RecSPARQL for Cross-Domain Recommendations

Victor Anthony Arrascue Ayala and Georg Lausen

University of Freiburg, Georges-Köhler Allee, Geb. 51, 79110 Freiburg, Germany
{arrascue,lausen}@informatik.uni-freiburg.de
http://dbis.informatik.uni-freiburg.de

Abstract. Recommender Systems (RS) benefit from the richly-structured information contained in publicly available RDF-graphs. Not only is the burden of extracting features from text thereby alleviated, but also the interconnections in the graph have been proven to be useful. These allow one to find related items in the graph which do not necessarily share a large number of features, e.g. items from different domains. Thus, these graphs can be exploited to generate cross-domain recommendations, i.e. to recommend items using feedback provided in a different domain. To benefit from RDF's data model, RecSPARQL has been proposed as an extension of SPARQL together with a system which evaluates such queries. Although this solution makes it possible to generate recommendations on top of arbitrary RDF-graphs, it is limited to single-domain recommendations. In this paper we present an extension of RecSPARQL's syntax and semantics for cross-domain recommendations. Our experiments on a very sparse dataset show that the added components can help to improve the quality of recommendations.

1 Introduction and Background

Driven by the Semantic Web (SW) initiative, a considerable number of knowledge graphs are published according to the Resource Description Framework (RDF). For Recommender Systems (RS) [5], i.e. systems which recommend to users new items they might be interested in, exploiting such structured knowledge has proven to have the potential to enhance the quality of recommendations [4]. In particular, the interconnections in the graph are very useful to navigate to other similar or related items [6], especially those which belong to different domains. For instance, the movie and music domains can be interconnected by paths consisting of *soundtrack* and *related artists* predicates. Therefore, semantic graphs, such as those published according to Linked Open Data (LoD) principles, have been widely investigated for this purpose due to their characteristic of linking domains [3]. Recommender Systems which exploit preferences for items that belong to a certain domain (source domain) to generate recommendations of items from a different domain (target domain) are called cross-domain [2]. They are based on the assumption that there are correspondences between user and item preferences in both domains. The typical goal of considering an additional domain is to overcome the lack of ratings in a single-domain scenario. Even for the simpler single-domain case, a RS cannot easily leverage both feature nodes and structure from an RDF-graph, since a mismatch exists between both data models. To tackle this, RecSPARQL was proposed in [1] as an extension to SPARQL. RecSPARQL queries empower users to generate recommendations from arbitrary RDF-graphs by allowing them to specify features in the query which are used to build a recommendation model. However, the authors limited this work to single-domain recommendations. In this paper we present an extension of RecSPARQL's syntax and semantics for cross-domain recommendations. We demonstrate that our extended profiling component and property path processor not only make it possible to generate these kinds of recommendations, but they can also significantly improve the quality of recommendations.

2 Approach and Experiments

Our extension consists of adding mechanisms to specify the source as well as the target domains. By specifying them, the user and item profiles are extended to embed this information. The new multi-domain profiles are the input to the similarity operator which computes, for instance, the neighborhoods, i.e. the most similar users with respect to each user in the dataset. In the experiments, we use the Facebook dataset (2nd LoD-enabled RS Challenge, ESWC 2015), which contains feedback for three domains: movies, books and music (see statistics in the table below). Moreover, items are directly given as DBpedia resources. Using these IRIs, we augmented the dataset with a subgraph extracted from DBpedia by keeping all paths between items up to three hops regardless of the direction.

	#Users	**#Items**	**#Ratings**	**Sparsity (%)**
Movie Domain	32,159	5,389	638,268	99.631
Books Domain	1,398	2,609	11,600	99.681
Music Domain	52,072	6,372	1,093,851	99.67

Queries. To show the benefits of having the feedback, features, and interconnections all in a single graph, we gradually introduce more information in each query (Figure 1). Query **Q1** is a traditional collaborative-filtering RecSPARQL query based only on liked movies (single-domain). In **Q2** (our extension) we introduced preferences from a second source domain, music. Note that the syntax allows one to define a domain (`?domain rdf:type recsparql:Domain`) and to set even multiple source and target domains (`recsparql:SourceDomain`, and `recsparql:TargetDomain`, correspondingly). The system adequately processes the additional information by creating the input to the recommendation model and customizing it. In **Q3** we additionally introduce with respect to Q2 some common features between movies and music items. To make use of interconnections in the graph, we introduced the processing of property paths in our

```
Q1 (traditional RecSPARQL query):
RECOMMEND ?user ?item.REC (AVG(?RATING)
    AS ?avgRat) USING CF
WHERE {
    ?user ex:likes ?item .
    ?item rdf:type ?type .
    FILTER(?SIMscore > 0.0 && (?type = dbo:Film ||
        ?type = sorg:Movie || ?type = rc:Movie_CW) }
BASED ON {
    ?user ex:likes ?item }
SETTINGS { (*) }
GROUP BY ?user ?item.REC
```

```
Q2 (our extension):
RECOMMEND ?user ?item.REC (AVG(?RATING)
    AS ?avgRat) USING CF
WHERE {
    ?user ex:likes ?item .
    ?item rdf:type ?domain .
    FILTER(?SIMscore > 0.0) }
BASED ON {
    ?user ex:likes ?item }
SETTINGS { (*), (▷),
    dbo:Film rdf:type recsparql:SourceDomain ,
    sorg:Movie rdf:type recsparql:SourceDomain ,
    rc:Movie_CW rdf:type recsparql:SourceDomain ,
    rc:MusicalPerformer rdf:type recsparql:SourceDomain ,
    rc:Band_MusicGroup rdf:type recsparql:SourceDomain ,
    dbo:Band rdf:type recsparql:SourceDomain ,
    sorg:MusicGroup rdf:type recsparql:SourceDomain ,
    dbo:Film rdf:type recsparql:TargetDomain ,
    sorg:Movie rdf:type recsparql:TargetDomain ,
    rc:Movie_CW rdf:type recsparql:TargetDomain }
GROUP BY ?user ?item.REC
```

```
Q3 (our extension):
RECOMMEND ?user ?item.REC (AVG(?RATING)
    AS ?avgRat) USING CF
WHERE {
    ?user ex:likes ?item .
    ?item rdf:type ?domain .
    ?item dbp:genre ?genre .
    ?item dbp:label ?label .
    ?item dc:subject ?subj .
    FILTER(?SIMscore > 0.0) }
BASED ON {
    ?user ex:likes ?item ,
    ?user ex:likes/dbp:genre ?genre ,
    ?user ex:likes/dbp:label ?label ,
    ?user ex:likes/dc:subject ?subj  }
SETTINGS { (*), (▷)
    dbo:Film rdf:type recsparql:SourceDomain ,
    sorg:Movie rdf:type recsparql:SourceDomain ,
    rc:Movie_CW rdf:type recsparql:SourceDomain ,
    rc:MusicalPerformer rdf:type recsparql:SourceDomain ,
    rc:Band_MusicGroup rdf:type recsparql:SourceDomain ,
    dbo:Band rdf:type recsparql:SourceDomain ,
    sorg:MusicGroup rdf:type recsparql:SourceDomain ,
    dbo:Film rdf:type recsparql:TargetDomain ,
    sorg:Movie rdf:type recsparql:TargetDomain ,
    rc:Movie_CW rdf:type recsparql:TargetDomain }
GROUP BY ?user ?item.REC
```

```
    COMMON:
(*) ?user rdf:type recsparql:User,
    ?item rdf:type recsparql:Item
(▷) ?domain rdf:type recsparql:Domain
```

Fig. 1: RecSPARQL queries used in the experiments (Left-top) Q1; (Left-bottom) Q2; (Right) Q3. Statements in red are applicable only in our extension.

cross-domain extension. This allows one to conveniently specify in the Based On clause any possible path between items from different domains. Our extensions uses the additional information in Q2 and Q3 to change the way user profiles are built. Consequently, the system directly alters the recommendation model based on the new neighborhoods. To choose the feature types used in Q3 (genre, label, and subject) we prioritized relevant ones, which were common to both domains and also show a high degree of interconnectivity. The relevance is based on Principal Component Analysis (PCA) and Information Gain (IG). These feature selection techniques measure how representative a feature type is for its corresponding domain. The interconnectivity is based on counting the number of item pairs from different domains which are connected by a feature of the same type. All queries were evaluated in a distributed fashion using Spark to alleviate the scalability challenge related to the neighborhood computation.

Results. Figure 2(A-B) shows the comparative results with respect to two metrics, mean reciprocal rank (MRR) and normalized discounted cumulative gain (NDCG). MRR tells us how well the recommendation model is doing in finding the first relevant item in the list of recommendations. NDCG gives a score for the complete list instead, taking into account the number of hits with respect to the test set but also with respect to the position of the hits in the list.

The plots clearly show, for different number of recommendations k, the benefits of introducing the additional graph components in each query. For instance, when we focus on a single metric, we found that from Q1 to Q2, the number of users with NDCG@50 equal to zero was reduced by 3.79%. From Q2 to Q3 this was reduced again by 3.3%, i.e. overall the reduction was 7.09%.

This was due to changes in the characteristics of the neighborhoods, which can be better understood by looking at the plots (C-E) in Figure 2. These plots show that the average number of neighbors in each neighborhood increases the more information is utilized to build them (C). Moreover, the standard deviation (D) shows that introducing a second domain affects only some users (probably those with common ratings in both domains). This inequality is later corrected when the features are included. Overall, the average similarity scores between the neighbors and active user consistently decreases, as the plot (E) shows.

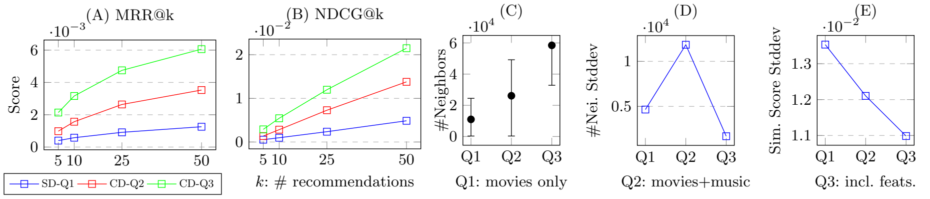

Fig. 2: (A-B) MRR@k−NDCG@k scores for the three queries; (C) Number of neighbors (min/avg/-max); (D) Std. deviation of the number of neighbors; (E) Std. deviation of the similarity score

3 Conclusions and Future Work

We believe there is still room for new tools on top of RDF-graphs to enable useful applications. While leveraging both the graph's content and structure continues to be a challenge, we showed that our extension of RecSPARQL for cross-domain recommendations fulfills its purpose of enhancing the quality of recommendations. Not only did feedback from the additional information source have a positive impact, but the features specified by means of property paths also helped to build fairer neighborhoods for the users.

References

1. Ayala, V.A.A., Przyjaciel-Zablocki, M., Hornung, T., Schätzle, A., Lausen, G.: Extending SPARQL for recommendations. In: SWIM (2014)
2. Cremonesi, P., Tripodi, A., Turrin, R.: Cross-domain recommender systems. In: Data Mining Workshops (ICDMW), IEEE (2011)
3. Fernández-Tobías, I.: Mining semantic data, user generated contents, and contextual information for cross-domain recommendation. In: UMAP (2013)
4. Musto, C., Lops, P., de Gemmis, M., et al.: Semantics-aware recommender systems exploiting linked open data and graph-based features. Knowl.-Based Syst. (2017)
5. Ricci, F., Rokach, L., Shapira, B. (eds.): Rec. Systems Handbook. Springer (2015)
6. Ristoski, P.: Exploiting semantic web knowledge graphs in data mining. Ph.D. thesis, University of Mannheim, Germany (2018)