Hybrid Recommendation in Heterogeneous Networks

Robin Burke, Fatemeh Vahedian, and Bamshad Mobasher

Center for Web Intelligence, DePaul University, Chicago, IL 60604
{rburke,fvahedia,mobasher}@cs.depaul.edu

Abstract. The social web is characterized by a wide variety of connections between individuals and entities. A challenge for recommendation is to represent and synthesize all useful aspects of a user's profile. 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 present a general approach to recommendation in heterogeneous networks that can incorporate multiple relations in a weighted hybrid. A key feature of this approach is the use of the *metapath*, an abstraction of a class of paths in a network in which edges of different types are traversed in a particular order. A user profile is therefore a composite of multiple metapath relations. Compared to prior work with shorter metapaths, we show that a hybrid composed of components using longer metapaths yields improvements in recommendation diversity without loss of accuracy on social tagging datasets.

1 Introduction

The social web is characterized by a diversity of data types and relations. For example, the music-oriented website Last.fm contains information about artists, groups, songs, albums, playlists, and users, and connections can be drawn among any of these entities. There are also tags and other descriptive content. Diversity of information means that there are many kinds of recommendation that can be made to users: other users with whom to connect, artists to listen to, new songs for existing playlists, etc. At the same time, the complexity of the data means that there are many more types of information that can be integrated into user models for recommendation: should the system recommend songs from your friends' playlists or new music that your friends might not know yet? Often building recommenders for such sites involves devising individual ad-hoc user models for each recommendation problem.

To illustrate this type of recommendation, 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 might discriminate among these tracks using data from the Last.fm social network, as summarized in the schema in Figure 1.

V. Dimitrova et al. (Eds.): UMAP 2014, LNCS 8538, pp. 49–60, 2014.

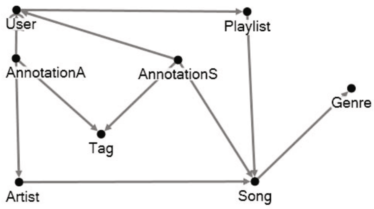

Fig. 1. Network schema for Last.fm

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 (AnnotationA). 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 tagging systems [14]; 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 users, 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 users represented by tags, it is quite straightforward to apply standard collaborative recommendation methodology:

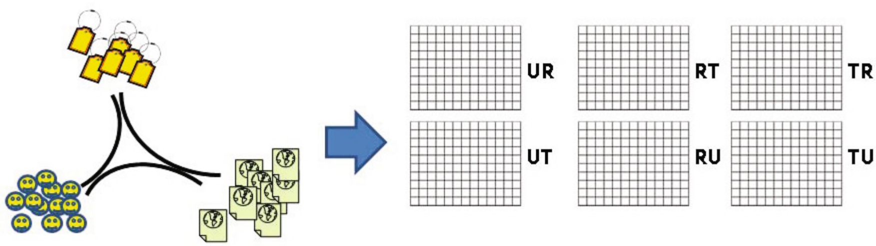

Fig. 2. Two-dimensional projections for a social tagging network

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 built of such components is more flexible and more accurate than integrative techniques such as matrix factorization that attempt to model all of the dimensions at once [14].

We extend this idea to more complex networks through the concept of the *metapath* [31]. 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 in a heterogeneous network into a sequence of edge types. Navigating a metapath from a node collects all destination nodes reachable by following edges of 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 for recommendation. 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 $\langle user \rightarrow user \rightarrow playlist \rightarrow song\rangle$. One of the key aims of this work is to investigate the value of using longer metapaths to build recommendation components.

2 Related Work

The integration of social network data into recommender systems has been studied extensively in recent years [11, 29, 30, 34]. Much of this work has been focused on system-specific solutions. For example, [20] shows a LastFM music recommender based on the combination of social data and annotations. A similar system incorporating social data and tags has been used to recommend publications in the Bibsonomy dataset [10]. In [26], Mihalkova et al. demonstrate a domain-specific approach for recommending collaborations on Wikipedia based on user-centered subgraphs. Hong et al. report on a domain-specific approach to recommending social streams in the social networking site LinkedIn [17]. In addition to click-through data, Hong's system uses features of users such as seniority of job title, as well as network-oriented features like PageRank. A more general technique is the multi-relational approach of [7] 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 [19] take a similar approach, treating the different kinds of relations in Flickr as "layers."

Recommendation in information networks is often equated with link prediction [3, 21]. Link prediction is the task of identifying missing, unobserved or yet-to-be-made connections in a network. 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. A variety of unsupervised techniques have been developed, including approaches based on graph metrics [1, 23], and random walks [18, 33]. Supervised methods for link prediction are gaining importance as well. See, for example, [24]. As discussed above, these approaches assume a homogenous network.

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 [15], 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 [32]. Sun and Han [31] argue that information propagation across heterogeneous nodes and links can be very different from that across homogeneous nodes and links.

Some researchers have examined link prediction in heterogeneous networks. Cai et al. [6] examined link prediction in two-mode social networks with reciprocation – where a tie must to reciprocated in order to be created. Although this is a very special network type, their approach using multiple collaborative recommendation components has some similarity to the hybrid that we propose in general form here. In [8, 9], Davis et al. propose a method to predict the location and type of new edges in multi-relational networks. They build a set of homogenous projections of the network and then use supervised learning with feature extraction to build a set of individual predictors in a weighted combination. More recently, Yu and colleagues [35] have proposed a algorithm that predicts user associations with items using metapath-based user clustering.

Link prediction is obviously an important problem both for homogeneous and heterogeneous networks, with many valuable applications. However, there are a number of reasons why the conflation of link prediction with recommendation is problematic. First, link prediction is undertaken with a global view of the network. In typical link prediction experiments, links are deleted from the network (either randomly or based on time intervals) and the task is to see if these links can be predicted by an algorithm [22, 24]. Recommendation, on the other hand, is inherently personalized. A recommendation is made for a particular user, and must be generated with a user profile in mind, and judged by how well it satisfies that user's needs.

The accuracy of recommendations is certainly important, but there are other metrics that have been identified as useful for evaluating recommender systems. See [25] for a comprehensive discussion. It can be important to measure to what extent a recommender is capable of producing diverse and even surprising results [2, 36]. Also, in many applications, it is important that recommendations are transparent: that the recommendations can be explained in way that users find comprehensible [16]. These types of considerations have not yet found a place in link prediction research.

As discussed above, the work reported here is an extension of research applying linear weighted hybrids to recommendation problems in social tagging systems. Our prior 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. Our results showed that it was at least as effective as other, more computationally-sophisticated techniques for the well-studied problem of tag recommendation, such as PITF [27], with the added advantages that it could be applied to a wider variety of recommendation problems and could be more easily updated. See [12–14] for more detail on this line of research.

3 Weighted Hybrid

A weighted hybrid recommender is a system comprised of multiple recommendation components, each of which returns a real-valued score for a combination of user and item. The scores from all the components are combined in a weighted sum [4]. 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, 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 tag disliked items with deprecatory tags, for example.

In the experiments reported in [14], the system (labeled H in our experiments) 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 generate 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. The matrix is 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 [28]. 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.

3.1 Metapath-Based Recommendation Components

Following Sun and Han [31], 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.

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.

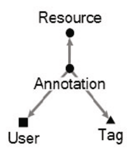

Fig. 3. Network schema for Social Tagging Sytems

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.

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. In our preliminary investigation found in [5], we opted to explore only a few possible components using short metapaths. These components together with the six from [14] make up the hybrid labeled H-M1 in the experiments that follow:

- 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*.

These components represent a one-step expansion of the UR and UT paths by incorporating the third link type. To investigate the value of using longer metapaths and of incorporating item-based approaches, we created four additional components for the hybrid H-M2. Two are additional expansions along the UR and UT dimensions, and two are item-based components analogous to kNN_{RU} and kNN_{RT} but with longer paths:

- User-based kNN with the user-tag matrix formed by following the URTRT metapath: kNN_{URTRT}.
- User-based kNN with the user-resource matrix formed by following the UTRTR metapath: kNN_{UTRTR}.
- Item-based kNN, with resource-tag matrix formed by following RUT metapath(kNN_{RUT}).
- Item-based kNN, resource-user matrix formed by following RTU metapath (kNN_{RTU}).

4 Experiments

For the experiments reported here, we used the **Bibsonomy** dataset, containing 357 users, 1.783 resources and 1,573 tags.[1] The data was filtered as described in [14] to eliminate rare and idiosyncratic tags and resources. 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.

[1] We performed similar experiments on the MovieLens dataset but do not report the results here for reasons of space.

4.1 Methodology

The α values for the hybrid are learned empirically from the first data partition using random-restart hill climbing with the overall precision of the hybrid as the optimization measure. After 5,000 iterations, the weights leading to the highest recall at 10 items are then chosen for the rest of the experiment and that fold of the data is discarded.[2]

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 of size 1 through 10 and compare these results with the held-out resources tagged by that user, calculating precision and recall for each user and averaging across all users, averaging across the four folds. Then we perform weight learning with a different partition and compute another average result, continuing and averaging across all five possible choices of the first data partition.

We also evaluated the diversity of the recommendations returned. For this calculation, we perform a pairwise similarity comparison of the top 10 results. Since we are recommending resources, we can calculate similarity in two ways: using the set of users who have tagged the resource or using the set of tags that have been applied to it. In each case, we compare using cosine similarity. In the experiments below, we report results for both types of diversity. An average dissimilarity between all pairs of items recommended to a user can be calculated as: $K \sum_{i_k \in R, l<k} d(i_k, i_l)$, where $d(i_k, i_l)$ refers to distance or dissimilarity between two distinct items in a recommendation list and K is a normalization constant based on the list size. This metric is calculated for each recommendation list for each user and averaged across all users.

4.2 Results

Figure 4 shows precision versus recall curves for three weighted hybrids and their subcomponents. The dashed line with square marks represents the original hybrid without extended meta-paths. The solid line with circular marks shows the results for the H-M1 hybrid; the H-M2 hybrid is also solid with asterisk marks. As we can see, the extended hybrid H-M2 has comparable or slightly poorer performance than the H-M1 version, and both improve on the original six component hybrid H, especially for shorter recommendation lists.

The figure also shows the performance of each component of the hybrids separately, omitting the non-personalized popularity-based component that has very poor performance on this data. The components of the original H hybrid also have dashed lines with square marks. The components of the H-M1 hybrid have circular marks; and the two extended components are far in the bottom left with asterisks. Each component is color-coded (and organized in the legend) by the two dimensions of the data that is associates. For example, kNN_{UR}, kNN_{UTR} and kNN_{UTRTR} are all user-based components that associate users with resources, using successively longer metapaths to do so.

Interestingly, if we rank the components by their dominance across the precision-recall space, we find that the extended components have exactly the opposite rank as the

[2] We are currently experimenting with particle swarm optimization algorithm as a more efficient optimization approach.

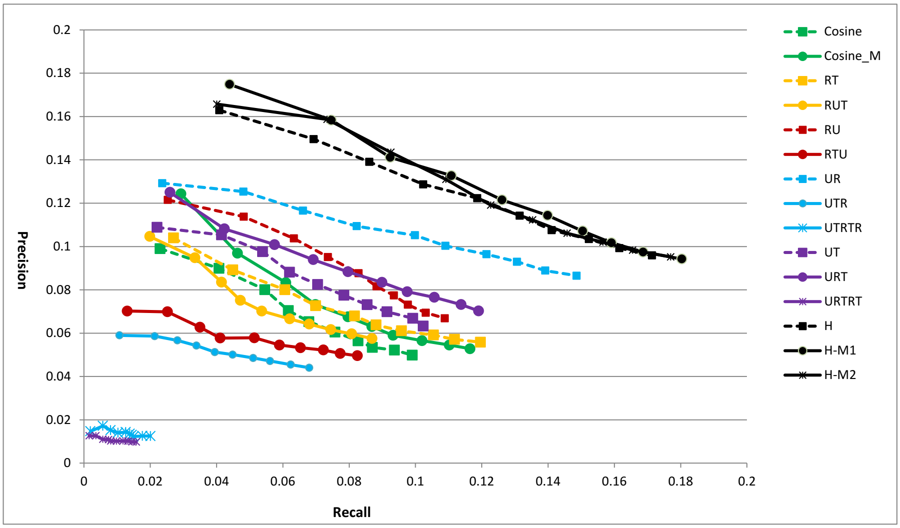

Fig. 4. Precision vs Recall for Bibsonomy Dataset

ones from the original hybrid. For example, kNN_{UR} is the best single step component for this task (not surprising because we are recommending resources to users), but kNN_{UTR} is the worst of the two-step components.

In the middle of the pack, there are the cosine and user-tag measures. Here the extended component actually has better precision-recall performance than the one based on shorter paths. We explain this phenomenon by considering the way that Bibsonomy data is generated. Users of the Bibsonomy system tend to tag articles with descriptive tags related to their reseach area. So, an article on population biology might be tagged by one user with labels having to do with its methodology and another user having to do with the specific species studied. In this scenario, it makes sense that a user's profile based on tags they have provided might not match a relevant article's tags because those tags might be have supplied by users with a different set of interests. However, the extended metapath creates a user profile based on all the tags that any user has given to the articles the user has tagged. There is a crowd-sourcing effect here so that the resources are better described by the union of all of their tags and personalization comes in the selection of resources rather than in the selection of tags. This pattern was also found in the MovieLens dataset, but we expect that other datasets with noisier tagging behavior might not exhibit it.

The diversity results are shown in Figure 5. These results are mixed and deserve greater exploration, but the key finding here is shown in the final set of results for tag-based diversity using the H-M2 hybrid. Tag diversity is probably what most users would understand as diversity: items that differ in terms of their content. So, the H-M2 hybrid is finding results of greater diversity with comparable precision and recall compared to the other hybrids. The user diversity measure provides some interesting clues about how this is achieved. User diversity goes down, suggesting that the H-M2 hybrid is

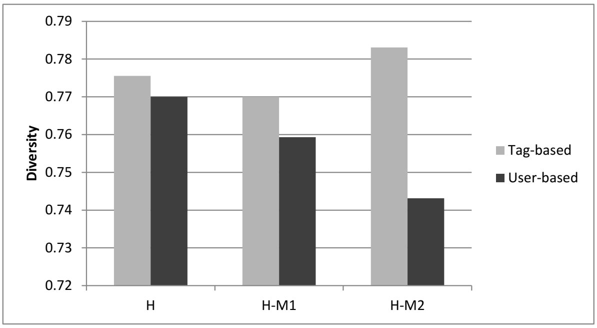

Fig. 5. Diversity Results for Bibsonomy Dataset

successful in finding the subgroup of peers whose research is of interest, even though they may not all describe that work using exactly the same tags.

5 Conclusions

One of the key challenges in social web recommendation is the effective integration of the many dimensions of the available data about users. 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. In this paper, we have shown that our metapath-based approach to recommendation in heterogeneous networks yields improvements in both accuracy and diversity in these social tagging systems. We expand on the work reported in [5] to show that hybrids with extended components achieve greater diversity without sacrificing accuracy. We view this as a proof of concept suggesting that our technique will be effective in the more general class of heterogeneous information networks.

There are a number of intriguing results. First is that there are a number of non-obvious tradeoffs in creating larger hybrids from extended network metapaths. Greater diversity would be expected, but our results show that in the Bibsonomy dataset at least, tag-based diversity goes down and then up again as more extended paths are considered. Second is that, at least in some cases, components built from longer metapaths actually perform better than the corresponding component with a shorter path: the cosine component being the example in Bibsonomy. Predictive power is therefore not a simple decreasing function of the length of the path and there are domain- and data-specific factors at play.

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, for example. 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 addition, a weight estimator might be useful for providing an initial seed for the hill-climbing step.

References

1. Adamic, L., Adar, E.: Friends and neighbors on the web. Social Networks 25(3), 211–230 (2003)
2. Adomavicius, G., Kwon, Y.: Improving aggregate recommendation diversity using ranking-based techniques. IEEE Transactions on Knowledge and Data Engineering 24(5), 896–911 (2012)
3. Benchettara, N., Kanawati, R., Rouveirol, C.: A supervised machine learning link prediction approach for academic collaboration recommendation. In: Proceedings of the fourth ACM Conference on Recommender Systems, pp. 253–256. ACM (2010)
4. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction 12(4), 331–370 (2002)
5. Burke, R., Vahedian, F.: Social web recommendation using metapaths. In: Proceedings of the Fifth ACM RecSys Workshop on Recommender Systems and the Social Web. ACM Press (2013)
6. Cai, X., Bain, M., Krzywicki, A., Wobcke, W., Kim, Y.S., Compton, P., Mahidadia, A.: Reciprocal and heterogeneous link prediction in social networks. In: Tan, P.-N., Chawla, S., Ho, C.K., Bailey, J. (eds.) PAKDD 2012, Part II. LNCS, vol. 7302, pp. 193–204. Springer, Heidelberg (2012)
7. Chen, J., Chen, G., Zhang, H.L., Huang, J., Zhao, G.: Social recommendation based on multi-relational analysis. In: International Conference on Web Intelligence and Intelligent Agent Technology, pp. 471–477 (2012)
8. Davis, D.A., Lichtenwalter, R., Chawla, N.V.: Multi-relational link prediction in heterogeneous information networks. In: ASONAM, pp. 281–288 (2011)
9. Davis, D.A., Lichtenwalter, R., Chawla, N.V.: Supervised methods for multi-relational link prediction. Social Netw. Analys. Mining 3(2), 127–141 (2013)
10. Doerfel, S., Jäschke, R., Hotho, A., Stumme, G.: 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 2012, pp. 9–16. ACM, New York (2012)
11. Durão, F.A., Dolog, P.: A personalized tag-based recommendation in social web systems. In: Proceedings of International Workshop on Adaptation and Personalization for Web 2.0 (2009)
12. Gemmell, J., Ramezani, M., Schimoler, T., Christiansen, L., Mobasher, B.: A fast effective multi-channeled tag recommender. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases Discovery Challenge, Bled, Slovenia, pp. 59–63 (2009)
13. Gemmell, J., Schimoler, T., Mobasher, B., Burke, R.: Hybrid tag recommendation for social annotation systems. In: 19th ACM International Conference on Information and Knowledge Management, Toronto, Canada, pp. 829–838 (2010)
14. Gemmell, J., Schimoler, T., Mobasher, B., Burke, R.: Resource recommendation in social annotation systems: A linear-weighted hybrid approach. Journal of Computer and System Sciences 78(4), 1160–1174 (2012)
15. Han, J.: Mining heterogeneous information networks by exploring the power of links. In: Gama, J., Costa, V.S., Jorge, A.M., Brazdil, P.B. (eds.) DS 2009. LNCS, vol. 5808, pp. 13–30. Springer, Heidelberg (2009)
16. Herlocker, J., Konstan, J., Riedl, J.: Explaining collaborative filtering recommendations. In: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, pp. 241–250. ACM (2000)

17. Hong, L., Bekkerman, R., Adler, J., Davison, B.D.: Learning to rank social update streams. In: SIGIR, pp. 651–660 (2012)
18. Jeh, G., Widom, J.: Scaling personalized web search. In: WWW, pp. 271–279 (2003)
19. Kazienko, P., Musial, K., Kajdanowicz, T.: Multidimensional social network in the social recommender system. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 41(4), 746–759 (2011)
20. Konstas, I., Stathopoulos, V., Jose, J.M.: On social networks and collaborative recommendation. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, pp. 195–202. ACM, New York (2009)
21. Kunegis, J., Lommatzsch, A.: Learning spectral graph transformations for link prediction. In: Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, pp. 561–568. ACM, New York (2009)
22. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. Journal of the American Society for Information Science and Technology 58(7), 1019–1031 (2007)
23. Liben-Nowell, D., Kleinberg, J.M.: The link prediction problem for social networks. In: 12th ACM International Conference on Information and Knowledge Management, pp. 556–559 (2003)
24. Lichtenwalter, R., Lussier, J.T., Chawla, N.V.: New perspectives and methods in link prediction. In: KDD, pp. 243–252 (2010)
25. McNee, S.: Meeting user information needs in recommender systems. PhD thesis, University of Minnesota (2006)
26. Mihalkova, L., Moustafa, W.E., Getoor, L.: Learning to predict web collaborations. In: WSDM Workshop on User Modeling for Web Applications (2011)
27. Rendle, S., Schmidt-Thieme, L.: Pairwise Interaction Tensor Factorization for Personalized Tag Recommendation. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, New York (2010)
28. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-Based Collaborative Filtering Recommendation Algorithms. In: 10th International Conference on World Wide Web, Hong Kong, China (2001)
29. Siersdorfer, S., Sizov, S.: Social recommender systems for web 2.0 folksonomies. In: Hypertext, pp. 261–270 (2009)
30. Song, Y., Zhang, L., Giles, C.L.: Automatic tag recommendation algorithms for social recommender systems. ACM Transactions on the Web 5(1), 4 (2011)
31. Sun, Y., Han, J.: Mining Heterogeneous Information Networks: Principles and Methodologies. Synthesis Lectures on Data Mining and Knowledge Discovery. Morgan & Claypool Publishers (2012)
32. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In: Proceedings of the 37th International Conference on Very Large Databases, pp. 992–1003 (2011)
33. Tong, H., Faloutsos, C., Pan, J.-Y.: Fast random walk with restart and its applications. In: ICDM, pp. 613–622 (2006)
34. Yu, S.J.: The dynamic competitive recommendation algorithm in social network services. Inf. Sci. 187, 1–14 (2012)
35. Yu, X., Ren, X., Sun, Y., Gu, Q., Sturt, B., Khandelwal, U., Norick, B., Han, J.: Personalized entity recommendation: a heterogeneous information network approach. In: WSDM, pp. 283–292 (2014)
36. Zhang, M., Hurley, N.: Avoiding monotony: improving the diversity of recommendation lists. In: Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 123–130. ACM (2008)