The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Explainable Cross-Domain Recommendations through Relational Learning

Sirawit Sopchoke
Graduate School of Information
Science and Technology,
Osaka University, Suita, Osaka 565-0871, Japan
sirawit@ai.sanken.osaka-u.ac.jp

Ken-ichi Fukui, Masayuki Numao
Institute of Scientific and Industrial Research,
Osaka University, Ibaraki,
Osaka 567-0047, Japan
{surname}@ai.sanken.osaka-u.ac.jp

Abstract

We propose a method to generate explainable recommendation rules on cross-domain problems. Our two main contributions are: i) using relational learning to generate the rules which are able to explain clearly why the items were recommended to the particular user, ii) using the user's preferences of items on different domains and item attributes to generate novel or unexpected recommendations for the user. To illustrate that our method is indeed feasible and applicable, we conducted experiments on music and movie domains.

Recommender systems (RS) currently become one of the most basic supportive techniques in an online landscape/world. RS has proven to be a major source of enhanced functionality, user satisfaction, and revenue improvement. The most common critical issues found with RS include maximizing prediction accuracy, solving cold-start problem, reducing sparsity, providing novelty, diversity and serendipity. However, solving one problem may create another problem, or a trade-off. All issues have not been perfectly solved since many current recommender algorithms seem to be locked away inside a black box. Once an algorithm is processed, it is quite difficult to understand why it gives a particular recommendation to a set of data inputs. If we can understand the reason behind the recommendation, we believe we will be able to possibly find the way to handle such problems more effectively.

Cross-domain approach improves prediction accuracy by reducing data sparsity and offering added values to recommendations by providing diversity, novelty and serendipity predictions (Cantador et al. 2015). In cross-domain recommendation tasks, the systems recommend items in the target domain to users in the source domain. There are two types of cross-domain approach: Aggregating and Transferring and they are different mainly based on how knowledge from the source domain is exploited.

Relational learning has already shown its use in RS. The evidences from the researches by Kouki et al. (2015) and Catherine and Cohen (2016) indicate that relational learning provides a better recommendation performance by incorporating additional information compared to traditional methods with a single dyadic relationship between the objects, i.e. users and items. Hence, the relational learning captured our interest to model and provide a potential solution for explainable cross-domain recommendations.

We propose a method to generate explainable recommendation rules on cross-domain problems using relational learning. The generated rules explain why the system gives a particular recommendation to a user. The rules are simple and understandable. Moreover, some novel recommendations in the primary domain are generated based on the user's preference on additional domains.

Research Methodology

We first investigated how a user's taste of another type-level item (e.g., movie, book, game) is used to predict the user's music taste, for example, could a user's movie taste be used to predict his music taste? and how?

Our initial experiments were mainly conducted over two datasets, user's music and movie preferences and item attributes, obtained from Amazon product dataset provided by UCSD.[1] The user preference dataset included reviews, product metadata and links. The item attribute dataset contained attributes of music and movies listed in the first dataset. This dataset was developed using Amazon Product API.

Our two proposed methods used to generate the music recommendation rules are described below.

Method1: Recommendations on cross domains using items. This method was designed based on traditional recommendation approach. Pearson correlation was used to find similarities between users, then deduce the certain recommendation rules. For the user-based recommendation, the similar taste on movies would lead to a similar taste on music. For the item-based recommendation, the high confidence of the defined rule would result in a music recommendation. The ProbLog[2] syntax was used to describe the model. The approach was performed in three steps: define rule, query and select output. Only output greater than or equal to the desired threshold would be selected, some examples are shown below.

```
Example: User-based recommendation
%"Music" will be recommended to user2 if user1 likes
%that "Music" and his movie preference influences user2
Defined rule: likedMusic(U2,Music)
:- influencesUser(U1,U2), likedMusic(U1,Music).
Query: query(likedMusic(U2,Music)).
```

[1]http://jmcauley.ucsd.edu/data/amazon/
[2]https://dtai.cs.kuleuven.be/problog/

```
Example: Item-based recommendation
%"Music" will be recommended to user2 if user1 likes
%that "Music", and both of them like the same "Movie"
%and "Movie" influences "Music"
Defined rule: likedMusic(U2,Music)
:- influencesItem(Movie,Music), likedMovie(U2,Movie),
likedMovie(U1,Movie), likedMusic(U1,Music).
Query: query(likedMusic(U2,Music)).
```

Method2: Recommendations on cross domains using items plus their attributes. In this method, the attributes of the items in the domains should also be taken into account to generate unexpected and useful recommendations resulting in wider and better set of solutions. To generate the music recommendation, the processing steps were designed as follows: Step 1 Sample Set Preparation: In this step, more information (attributes of music and movie) is required, e.g., songA is a type of country, songB sung by artist1, movieC is a type of comedy. Step 2 Recommendation: Learning step using ProbFOIL, a probabilistic extension of FOIL that is capable of learning probabilistic rules from probabilistic data.[3] Some examples of the output are shown below.

```
%People who like type1 movie, also like type2 music
P::likedMusic(User,Music) :- likedMovie(User,Movie),
musicType(Music,country), movieType(Movie,western).

%People who like the same type of movie,
%also like the same music
P::likedMusic(U2,Music) :- likedMovie(U2,Movie1),
likedMovie(U1,Movie2), movieType(Movie1,thriller),
movieType(Movie2,thriller), likedMusic(U1,Music).

%People who like two different types of movie,
%also like theme1 music
P::likedMusic(User,Music) :- likedMovie(User,Movie1),
likedMovie(User,Movie2), movieType(Movie1,drama),
movieType(Movie2,comedy), musicTheme(Music,lovesong).
```

Results and Discussion

The recommendation rules generated by Method2 provide explanation facilities for our RS. They state clearly why the items were recommended to the particular user. The first rule, for example, the country music was recommended to the user based on his preference on western (cowboy) movie.

For evaluation metrics and data partitioning, the evaluation was performed by precision evaluation and rank-based evaluation metrics as per the traditional evaluation method for RS. Partitioning technique was hold-out setting. In this setting, test reviews were sampled and hidden from our dataset without partitioning the users.

Our experiment shows 60% accuracy for the recommendations from Method1 and 73% from Method2. While the accuracy was moderate, our methods produced recommendations not found in the single-domain based system and not expected by users but they (users) may find them interesting (Figure 1). The usefulness of the unexpected recommendations measured with SRDP (Ge, Delgado-Battenfeld, and Jannach 2010) is 21%. It is promising and encouraging to generate the novel recommendation music based on user's preference on movies. Our result can be merged with the result from the traditional method on users' music preference only (single domain) to generate a better recommendation. Our generated recommendation rules are understandable and effective. With such explanation facilities, we believe that we can and will possibly find the way to solve the problems commonly found in the RS.

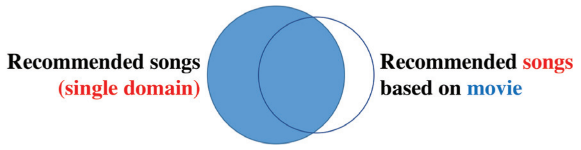

Figure 1: The results contain unexpected and useful recommendations - not found in single-domain based system

Our method is also extendible since the system does not limit to only two domains and can be considered as multi-domain RS. Additional domains, new data (e.g., features), and context (e.g., location, time, and mood) can be incorporated. Furthermore, the proposed method provides the set of solutions which are broader and better because serendipity is included.

However, additional domain, data, or context may create a scalability challenge. Nonetheless, it will be traded off with a wider recommendation solution. A further study should be conducted to address the scalability issue.

Acknowledgments

This work was supported by JSPS Core-to-Core Program, A. Advanced Research Networks. In particular, we would like to express our deepest thanks to Professor Dr. Luc De Raedt for his invaluable guidance and advice and to Dr. Anton Dries for his support throughout the program.

References

Cantador, I.; Fernández-Tobías, I.; Berkovsky, S.; and Cremonesi, P. 2015. *Cross-Domain Recommender Systems*. Boston, MA: Springer US. 919–959.

Catherine, R., and Cohen, W. 2016. Personalized recommendations using knowledge graphs: A probabilistic logic programming approach. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, 325–332. New York, NY, USA: ACM.

Ge, M.; Delgado-Battenfeld, C.; and Jannach, D. 2010. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, 257–260. ACM.

Kouki, P.; Fakhraei, S.; Foulds, J.; Eirinaki, M.; and Getoor, L. 2015. Hyper: A flexible and extensible probabilistic framework for hybrid recommender systems. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, 99–106. ACM.

[3]https://dtai.cs.kuleuven.be/software/probfoil/