

Using Linked Data to Build Open, Collaborative Recommender Systems*

Benjamin Heitmann and Conor Hayes

Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland
firstname.lastname@deri.org

Abstract

While recommender systems can greatly enhance the user experience, the entry barriers in terms of data acquisition are very high, making it hard for new service providers to compete with existing recommendation services. This paper proposes to build open recommender systems which can utilise Linked Data to mitigate the new-user, new-item and sparsity problems of collaborative recommender systems. We describe how to aggregate data about object centred sociality from different sources and how to process it for collaborative recommendation. To demonstrate the validity of our approach, we augment the data from a closed collaborative music recommender system with Linked Data, and significantly improve its precision and recall.

1. Introduction

Recommender systems like they ones deployed by Amazon¹ for books, Netflix² for movies, or Last.fm³ for music, greatly enhance the user experience of searching, exploring and finding new and interesting content.

However the entry barriers in terms of data acquisition are very high, which makes it hard for new service providers to compete with existing recommendation services in a domain.

Most real-world recommender systems employ collaborative filtering or combine it with another recommendation approach, such as content based recommendation (Adomavicius and Tuzhilin 2005). Collaborative filtering aggregates user ratings for items and uses statistical methods to discover similarities between items. The high entry barriers of providing good recommendations using collaborative filtering can be characterised by three challenges: providing recommendations for (a) **new items** or for (b) **new users** is a challenge if no data about the item or user is available at all. Together, the new-item and new-user problems are known as the ramp-up or cold-start problem (Schein et al. 2002).

*The work presented in this paper has been funded in part by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2).

Copyright © 2010, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

¹<http://amazon.com>

²<http://www.netflix.com/>

³<http://last.fm>

If the number of ratings is low compared to the number of items, then the (c) **sparsity** of the data will lead to ineffective recommendations (Balabanovic and Shoham 1997). The common cause of these challenges is the **data acquisition problem**: in order to provide recommendations, suitable data about the users and items needs to be available for the recommendation algorithm.

In closed recommender systems, such as on Amazon (Linden, Smith, and York 2003), the data acquisition problem is typically solved by collecting data about the users of the system, storing it in a closed, private database and implementing a recommendation algorithm for this data. In this paper, we describe an alternative to building such closed recommender systems: by utilising open data sources from the Linking Open Data (LOD) community project, and making the recommendation algorithm portable across data from different sources, it is possible to build open recommender systems, which can mitigate the challenges introduced by the data acquisition problem.

This paper describes how to use collaborative filtering on Linked Data to build recommender systems. As we explain in the background section 2.1, Linked Data includes semantic features, which could be used for content-based or knowledge-based recommendations. However, the inconsistent use of these semantic features makes the cost of exploiting them high. Instead we will focus on using collaborative filtering to exploit simple, binary connections between users and items. Such data can be found in any data source which is modelled after the principle of object centred sociality (Bojars et al. 2008). In addition, by using such abundantly available data, we can mitigate the new-item, new-user and sparsity problems by exploiting the Web scale of Linked Data.

1.1 Outline

The rest of the paper is organised as follows: We start by explaining the necessary background for building recommender systems on Linked Data: section 2.1 describes the principles of Linked Data, and which types of data from the Web of Data can be used for our open recommender system. Section 2.2 then describes the required data for different recommendation algorithms and the challenges which contribute to the high entry cost of new recommender systems.

Next we describe how to build a recommender system for the music domain, which can utilise Linked Data to mitigate the high entry costs for new recommender systems. Section 3.1 describes the necessary steps to process linked data for collaborative filtering. Section 3.2 then describes how Linked Data can be added to “fill in the gaps” of a recommender system. This is followed by a preliminary evaluation in section 4, for which we describe the implementation and the results. We conclude the paper by listing related work and discussing future research.

The main contributions of this paper are: (i) identifying how linked data can significantly reduce the entry barriers for starting and operating a recommender system, (ii) an architecture for open recommender systems based on Linked Data, (iii) describing and evaluating the application of collaborative filtering on Linked Data modelled after object centred sociality.

2. Background

In order to provide relevant recommendations a recommender system needs to have suitable data for the recommendation algorithm. In this section we are first going to explain the principles of Linked Data and the Web of Data and characterise the available data. Based on this discussion, we explain the different types of recommendation algorithms and discuss which algorithm is most suitable to the available data. Due to the large number of simple user-item connections, and due to the low consistency of properties describing items, we choose collaborative filtering instead of content or knowledge-based recommendation.

2.1 Linked data and the Web of Data

The term Linked Data refers to a set of best practices for publishing and connecting structured data on the web (Bizer, Heath, and Berners-Lee 2009). Taken together, all linked data constitutes the Web of Data. While the World Wide Web provides the means for creating a web of human readable documents, the Web of Data aims to create a web of structured, machine-readable data.

The Web of Data utilises technologies from the Semantic Web technology stack: the Resource Description Framework (RDF) provides a graph based data model and the basic, domain independent formal semantics for the data model (Decker et al. 2000); the SPARQL Query Language allows querying RDF data with graph patterns, and provides basic means for transforming RDF data between different schemata. In addition, technologies from the World Wide Web provide the fundamental infrastructure: Uniform Resource Identifiers (URIs) are used to provide globally unique identifiers for the data, and the HyperText Transfer Protocol (HTTP) is used for accessing and transporting the data.

In order to build a single Web of Data, all data providers have to follow the same guidelines for publishing their data and connecting it to other data sources. These guidelines are provided by the Linked Data principles (Bizer, Heath, and Berners-Lee 2009), which specify how to use the different standards of the Web of Data together:

1. Use URIs as names for things (and e.g. persons, places).

2. Use HTTP URIs so that people can look up and access those names via HTTP.
3. When someone looks up a URI, provide useful information, using the standards RDF and SPARQL.
4. Include links to other URIs, so that data about more things can be discovered.

The Linked Data principles have been adopted by an increasing number of data providers, especially from the Linking Open Data community project⁴, which makes free and public data available as linked data. The nucleus of the linked data cloud is formed by DBpedia⁵, which extracts RDF from wikipedia topic pages, and thus provides URIs and RDF data about topics from any domain. However the consistency of this data presents difficulties. As an example, DBpedia contains about 23000 musical artists with a total of 1200 distinct properties, at the time of writing. This makes it difficult to rely on consistent use of properties to describe the features of resources from dbpedia.

Social Web sites provide a big contribution to the linked data cloud, by making information about their users available. This data is modelled after the principle of object centred sociality (Bojars et al. 2008): persons are not only directly connected to other persons, but also indirectly via objects of a social focus. In this way a community is connected to each other not only via direct links from person to person, but also via their links to e.g. music from an artist. Such data uses the Friend of a Friend (FOAF) vocabulary for describing users and their connections to interests and other users, and the Semantically-Interlinked Online Communities (SIOC) vocabulary for describing user generated content on forums, weblogs and web 2.0 sites, as described in (Bojars et al. 2008). As an example, the DBTune project (Raimond, Sandler, and Mary 2008) makes such data about users and their connections to musical artists available. At the time of writing, the DBTune Myspace data contained at least 6 million user-item connections.

2.2 Recommender systems

Recommender systems require three components to provide recommendations (Burke 2002): (1) **background data**, which is the information the system has before the recommendation process begins, (2) **input data**, which is the information provided about the user in order to make a recommendation, and (3) the **recommendation algorithm** which operates on background and input data in order to provide recommendations for a user.

Different recommendation algorithms require different types of background and input data in order to provide recommendations. Current recommendation algorithms can be grouped in 3 classes (Burke 2002; Adomavicius and Tuzhilin 2005):

(i) **Collaborative filtering** aggregates ratings for items from different users, and uses similarities between items

⁴<http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData/>

⁵<http://dbpedia.org>

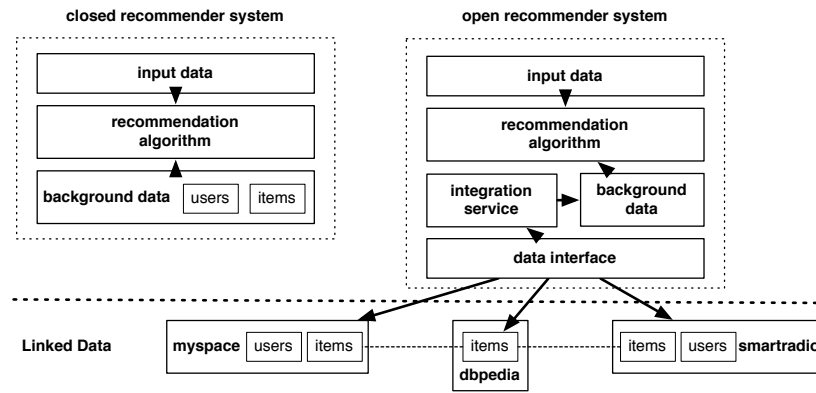


Figure 1: A comparison of a closed and an open recommender system

to recommend items. It is probably the most mature and widely implemented recommendation algorithm, because it achieves fairly good results and is easy to implement. It only requires data about the ratings between users and items as background data, no other information about either the users or the items is required. The input data usually consists of a user profile providing ratings for one or more items. The recommendation algorithm uses the background data to calculate the pair-wise similarity between all items or all users, and then uses the input data to recommend similar users or items.

(ii) **Content-based recommendation** uses the features of the items as the background data for the recommendation. These can either be directly derived from the content, e.g. keywords from text, tempo of music, or derived from the meta-data of the items, e.g. author, title and genre. The input data needs to describe the users preferences in terms of content features. Both the background and input data requires the consistent description of content features, in order to match the user preferences to the features of the content.

(iii) **Knowledge-based recommendation** aims to suggest items based on inferences about a users needs and preferences. This requires background data which includes knowledge about users and items, which is sufficient in consistency and scale for making inferences. The input data needs to provide knowledge about the user's needs and preferences which can be mapped to the knowledge about users and items in the background data.

To summarise the data required by the three recommendation algorithms: both content and knowledge-based recommendation require high quality background data about users and items. In contrast, the collaborative filtering algorithm only requires vectors of user-item connections. These can include numerical ratings, but can also operate on simple binary user-item connections.

Due to current problems with the consistency of attributes in Linked Data and the abundance of Linked Data about object centred sociality, we choose to implement our open recommender system using collaborative filtering. This class of recommendation algorithms is affected by three challenges (Adomavicius and Tuzhilin 2005) which we can mitigate by

adding background data from the Web of Data:

(a) **the new item problem**: to provide good recommendations for any item, the recommendation algorithm needs information about the item. If a new item has been added, then no information about user preferences has been collected for the item. This makes it challenging to provide collaborative recommendations for new items. (b) **the new user problem**: in order to personalise the recommendation, the recommendation algorithm needs a user profile. For collaborative recommendations new users are a challenge because the user has no profile of preferences connecting him to items. (c) **the sparsity problem**: if the number of ratings is low compared to the number of items in the background data then it will be hard to match other users or item profiles. In order to provide effective and relevant recommendations, collaborative recommendation algorithms need the connections between users and items to be dense.

3. Using Linked Data for open recommender systems

In the following sections we explain how to build an open recommender system using inexpensive Linked Data. We first discuss the architecture of an open recommender system. Then we show how Linked Data from different sources can be integrated for recommendations, and we describe the different steps of a collaborative filtering algorithm operating on the integrated data. We then describe how to exploit the Web scale of Linked Data to mitigate the high cost of data acquisition in closed recommender systems.

3.1 An architecture for integrating linked data for collaborative recommendations

As explained in section 2.2, recommender systems usually consist of three components. In order to use Linked Data, we need to extend the architecture of the recommender system with two components: the **data interface** allows accessing URIs via HTTP in order to acquire RDF data. This provides an abstraction layer for accessing the data. RDF libraries such as Redland⁶ provide a data interface component. The

⁶<http://librdf.org/>

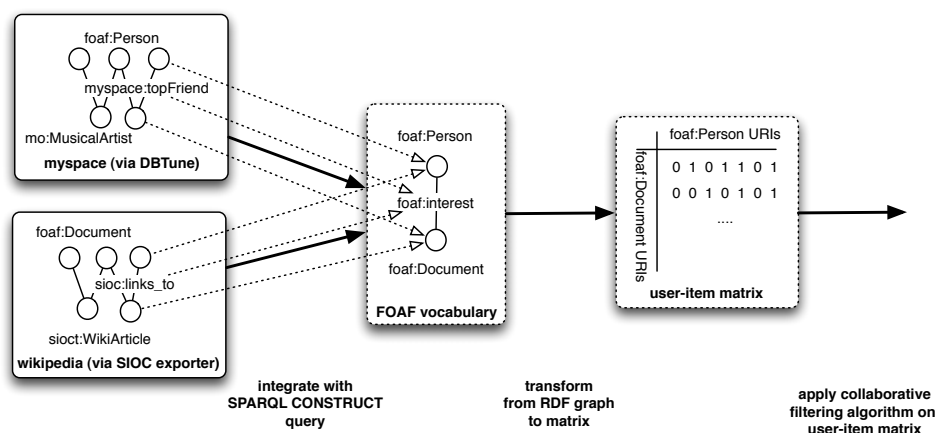


Figure 2: Processing Linked Data for Collaborative Recommendations

integration service transforms the data from the different sources into a unified representation, which we describe in the next section. Figure 1 compares the two architectures. For a detailed discussion of the components of Semantic Web applications see (Heitmann et al. 2009).

Figure 2 shows all the steps of processing Linked Data for collaborative recommendations: (1) **integrating** the data about user-item connections from different sources to a common vocabulary. (2) **Transforming** the representation of the data from an RDF graph to a user-item matrix. (3) **Applying** a specific collaborative filtering algorithm on the user-item matrix.

Integrating data is necessary, because different data sources will use different vocabulary to denote the connection between a user and an item. However, as we are only interested in those connections, extending the integration service only requires writing one new rule. Figure 2 shows the vocabulary used by two example data sources: One source uses the `foaf:Person` class for the users, `mo:MusicalArtist` for musicians and `myspace:topFriend` to connect both. The second source uses a different vocabulary to express the user-item connections, `foaf:Document` indicates the page with user information, `sioc:links_to` describes a link from that user page, and `sioc:WikiArticle` is a wikipedia article about the topic the user is interested in. By using SPARQL CONSTRUCT queries, we can instruct the data interface to fetch the data and then provide a rule for expressing the data with a different vocabulary (Passant and Raimond 2008). This is the query for transforming the first data source to FOAF:

```
CONSTRUCT { ?user foaf:interest ?item }
WHERE { ?user myspace:topFriend ?item .
?user rdf:type foaf:Person .
?item rdf:type mo:MusicalArtist }
```

The resulting RDF graph of user-item relations then needs to be **transformed** into a user-item matrix. Columns represent users, rows represent items, and their connection is indicated by a binary value. Representing the user-item connec-

tions as a matrix is necessary, because the user-item matrix is typically required by collaborative filtering algorithms for calculating similarity between users or items.

Finally the collaborative filtering algorithm can be **applied** to process the user-item matrix into its background data, which might be an item-item matrix, a user-user matrix or any other data structure (Montaner, López, and De La Rosa 2003).

3.2 Mitigating the cost of data acquisition

Collaborative filtering algorithms require sufficient data about users and items in order to make recommendations. We can mitigate the data acquisition problem for collaborative filtering algorithms by utilising Linked Data sources. They provide data which can be used to “fill in the gaps” in the background data.

Providing recommendations for new users: Consider the use case of a wikipedia editor creating a new account on a music recommendation web site. If a new user joins a web site in order to receive recommendations, then the system has no background data about the preferences of the user. However, if we can find Linked Data about the user, which connections him to the musical artists in the background data of the recommender system, then we can instantly provide recommendations for him. By using e.g. the SIOC MediaWiki exporter⁷ we could access SIOC data for a wikipedia editors homepage. This will include links to different topics for which the user indicates interest or to which the user has contributed. If the recommender system has background data about one or more of these topics, then it can add one row to the user-item matrix, and provide recommendations for the user.

Adding new items to the inventory: A recommender system can not recommend a new item until it has been rated by other users. Using e.g. Linked Data from DB-Tune (Raimond, Sandler, and Mary 2008) about MySpace musicians, we can add data about the new musician, without waiting for users to add the item to their preferences. One

⁷<http://ws.sioc-project.org/mediawiki/>

way to do this, is to collect all users from MySpace which are connected to the new musician and to at least one other musician, for which the recommender system already has background data. If we add this data to the user-item matrix, there will be one new row for the item and multiple new user columns. The anonymised data of these users from MySpace allows us to add indirect connections between the new item and existing items.

Improving recommendations by reducing sparsity: If the number of connections between users and items is low, compared to the total number of items, then the number and quality of recommendations will be low. In order to discover similarities between items, the user-item matrix should provide as many indirect connections between items as possible. One way to do this, is by adding new users from a relevant external source, e.g. from the DBTune MySpace data which are connected to more than one musician. This will add multiple user columns, which add to the total number of connections between users and items. This in turn provides more background data for the collaborative filtering algorithm to discover similarities.

4. Preliminary Evaluation

To demonstrate the validity of our approach, we augment the data from a real collaborative music recommender system with Linked Data, and significantly improve its precision and recall.

Smart Radio was the first on-line music streaming and recommendation service (Hayes and Cunningham 2000) which operated on similar principles to Last.fm but predated it. The Smart Radio service used a closed database as background data for the recommendation algorithm. However as a research proof-of-concept it only had a limited number of users and items: 190 users and 330 musical artists.

We have expressed the user-item connections of the Smart Radio background data as Linked Data using the FOAF vocabulary. The users have been anonymised for this, and all musical artists have been linked to both DBTune MySpace data and DBpedia. We then used the links from Smart Radio artists to MySpace artists to reduce the sparsity of background data, as explained in the previous section. This added 11000 users from MySpace and around 25000 new connections between users and artists. We then created user-item matrices for the Smart Radio FOAF data and the combined FOAF data from Smart Radio and MySpace.

As recommendation algorithm, we applied a binary cosine similarity measure from (Sarwar et al. 2001):

$$\text{cosine}(i_1, i_2) = \frac{\text{count}(i_1, i_2)}{\text{count}(i_1)\text{count}(i_2)} \quad (1)$$

This is a simple baseline recommendation algorithm, which can be used to compute an item-item similarity matrix. Each entry in the item-item matrix expresses the similarity between two items. $\text{count}(i)$ is the number of users who have a connection to item i , and $\text{count}(i_1, i_2)$ is the number of users who have a connection to both item i_1 and item i_2 . Ranking the entries in row i provides the most similar items to i .

In order to evaluate the recommendation results, we compared them to the last.fm recommendations for the same artist: $B_{artists}$ is the set of all artists in the background data, $R_{lastfm}(a)$ is the set of last.fm recommendations for artist a . Then $D(a) = B_{artists} \cap R_{lastfm}(a)$ is the set of relevant recommendations for artist a . For a second recommendation $R(a)$ we can then define precision and recall (Herlocker et al. 2004): Precision is the number of relevant artists in a specific recommendation divided by the total number of recommendations, while recall is the number of relevant artists in a recommendation divided by the number of relevant artists in our background data:

$$\text{precision}(R(a)) = \frac{|R(a) \cap D(a)|}{|R(a)|} \quad (2)$$

$$\text{recall}(R(a)) = \frac{|R(a) \cap D(a)|}{|D(a)|} \quad (3)$$

Results: Computing the average recall and precision for both data sets, shows that augmenting the Smart Radio data significantly improves the recommendations. Using only Smart Radio data we get an average precision of 2% and average recall of 7%. By augmenting the Smart Radio data with Linked Data from MySpace we get an average precision of 14% and average recall of 33%. This increase in the relevance of the recommendations shows that our approach is a viable first step.

5. Related work

“Foafing the Music” (Celma and Serra 2008), might be the first recommender system for music which used RDF data. In order to provide its recommendations it crawled data from a large number of web sites, such as Amazon, Eventful.com, music newspapers, and music blogs (about 1100 different sources). The user preference data is stored as FOAF data. The recommendation algorithm combines knowledge-based and content-based approaches. No evaluation of the recommendation results is performed. While this recommender system saves preferences as FOAF data, it does not use RDF data or Linked Data as background or input data of the recommendation algorithm.

(Passant and Raimond 2008) describe the Linked Data from the music domain, which could be used for recommendations. However the suitability of Linked Data for different recommendation algorithms is not discussed, and no implementation is described. (Raimond, Sutton, and Sandler 2009) describe their work on the Music Ontology, which can be used to encode meta-data about music and connect it to the LOD cloud. Based on this, an approach for providing content-based recommendations is discussed, which uses not only meta-data but also the audio signal of the music for recommendations.

(Shani, Chickering, and Meek 2008) compares the efficiency of recommendation systems built using data from the public web as background data, with recommender systems using closed, private data sets. Search engine and web crawler results are used to build a user-item matrix for a collaborative filtering algorithm. The recommendation results are evaluated against the results of recommender systems from the same domain with a private, closed data set.

The evaluation shows that the results are on an equal level. However, their collaborative filtering algorithm does not use RDF or Linked Data as background data. (Berkovsky, Kufflik, and Ricci 2007) introduces different approaches for exchanging data about users or items between collaborative recommender systems. However, the underlying technology for storing and exchanging data is not discussed.

6. Discussion

In 2001 the Smart Radio researchers attempted to use RDF to link its collaborative database to external sources to improve recommendation quality and portability (Clerkin, Cunningham, and Hayes 2001). However, as there were no resources available like those of the Linking Open Data community initiative, the effort eventually failed. This paper demonstrates how Linked-data can be used to alleviate the data acquisition bottleneck and create new opportunities through the development of open recommender systems.

7. Conclusion

In this paper we have shown that Linked Data can be used to lower the high entry barriers of operating a recommender system. By mitigating the data acquisition problem, Linked Data about object centred sociality can be used to “fill in the gaps” of a collaborative filtering algorithm. This allows us to acquire the data which is needed to mitigate the problems of providing relevant recommendations for new users and new items, as well as increasing the value of recommendations in general. To demonstrate the validity of our approach we have augmented the data of a closed recommender system with Linked Data. Evaluation shows a significant improvement in precision and recall of the recommendations.

As future research we will continue the evaluation with bigger data sets from different domains. In addition we will incorporate knowledge-based recommendation based on Linked Data into our recommendation approach.

References

- Adomavicius, G., and Tuzhilin, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17(6):734–749.
- Balabanovic, M., and Shoham, Y. 1997. Fab: Content-Based, Collaborative Recommendation. *Communications of the ACM* 40(3):67.
- Berkovsky, S.; Kufflik, T.; and Ricci, F. 2007. Cross-Domain Mediation in Collaborative Filtering. In *International Conference on User Modeling*, 355–359. Springer-Verlag Berlin, Heidelberg.
- Bizer, C.; Heath, T.; and Berners-Lee, T. 2009. Linked data-the story so far. *Journal on Semantic Web and Information Systems (in press)*.
- Bojars, U.; Passant, A.; Cyganiak, R.; and Breslin, J. 2008. Weaving sioc into the web of linked data. In *Linked Data on the Web Workshop*.
- Burke, R. 2002. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction* 12(4):331–370.
- Celma, O., and Serra, X. 2008. Foafing the music: Bridging the semantic gap in music recommendation. *Web Semantics: Science, Services and Agents on the World Wide Web* 6(4):250–256.
- Clerkin, P.; Cunningham, P.; and Hayes, C. 2001. Ontology Discovery for the Semantic Web Using Hierarchical Clustering. *Semantic Web Mining Workshop co-located with ECML/PKDD*.
- Decker, S.; Melnik, S.; Van Harmelen, F.; Fensel, D.; Klein, M.; Broekstra, J.; Erdmann, M.; and Horrocks, I. 2000. The semantic web: The roles of XML and RDF. *IEEE Internet computing* 4(5):63–73.
- Hayes, C., and Cunningham, P. 2000. Smart radio-building music radio on the fly. In *International Conference on Knowledge Based Systems and Applied Artificial Intelligence*. Springer Verlag.
- Heitmann, B.; Kinsella, S.; Hayes, C.; and Decker, S. 2009. Implementing semantic web applications: reference architecture and challenges. In *Workshop on Semantic Web Enabled Software Engineering*.
- Herlocker, J.; Konstan, J.; Terveen, L.; and Riedl, J. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* 22(1):53.
- Linden, G.; Smith, B.; and York, J. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7(1):76–80.
- Montaner, M.; López, B.; and De La Rosa, J. 2003. A taxonomy of recommender agents on the internet. *Artificial intelligence review* 19(4):285–330.
- Passant, A., and Raimond, Y. 2008. Combining Social Music and Semantic Web for music-related recommender systems. In *Social Data on the Web Workshop*.
- Raimond, Y.; Sandler, M.; and Mary, Q. 2008. A web of musical information. In *International Conference on Music Information Retrieval*.
- Raimond, Y.; Sutton, C.; and Sandler, M. 2009. Inter-linking music-related data on the web. *IEEE Multimedia* 16(2):52–63.
- Sarwar, B.; Karypis, G.; Konstan, J.; and Reidl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *International Conference on the World Wide Web*, 285–295. ACM New York, NY, USA.
- Schein, A. I.; Popescul, A.; H., L.; Popescul, R.; Ungar, L. H.; and Pennock, D. M. 2002. Methods and metrics for cold-start recommendations. In *Conference on Research and Development in Information Retrieval*, 253–260. ACM Press.
- Shani, G.; Chickering, M.; and Meek, C. 2008. Mining recommendations from the web. In *ACM Conference on Recommender Systems*, 35–42. ACM New York, NY, USA.