommendations, the remaining partitions are used for four-fold cross validation. For each user in the test partition, we calculate recommendation lists Rec_u of size 1 through 10 and compare these results with the resources R_u tagged by that user in the test partition, calculating precision and recall as below.

$$recall = \frac{|R_u \cap Rec_u|}{|R_u|} \tag{2}$$

$$precision = \frac{|R_u \cap Rec_u|}{|Rec_u|} \tag{3}$$

The recall and precision are calculated for each user and averaged across all users, and then averaged across the four folds.

We evaluated two recommendation hybrids: the original hybrid (labeled "H") in the figures containing the 6 components used in [7], and an enhanced hybrid (labeled "HM") adding the three components using extended metapaths.

6. RESULTS

In all experiments we see that hybrid model that includes components with longer metapaths offers improved results. Figure 5a shows the results for Amazon dataset. This is one of the most difficult recommendation data sets in our evaluations: note the extremely low recall and precision results. We can see the HM model offers better results than the original hybrid method.

Looking at the performance of the individual components, we see that the components with longer metapaths are not better than their shorter-path counterparts. kNN_{UR} is the top individual component but kNN_{UTR} is a distant third. Cosine-M and kNN_{URT} appear to be about the same as the non-extended versions.

Figure 5b shows the results for the Bibsonomy dataset. The metapath enhanced HM hybrid shows a slight improvement over original hybrid model for this dataset. Most interesting are the characteristics of the URT and UTR components. Unlike the Amazon results, the kNN_{URT} component offers higher precision across almost the whole range of recall values and the kNN_{UTR} component is roughly comparable to the kNN_{UR} one, with slightly higher precision at lower recall.

Finally, the Last.fm results are shown in Figure 5c. One well-known characteristic of this data set is the high degree of noise associated with the tags. Users often apply vague tags such as "rock" or (worse) "favorite song" to music tracks on this site. Again the HM hybrid is superior. The UT and URT components are both equally bad – not surprising given the characteristics of the tag dimension. The UR and UTR components are quite similar in performance, which is rather surprising, given that the UTR metapath makes use of these same problematic tag links.

Figure 6 shows the learned α weights for the components in the recommendation experiments. In each graph, the grey bar represents the weights of the components in the original hybrid and the striped bars are the weights for the HM algorithm. The first set of results in Figure 6a are for the Amazon.com data. In this data, the Cosine component is a strong contributor to the hybrid. When we add the longer metapaths Cosine-M also proves useful. kNN_{UR} also has relatively high weight in the original version. The extended version of this component via the UTR metapath gets some weight in the HM model. Most surprisingly, UT, which has very low weight in the original model, becomes the single most heavily-weighted component in the HM model. This is the only data set where the simple popularity recommender has a relatively large weight.

Figure 6b shows the weight contribution of recommender components learned for the Bibsonomy dataset. The original hybrid showed a very strong contribution from the kNN_{UR} component, which in the HM model is dispersed to the longer metapath components kNN_{URT} and kNN_{UTR}. Interestingly, the contribution of the Cosine component drops by more than 50%, while the extended version Cosine-M barely makes a showing. It seems that perhaps the components with longer metapaths are doing a better job of capturing the tag-based connections between users and resources, rendering the Cosine component less valuable. We will need to do additional experimentation to characterize this phenomenon.

The learned weights for the two hybrids on the Last.fm data appear in Figure 6c. kNN_{UTR} and kNN_{UR} have similar weights in the larger hybrid. Again, the Cosine metric loses weight in the HM model, as the Cosine-M gains comparable weight.

7. FUTURE WORK

The work described here is very preliminary. While our aim is to explore recommendation is data sets with complex network schemas, to date we have only extended our prior work on social tagging systems. The results are very encouraging, however, showing improvement on the baseline system using a few components based on deeper paths into the same network. We expect that the incorporation of additional object types and relations will yield even more improvement in recommendation accuracy.

One important question is whether the hybrid weights can be predicted or at least estimated from the characteristics of the data. This issue takes on greater urgency when we consider the fact that the set of metapath-based components is unbounded – it is always possible to consider more friends of friends or to follow a link back and forth again: UTR, UTTR, UTTTR, etc. Intuitively, the value of longer paths should be less in the limit – eventually there will be a fixed point. But, as the results here show, under some conditions components built with longer metapaths are actually more successful than those with shorter paths, so we cannot assume that weights will decrease monotonically with path length.

We are experimenting with entropy-based measures of the contribution of each component, with the aim of finding a metric with which to discriminate between components and filter out those unlikely to be useful, prior to the weight learning step. Limiting the number of components is key to making weight learning efficient. In the experiments reported here, we found that adding three more components (50% increase) quadrupled the amount of training time required to learn the α values. So it is not possible add components indiscriminately. In addition, a weight estimator might be useful for providing an initial seed for the hill-climbing step. All of these questions will be explored in future work.

8. CONCLUSION

One of the key challenges in social web recommendation is the effective integration of the many dimensions of the available data. In this paper, we describe a linear-weighted hybrid approach that generalizes our prior work on social tagging systems to a larger space of heterogeneous networks. We show that this more general approach can offer performance improvement in social tagging data, and offers significant potential for incorporating additional data dimensions.

9. REFERENCES

[1] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[2] J. Chen, G. Chen, H. L. Zhang, J. Huang, and G. Zhao. Social recommendation based on multi-relational analysis. In *International Conference on Web Intelligence and Intelligent Agent Technology*, pages 471–477, 2012.

[3] S. Doerfel, R. Jäschke, A. Hotho, and G. Stumme. Leveraging publication metadata and social data into folkrank for scientific publication recommendation. In *Proceedings of the 4th ACM RecSys workshop on Recommender systems and the social web*, RSWeb '12, pages 9–16, New York, NY, USA, 2012. ACM.

[4] F. A. Durão and P. Dolog. A personalized tag-based recommendation in social web systems. In *Proceedings of International Workshop on Adaptation and Personalization for Web 2.0*, 2009.

[5] J. Gemmell, M. Ramezani, T. Schimoler, L. Christiansen, and B. Mobasher. A fast effective multi-channeled tag recommender. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases Discovery Challenge*, pages 59–63, Bled, Slovenia, 2009.

[6] J. Gemmell, T. Schimoler, B. Mobasher, and R. Burke. Hybrid tag recommendation for social annotation systems. In *19th ACM International Conference on Information and Knowledge Management*, pages 829–838, Toronto, Canada, 2010.

[7] J. Gemmell, T. Schimoler, B. Mobasher, and R. Burke. Resource recommendation in social annotation systems: A linear-weighted hybrid approach. *Journal of Computer and System Sciences*, 78(4):1160–1174, 2012.

[8] J. Han. Mining heterogeneous information networks by exploring the power of links. In J. Gama, V. Costa, A. Jorge, and P. Brazdil, editors, *Discovery Science*, volume 5808 of *Lecture Notes in Computer Science*, pages 13–30. Springer Berlin Heidelberg, 2009.

[9] P. Kazienko, K. Musial, and T. Kajdanowicz. Multidimensional social network in the social recommender system. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(4):746–759, 2011.

[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*, SIGIR '09, pages 195–202, New York, NY, USA, 2009. ACM.

[11] D. Liben-Nowell and J. M. Kleinberg. The link prediction problem for social networks. In *12th ACM International Conference on Information and Knowledge Management*, pages 556–559, 2003.

[12] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-Based Collaborative Filtering Recommendation Algorithms. In *10th International Conference on World Wide Web*, Hong Kong, China, 2001.

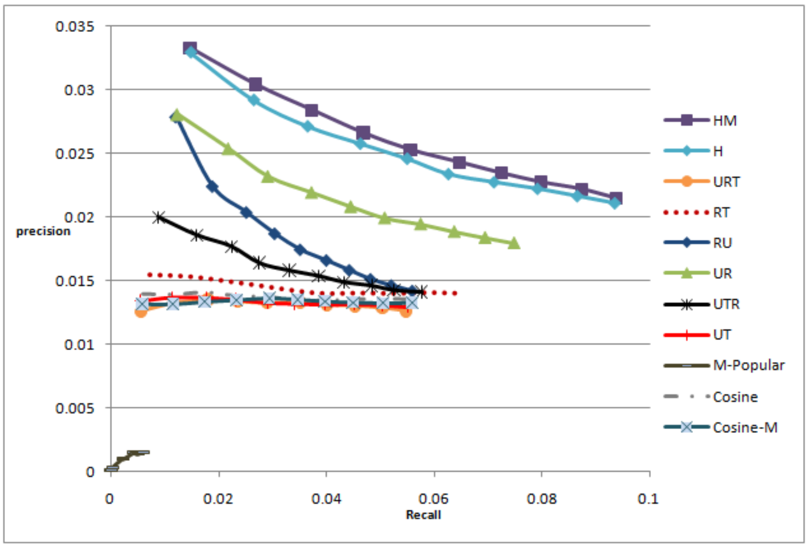
[13] S. Siersdorfer and S. Sizov. Social recommender systems for web 2.0 folksonomies. In *Hypertext*, pages 261–270, 2009.

[14] Y. Song, L. Zhang, and C. L. Giles. Automatic tag recommendation algorithms for social recommender systems. *ACM Transactions on the Web*, 5(1):4, 2011.

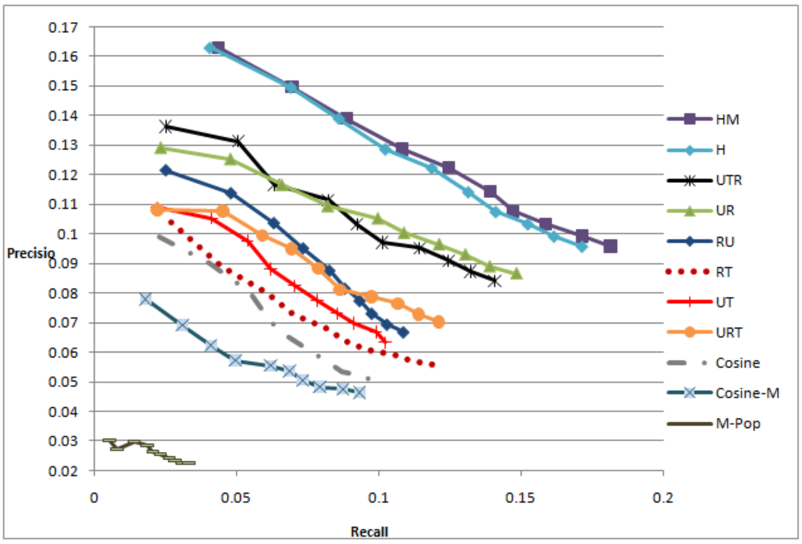
[15] Y. Sun and J. Han. *Mining Heterogeneous Information Networks: Principles and Methodologies*. Synthesis Lectures on Data Mining and Knowledge Discovery. Morgan & Claypool Publishers, 2012.

[16] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *Proceedings of the 37th International Conference on Very Large Databases*, pages 992–1003, 2011.

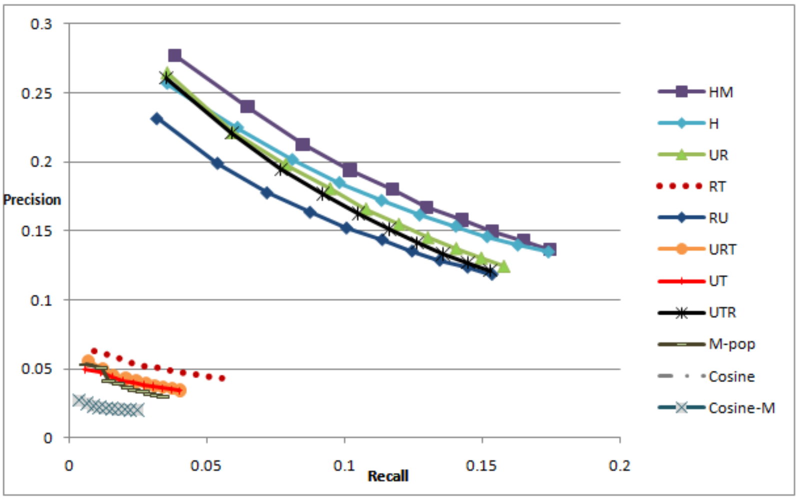
[17] S. J. Yu. The dynamic competitive recommendation algorithm in social network services. *Inf. Sci.*, 187:1–14, 2012.



(a) Amazon dataset

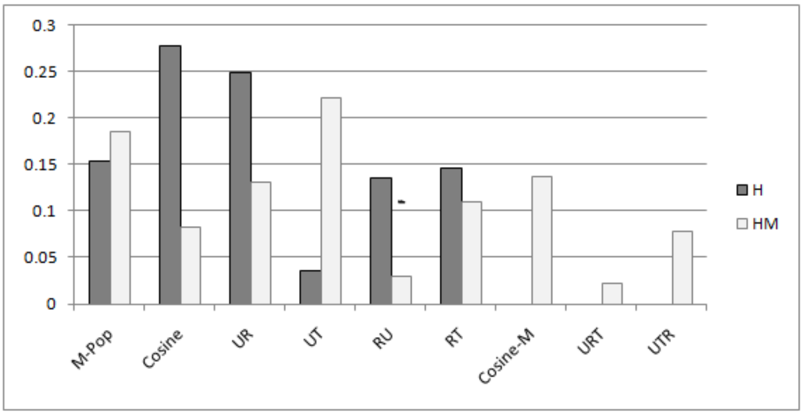


(b) Bibsonomy dataset

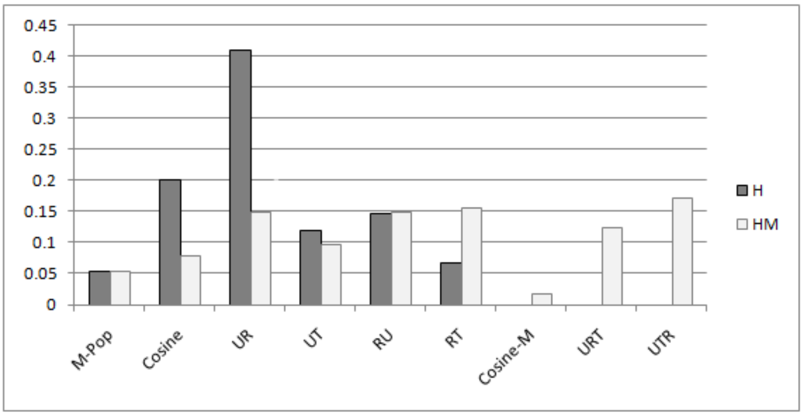


(c) Last.fm dataset

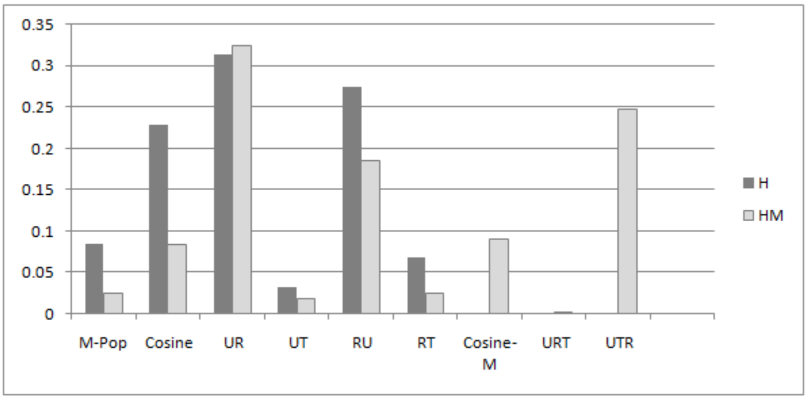
Figure 5: Resource recommendation results



(a) Amazon dataset



(b) Bibsonomy dataset



(c) LastFM dataset

Figure 6: Hybrid Weights