Analyzing Social Relations for Recommending Academic Conferences

Felix Beierle
Service-centric Networking
Technische Universität Berlin
Telekom Innovation Laboratories
Berlin, Germany
beierle@tu-berlin.de

Julian Tan
Corussoft GmbH
Berlin, Germany
julian.tan@corussoft.de

Kai Grunert
Service-centric Networking
Technische Universität Berlin
Telekom Innovation Laboratories
Berlin, Germany
kai.grunert@tu-berlin.de

ABSTRACT

Recommender systems are used to filter through vast amounts of items and recommend those that potentially have the highest relevance for the user. Recently, research dealing with recommendations in academia increased. In this paper, we analyze to what extent social relations from existing data can be utilized to generate academic conference recommendations. We design and implement a social recommender system and show how, without the need for explicit ratings, viable recommendations can be made, while at the same time reducing the cost of kNN-neighborhood selection.

CCS Concepts

•Information systems → Recommender systems; •Human-centered computing → Collaborative filtering; Social recommendation;

Keywords

Social Graphs; Academic Recommendations

1. INTRODUCTION

The proliferation of the Internet and the ever-increasing amount of online data has led to the ubiquity of recommender systems; this has also led to a growing number of research about recommenations in the field of academic user scenarios [3]. In most cases, the focus lies on recommending research papers. So far, only two studies that we are aware of tackled the problem of recommending academic conferences [8, 9]. In this paper, we will investigate, which social relations between authors are how useful when recommending conferences. With the lack of explicit ratings for academic conferences, we develop a social recommender system that uses social relations between authors to recommend the attendance at future conferences and perform a study with a much larger data set than the one used in [9].

HotPOST 16, July 04-08, 2016, Paderborn, Germany
© 2016 ACM. ISBN 978-1-4503-4344-2/16/07…$15.00
DOI: http://dx.doi.org/10.1145/2944789.2944871

We investigate to what extent already existing data about the academic community can be used to design social recommender systems. The user scenario in this paper is academic conferences, which should be recommended to particular users, based on different relations in a social graph. The usefulness of these recommendations is examined and compared to conventional recommendation approaches. The social graphs from the academic community used in this work are based on, for example, who published which work or which publication references (or is referenced by) what other papers. This leads to different possible relations between authors. For instance, if two or more authors published a joint paper, a symmetric co-authorship relation can be inferred, and when authors deal with similar topics, a co-interest relation can be assumed. Other relations can also be non-symmetric, such as citations between authors. We examine how these social relations between users can be employed in a recommendation task and show that they indeed can be used to produce quality predictions.

In the rest of this paper, we present our design and implementation of different social recommender systems and evaluate their performance. For this, we collect a data set for the user scenario of academic conferences. Evaluating our approach, we show that (1) academic social graphs can be used to derive personal preferences of authors and, (2) different social graphs might be useful for different applications, and overall, social relations are useful when building a recommender for conference recommendations. Furthermore, (3) social recommenders in the application domain of academic conferences can be a viable way to reduce the cost of traditionally expensive kNN-neighborhood selection.

2. RELATED WORK

In this section, we summarize research about recommender systems in general and social and academic recommenders in particular.

2.1 Recommender Systems

Recommender systems can help with the flood of information users get offered on the Internet. A recommender selects items from a pool of objects to recommend them to a user. In general, there are two main approaches for recommender systems. The first one is content-based filtering. Here, all items are assigned attributes which describe the item's properties. Users have attribute profiles which represent their preference for the different attributes. Predictions for an unseen item can then be made by comparing this item's attributes with the user's profile and see how well the attributes match. The problem with content-based filtering is that it is strongly dependent on well-structured attributes which are reasonably distributed across all items. Other drawbacks are that the item quality is usually not taken into account, and that serendipitous recommendations are unlikely if the static attribute structure is adhered.

The second approach is collaborative filtering. The basic concept of user-based collaborative filtering is to recommend items that similar users like, which is based on the assumption that users who agreed in the past will also agree in the future. Positive and negative preferences of a user u are stored in a vector $\vec{r_u} = (r_u(1), r_u(2), ..., r_u(n))$, in which each element $r_u(i)$ corresponds to the rating of u for item i. The length of $\vec{r_u}$ is therefore equal to $|\mathcal{I}| = n$, the total number of items in the system. $r_u(i) = 0$ indicates that user u has not yet rated item i. The first step of generating a recommendation is to find all users who have similar preferences as the target user u. One of the most widely used approaches for this task is the k-Nearest Neighbor algorithm (*kNN*). Here the k most similar users $\mathcal{U}_u = \{u_1, u_2, \dots, u_k\}$ to a target user u are selected, together with their corresponding similarity values $\mathcal{S}_u = \{s_{u_1}, s_{u_2}, \dots, s_{u_k}\}$. The users in $\mathcal{U}_u$ are termed the neighbors of u and the Cartesian product $\mathcal{N}_u = \mathcal{U}_u \times \mathcal{S}_u$ the neighborhood of u. The similarity between two users can be determined by using a similarity measure like Pearson correlation or cosine similarity. After computing the neighborhood $\mathcal{N}_u$ of user u, the next step is to compute the prediction $p_u(i)$ for the unseen item i. This is done by calculating the weighted sum of the neighbors' preferences for i [1]. Note that each neighbor n who has not rated i is automatically factored out because of the default rating of $r_n(i) = 0$ for unrated items.

The strength of this approach is that it usually recommends high quality items and that it allows for *serendipitous* recommendations without the need for any kind of item attributes. It however suffers from the *cold-start* and *sparsity* problem. Another major challenge of user-based filtering is its scalability. Since calculating the neighborhood is of quadratic complexity, this process gets computationally expensive with a growing amount of users. Better scalability can be achieved by *item-based collaborative filtering* or by using social relations for the neighbor selection. Item-based collaborative filtering works very similar to the user-based approach. The main difference is that similarities are calculated between items instead of users. So rather than looking at how similar users liked an unseen item, one first computes a neighborhood of similar items and then bases the prediction for an unseen item on how the user rated similar items.

2.2 Social Recommenders

Social recommendation is still an emerging approach to collaborative filtering. It is based on networks of social relationships which exist in various forms. A social graph consists of nodes and edges, where the nodes correspond to individuals and edges to relations between them. Most prominent examples of social graphs are online social networks such as *Facebook* where the users represent the nodes and their intermediate friendships the relations. By introducing additional node and relationship types for different objects a person can interact with, the graph can significantly gain expressiveness: In addition to direct relations between individuals, the graph can be analyzed for indirect ones. For example, the fact that two users interacted with the same item can be a valuable piece of information for a recommender system. In general, social relations can be exploited to improve recommendations, e.g., [6] and [2] both propose models of trust which make recommendations more reliable by locating more trustworthy sources through information from social networks. But apart from making recommendations more accurate, social data can help to cope with the main difficulties of collaborative filtering: *data sparsity* and the high computational complexity of neighborhood selection. In case of *data sparsity*, the social relations can be used to enrich the information domain. User interests and preferences can be inferred from social profiles and contacts, without even having users' ratings in the system. For the neighborhood creation, social relations represent a viable alternative to the conventional expensive neighbor search through comparisons of each user.

In [7], the authors developed a system based on social relations of *Flickr*-users, which makes recommendations for other users of interest. They identified different types of relations between two users: relations through common tags or groups, friend-relations (based on who is on whose contact list), favorite-relations (based on who added whose photo to their favorites), opinion-relations (based on who commented on whose photo). A user study with eight subjects showed an approval rate of roughly 50% of the recommendations, which could be slightly increased further by weighting the relations differently. Despite the low number of participants, the results reveal the expressiveness of social relations.

In [5], the data *sparsity* and *cold-start* problem is tackled with collaborative filtering. In their work, the authors construct a social network from user reviews on *Yelp* and analyze how preferences of immediate and distant friends can improve prediction accuracy for restaurant ratings. They evaluate their approach on a data set of about 4100 restaurants and 9400 users in a *10-fold cross-validation* and compare their approach with the conventional *kNN* algorithm. The results show that making use of social relations improves the general prediction accuracy by 17.8%.

2.3 Academic Recommenders

What we call academic recommenders are recommendation systems in the academic environment. Users are represented by the authors of publications and social connections stem from their professional relationships. For example the friend-relation in a non-academic network can here be substituted with the co-author-relation, which is typically a connection with a high significance. Other relations originate from common keywords, institutions, publication venues or the citation network. The recommendation items are usually scientific materials such as research papers, articles and books; however, recommending journals, conferences, other authors, or institutions is also conceivable.

When it comes to designing a recommender, it is important to understand the characteristics of the information domain. Approaches and algorithms that work well in one environment might not achieve the best results in another and certain challenges can have different impacts in different domains [4]. Most of the research in the field of academic recommenders deals with the recommendation of scientific research papers, in fact we are only aware of two publications that address the recommendation of scientific conferences [8, 9]. But taking a closer look at the whole field is still worthwhile, since the information domain is the same.

Beel et al. present an extensive survey of recommender systems for research papers, in which they review over 100 articles on this topic [3]. They find that the number of publications in this field has increased considerably over the recent years, which demonstrates the growing significance of academic recommendations. The main problem however is inconsistent evaluation practices, which make it difficult to compare and assess different approaches. Furthermore, precise recommendation algorithms are often underreported so that they can hardly be reproduced in order to act as a baseline for new approaches.

Klamma et al. construct a model of academic communities and events to utilize social network analysis and collaborative filtering in order to recommend conferences to users [8]. Similar to our approach, data is crawled from the world wide web. The key difference in our paper is that we analyze the social relations between authors to investigate to what extend each of those relations contributes to a good recommendation.

The work of Luong et al. deals with the recommendation of scientific conferences using social relations between authors [9]. They examine how social network analysis can be used to make recommendations of venues, which are most suitable to release a certain publication. They developed three approaches which all are based on what conferences where visited in the past by the publication's authors and their co-authors (neighbors). The first two are simplified non-personalized precursors of the last approach, which calculates for each neighbor a normalized sum of publications released at a certain conference. The prediction score is then computed by forming the weighted sum of these sums. Their offline evaluation shows that this approach can predict the correct conference in the first attempt over 50% of the times. Recommending up to 3 conferences raises the probability that they contain the correct one to over 90%. It is unclear though to what extend neighbors contribute to the results, since the prediction score also contains the author's own publication counts. In this paper, we focus on explicitly distinguishing between different social relations when recommending conferences. Furthermore, we use a much larger data set than [9].

3. DESIGN

In this section, we describe the design and implementation of our setup. For this, we first present the data set we obtained and then detail the implemented social recommendation processes.

3.1 Data Set

The first step in designing our social recommender approach with academic conferences is the obtainment of data. The ACM Digital Library[1] is one of the most comprehensive collections of scientific publications in this field. It provides extensive information on both authors, publications, conferences, and journals as well as relations between each other. Since there is no public API, the library has to be crawled in order to get the information. After crawling, the data needs to be pre-processed, e.g., to eliminate duplicates.

[1]http://dl.acm.org

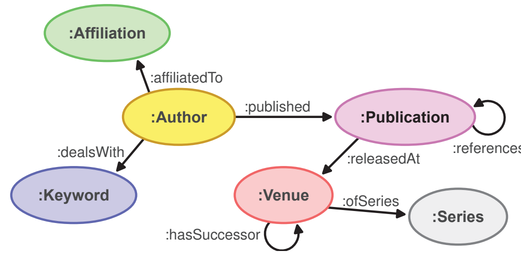

Figure 1: The different types of nodes and relations in the graph.

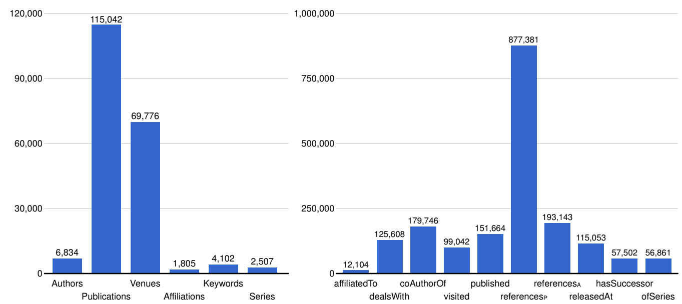

Figure 2: Total number of elements for each node and relationship type.

In a second step, the collected data has to be structured appropriately so that it can be conveniently accessed. Here, we used Neo4j[2] to store our graph we use for our social recommender system.

The structure of the data set we use in our experiments is described in Figure 1. The size of the data is indicated in Figure 2. *Venue* refers to the 'place' of publication, a conference or a journal, while *series* refers to a series of such venues. The *:coAuthorOf*- and *:visited*-relations in the right chart of Figure 2 do not appear in our data model of Figure 1. These are introduced for efficiency reasons in the implementation step and are listed here for the sake of completeness. The same applies to the $:references_A$-relation between authors. The $:references_P$-relation denotes the relation as shown above in Figure 1.

3.2 Recommendation Processes

In the recommendation process, we create different neighborhoods for the *kNN* algorithm (see Section 2). In our use case of recommending academic conferences, we have a specific time constraint: it only makes sense to recommend venues that lie in the future. In that sense, our items have an expiration date, recommending a venue in the past is of no use, since the user cannot visit it anymore.

User preference is expressed implicitly: by visiting a venue we assume a unary rating for the venue. As venues are part of series, visiting multiple venues from a series denotes higher preference.

With the available data, we can use different neighborhood selections: *co-activity* (having visited the same venue), *co-interest* (being associated with the same keywords), *co-author*, *colleague*, *reference*, and *citation*. Additionally, we implemented a *hybrid* combining the above six recom-

[2]http://neo4j.com

menders. We use cosine similarity for the similarity calculations where applicable. Utilizing the existing social relations for *kNN* neighborhood selection, the expensive process of comparing a user with all other users in order to find the most similar ones, can be reduced significantly.

In order to compare the results with recommenders that do not take social relations into account, we implemented two baseline recommenders. The first one is a random recommender that randomly picks users from the data set as neighbors. The second baseline recommender is the *self* recommender. In this one, the user herself is the only neighbor used for *kNN*, which means that the recommendations are only based on her own visited conferences.

4. EVALUATION

For the evaluation, we do an offline evaluation utilizing leave-one-out cross-validation. To get training and test sets, we split the available data into two parts. The training data consists of data up to a certain point in time, the test set contains all data from subsequent years. In the context of this evaluation, a recommendation is deemed relevant when the author visited the recommended conference in the test set. We focus on two aspects, accuracy and serendipity. In order to measure accuracy, we use the measures *precision* and *recall* which work well with binary relevance. With $\mathcal{I}_s$ being the set of all recommended items and $\mathcal{I}_r$ the set of all relevant items, *precision* and *recall* are defined as follows:

$$Precision = \frac{|\mathcal{I}_s \cap \mathcal{I}_r|}{|\mathcal{I}_s|} \quad (1) \qquad Recall = \frac{|\mathcal{I}_s \cap \mathcal{I}_r|}{|\mathcal{I}_r|} \quad (2)$$

Coverage does not look at the quality of recommendations, but rather at the percentage of items the recommender can make predictions for.

The second aspect we look at in the evaluation is *serendipity*. A *serendipitous* item is novel, the user has not seen it before. It is also non-obvious, which means that it is unlikely for the user to discover it without the help of the recommender. A system could have high accuracy and coverage and still be of no use, if everything it recommends is already known to the user. Therefore it is important that suggestions include enough items the users are not already aware of. However, presenting only new items also doesn't seem to work best either, since users like to receive some recommendations of items they are already familiar with [11]. This is because it creates trust and confidence in the quality of the recommendations.

4.1 Accuracy

For comparison of the different recommenders' accuracy levels, we generate top-N lists (indicating the number of recommendations) with every recommender and measure their precision and recall. For N we use 1, 3, 5, and 10. Figure 3 shows the resulting values. The depicted curves represent the precision and recall for the four different sizes of N. The chart moreover displays the *coverage* and the average number of neighbors for each approach.

We find some quite significant differences: Apart from the baseline approaches, the *co-activity*-recommender reaches the best accuracy values. This makes sense, since here the neighbor similarity is based on equal likings for the recommended items. Reaching a good accuracy with this approach was therefore to be expected. The *colleague*-recommender performs the worst, which suggests that this relation provides the weakest correlation to an author's interest. This can be explained by institutions often having various fields of interest: The fact that two individuals are affiliated to the same university does not necessarily reveal much about their shared interests. The results for the *reference*- and *citation*-recommenders are very similar to each other, so referencing or being referenced seems to be equally expressive. However, both relations appear to be less expressive than sharing a set of common keywords. The *co-author*-recommender produces decent accuracy results, even though its top-1 precision is half of the *co-activity*'s.

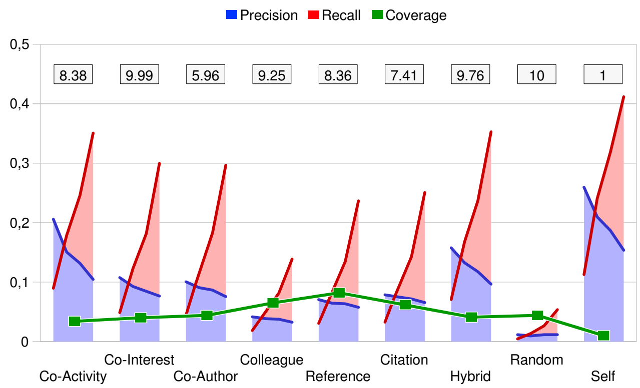

Figure 3: Accuracy and coverage of each recommender for the top-N sizes 1, 3, 5, and 10. Each curve consists of 4 data points, one for each N from left to right in ascending order. The boxes at the top contain the average number of neighbors for each approach.

Looking at the baseline approaches, we find two extremes: The *random*-recommender yields poor accuracy results since the neighborhood was constructed randomly. Accuracy-wise, the *self*-recommender outperforms all other approaches. This demonstrates that users tend to do and like things similar to their past actions and preferences. The question however is, of how much use such recommendations are, since they only consist of *non-novel* venues, i.e., venues of series the author is already aware of. Despite the high accuracy, such a recommender creates little added value, since it is essentially just good at predicting what the user is going to do anyway. This is also reflected by the low *coverage*: The *self*-recommender can only make predictions for series an author has visited before.

In general, we can observe that higher *coverage* comes with lower accuracy, with the *reference*-recommender reaching the highest value. The reason for the *citation*-recommender having a lower coverage than the *reference*-recommender is the smaller neighbor count, which in turn comes from the fact that particularly authors with few publications have more references than citations. The high amount of authors with very few publications is also the reason for the low number of average neighbors of the *co-author*-recommender. For such authors, this relation suffers the most in quantity, since having published just a couple of works often leads to just a handful of co-authors.

4.2 Serendipity

A recommendation that potentially benefits the user is one that suggests something new. The ratio of new items is

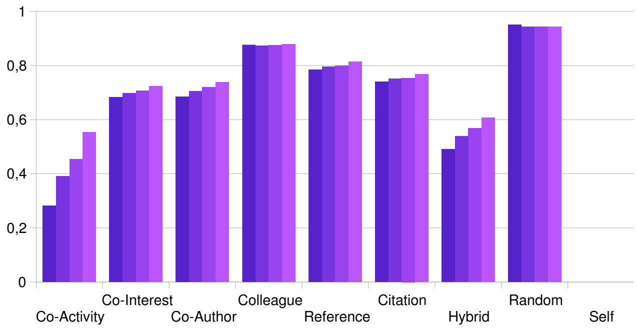

Figure 4: The ratio of serendipitous recommendations for the top-N sizes 1, 3, 5, and 10.

what we want to find out when measuring the serendipity. We measure the ratio of serendipitous recommendations, together with an estimate of how well they are received. In our test sets, about half of the visited venues are novel, while the fewer publications an author has, the more likely it is for her to visit new venues.

Figure 4 illustrates the amount of serendipitous recommendations produced by each recommender. This gives a very different picture and reveals how accuracy and serendipity are inversely correlated: The recommenders which reached the highest accuracy values make the most conservative recommendations. The *self*-recommender has a serendipity of 0, since it only suggests venues of already known series. By contrast, the *random*-recommender recommends known series only by coincidence and therefore reaches a serendipity ratio close to 1. Of the non-baseline approaches, the *colleague*-recommender generates the highest number of serendipitous recommendations.

Of course, a high serendipity does not necessarily imply better quality of recommendations. A serendipitous recommendation is only beneficial when it is of relevance to the user. We try to measure this aspect by looking at the novel venues the authors visited. That is, the relevant items are represented by the novel venues in the test set. Here, we can apply the measures of precision and recall again. The precision reveals how many of the suggested serendipitous venues were actually visited by the author, while recall specifies how many of the novel venues the author visited were suggested by the recommender. The results are displayed in Figure 5.

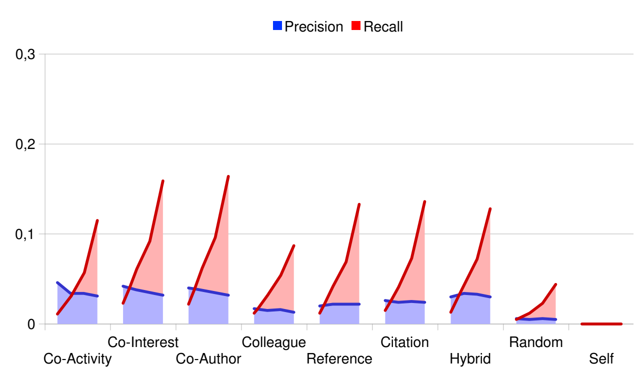

Figure 5: Accuracy of serendipitous recommendations for the top-N sizes 1, 3, 5, and 10.

Comparing the results with each other and assuming that the *random*-recommender generates no meaningful recommendations, a useful recommender should exceed the *random*-recommender's results significantly. Since the *colleague*-recommender seems to lag behind both in accuracy and in recommending unseen items, we can infer that it does not present a good approach.

4.3 Discussion

The overall results show that recommender systems can make use of data from academic social networks in a meaningful way. However, it can hardly be said which recommenders and which parameters work best, since this is strongly dependent on the application. If we just want to recommend some potentially relevant venues, a high precision is important. If we need predictions for arbitrary venues, we need a high coverage in order to be able to compute a corresponding score. A high recall is needed, if it is important to present as many relevant venues as possible. To focus on the right measures, it is important to identify what kind of recommendations benefit the users. This can depend on different contexts and use cases: One author might wish to receive more serendipitous recommendations, e.g., to find different options on where to publish a certain paper, while another one may want more conservative ones.

The different recommending approaches also differ in complexity. The *co-activity* and *co-interest*-recommenders have to compare every author with each other in order to find the best neighbors. The resulting quadratic complexity could pose a problem, if the number of users in the system is too high. In such cases, the *co-author*-recommender is very efficient, since the neighbor relations can directly be inferred and don't have to be calculated. The same applies to the *reference-*, *citation-*, and *colleague*-recommenders. Judging by the presented results, the *hybrid*-recommender seems not to provide any advantage over the other approaches. While its accuracy does not surpass the others, it can though reach the highest coverage levels, since it combines all others' neighborhoods.

We consider the overall results as satisfactory. The levels of accuracy and novelty seem adequate, also in consideration of a related study by McNee et al. [10]. Although they recommend research papers instead of venues, the user scenario of academic social relations is the same. They also use the top-N evaluation method and reach a recall of about 0.15 for $N = 1$ and about 0.5 for $N = 10$, which in average exceeds our recall by about 0.1 to 0.2. However, their test sets only contain 1 removed item for each user which has to be found to reach a recall of 1. Because of our time constraint we could not just remove one item per user from the training data, but a variable number instead. For the authors we evaluated on, this corresponds to an average of about 1.8 removed items per author. Since the number of recommended items per user is the same (N), our lists have to contain 1.8 times more of the relevant items than theirs in order to reach the same level of recall. In consideration of this, the results seem quite similar.

In their evaluation, Luong et al. only used 16 different conferences [9], which makes it difficult to judge how the algorithms would perform for a higher number. Furthermore it is unclear how the neighbors contribute to the result, since the sum in the prediction score also includes the author's own publication history. This puts a strong weight on the user's previously visited conferences, yielding recommendations with possibly little novelty. In [8], Klamma et al. did a user study to evaluate a prototype on a four point scale where users indicated their level of satisfaction with a recommendation. Given the different nature of this evaluation, a comparison is hardly possible.

With this evaluation, we have a first impression of the usefulness of applying social recommenders for academic conference recommendations. Because this is an offline evaluation, we lack user feedback. Users might have visited one of the recommended venues if they had received the recommendation. Because of that, we could argue that the recall is more expressive than precision. This is because we know that visited venues are relevant, but we cannot make any statement about venues which were not visited. They could either be relevant or irrelevant. Another related issue is that our data set only contains data about authors whose publications were accepted for a certain conference or journal. Information on whether an author applied for a conference to which he was not accepted is not available, but would be a valuable piece of information. Because of that, our venue recommendations can also be understood as recommendations for venues for which it is likely to get accepted.

5. CONCLUSION AND OUTLOOK

In this paper, we retrieved social graphs from the *ACM Digital Library* and developed seven different approaches to make recommendations for academic conferences and journals based on this data. The graph contains data about authors, their publications, affiliations, visited venues, and their intermediate relations. We identified six main ways two authors can be related: Through common publications (*co-authorship*) or affiliations (*colleagueship*), similar keywords (*co-interest*), commonly visited venues (*co-activity*), or through referencing or being referenced by the other. Based on that, we designed six recommenders and a hybrid approach and compared their results to two baseline algorithms. We used cross-validation to measure the accuracy, coverage, and serendipity. There is no single recommender that performs best for every criteria, so selecting the right approach for the right scenario is important. Besides the two baseline recommenders, there is only the *colleague*-relation which seems not to be eligible to be used in a recommendation process. Accuracy-wise the *co-authorship*, *co-interest*, and *co-activity*-relations lead to the best results. However, we discovered that they also produce the least novel recommendations, while the other relations with lower accuracy result in higher levels of both coverage and serendipity. In an attempt to measure the quality of the novel recommendations we found that all approaches, except for the *colleague*-relation, don't differ too much and could therefore prove to be equally useful in finding new content.

Overall, we showed that the data from an academic social graph can be used to derive personal preferences of authors for academic conferences and journals. Different social graphs can be useful for different application scenarios, while in general, social relations can be exploited for conference recommendations. Furthermore, we indicated that these existing relations between authors provide an alternative to the traditionally expensive neighborhood selection. This demonstrates the expressiveness of social relations between users. Future work includes applying the concept of social recommenders to other user scenarios and conducting real user studies with user feedback about the given recommendations.

6. ACKNOWLEDGMENTS

This work was done in the context of project DYNAMIC[3] (grant No 01IS12056), which is funded as part of the Software Campus initiative by the German Federal Ministry of Education and Research (BMBF).

7. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.

[2] O. Arazy, N. Kumar, and B. Shapira. Social Recommendations Systems: Leveraging the Power of Social Networks in Generating Recommendations. In *Proc. of the 1st Design Science Research in Information Systems and Technology conference (DESRIST), Feb. 2006*, Claremont, CA, USA, 2006.

[3] J. Beel, B. Gipp, S. Langer, and C. Breitinger. Research-paper recommender systems: a literature survey. *Int. J. on Digital Libraries*, pages 1–34, 2015.

[4] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan. Collaborative Filtering Recommender Systems. *Foundations and Trends in Human-Computer Interaction*, 4(2):81–173, Feb. 2011.

[5] J. He and W. W. Chu. A Social Network-Based Recommender System (SNRS). In N. Memon *et al.*, editors, *Data Mining for Social Network Data*, pages 47–74. Springer, 2010.

[6] T. Hogg and L. Adamic. Enhancing Reputation Mechanisms via Online Social Networks. In *Proc. of the 5th ACM Conference on Electronic Commerce*, EC '04, pages 236–237, New York, NY, USA, 2004. ACM.

[7] P. Kazienko, K. Musial, and T. Kajdanowicz. Multidimensional Social Network in the Social Recommender System. *Trans. Sys. Man Cyber. Part A*, 41(4):746–759, July 2011.

[8] R. Klamma, P. M. Cuong, and Y. Cao. You Never Walk Alone: Recommending Academic Events Based on Social Network Analysis. In O. Akan *et al.*, editors, *Complex Sciences*, volume 4, pages 657–670. Springer, 2009.

[9] H. Luong, T. Huynh, S. Gauch, L. Do, and K. Hoang. Publication Venue Recommendation Using Author Network's Publication History. In J.-S. Pan *et al.*, editors, *Intelligent Information and Database Systems*, pages 426–435. Springer, 2012.

[10] S. M. McNee, I. Albert, D. Cosley, P. Gopalkrishnan, S. K. Lam, A. M. Rashid, J. A. Konstan, and J. Riedl. On the Recommending of Citations for Research Papers. In *Proc. CSCW '02*, pages 116–125, New York, NY, USA, 2002. ACM.

[11] S. Sinha, K. S. Rashmi, and R. Sinha. Beyond algorithms: An HCI Perspective on Recommender Systems, 2001.

[3]http://www.dynamic-project.de

Additional Information

Bibliographic Data

F. Beierle, J. Tan, and K. Grunert, “Analyzing Social Relations for Recommending Academic Conferences,” in *Proceedings of the 8th ACM International Workshop on Hot Topics in Planet-scale Mobile Computing and Online Social Networking (HotPOST)*. ACM, 2016, p. 37–42.

Pre-print from

https://beierle.de

Online at

http://dx.doi.org/10.1145/2944789.2944871

Authors

Felix Beierle

Julian Tan

Kai Grunert

BibTeX

@inproceedings{Beierle2016HotPOST,
title = {{Analyzing Social Relations for Recommending Academic Conferences}},
author = {Beierle, Felix and Tan, Julian and Grunert, Kai},
booktitle = {{Proceedings of the 8th ACM International Workshop on Hot Topics in Planet-scale Mobile Computing and Online Social Networking (HotPOST)}},
publisher = {ACM},
year = {2016},
pages = {37—42},
doi = {10.1145/2944789.2944871}
}

Copyright Note