Exploiting the Web of Data in Model-based Recommender Systems

Tommaso Di Noia[1], Roberto Mirizzi[1,2], Vito Claudio Ostuni[1], Davide Romito[1]*
[1]Politecnico di Bari – Via Orabona, 4 – 70125 Bari, Italy
[2]HP Laboratories – 1501 Page Mill Road – Palo Alto, CA 94304
t.dinoia@poliba.it, {mirizzi,ostuni,d.romito}@deemail.poliba.it

ABSTRACT

The availability of a huge amount of interconnected data in the so called Web of Data (WoD) paves the way to a new generation of applications able to exploit the information encoded in it. In this paper we present a model-based recommender system leveraging the datasets publicly available in the Linked Open Data (LOD) cloud as DBpedia and LinkedMDB. The proposed approach adapts support vector machine (SVM) to deal with RDF triples. We tested our system and showed its effectiveness by a comparison with different recommender systems techniques – both content-based and collaborative filtering ones.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval

Keywords

Model-based RSs, SVM, Linked Data, DBpedia, LinkedMDB, Semantic Web, MovieLens, Precision, Recall

1. INTRODUCTION

The need for a semantic representation of data and user profiles has been identified as one of the next challenges in the field of recommender systems [8]. There are several advantages in the use of semantic data in recommendation tasks spanning from a pure knowledge-based perspective (e.g., we may have a richer representation of data) to more practical points of view such as the easy adaptation of the same approach to different domains. In the recent years, thanks to the Web of Data advance, we are assisting to a flourishing of semantic datasets freely available on the Web encoding machine-understandable RDF triples related to different domains and sometimes representing different points of view on the same domain. All this information can be exploited to model items and user profiles in an LOD-enabled content-based recommender system where the domain knowledge plays a fundamental role. One of the main components of these systems is represented by the *Content Analyzer* (CA) [6]. This module is responsible for the preprocessing of the information usually coming from textual sources and for extracting keywords used to model both the items and the user profile. The use of the Web of Data can contribute to reduce the effort associated to the definition of the Content Analyzer. In fact, in order to retrieve and use data related to a specific domain of interest, the LOD-based CA has just to formulate SPARQL[1] queries. Moreover, as resources in LOD datasets are identified by unique URIs and are semantically interlinked with each other, the problems related to a keyword-based approach such as synonymy and polysemy [8] are automatically solved.

In this paper we present a model-based approach for a content-based recommender system exploiting exclusively LOD data to represent both the information on the items and on the user profiles. We show how a model-based approach can be easily adapted to cope with the Web of Data. In this paper we use a Support Vector Machine but the overall framework does not rely on any particular classifier. We extensively tested our system to show the effectiveness of the adoption of semantic data for recommendation tasks. The obtained results evidence quality and richness of the information encoded in LOD datasets and we believe they represent a preliminary step towards a new generation of semantic-enabled recommender systems. It is noteworthy that although we performed our experiments in the movie domain, the techniques we proposed are not tied to this particular domain and can be easily adapted to cope with whatever RDF set of triples.

The remainder of the paper is structured as follows: in Section 2 we illustrate the main concepts behind the Web of Data. In Section 3 we detail our model-based approach to recommendation. Section 4 is dedicated to the evaluation. In Section 5 we give a concise overview of related work. Conclusion and future work close the paper.

2. THE WEB OF DATA

The term *Web of Data*, often referred to as *Semantic Web*, *Web 3.0* or *Linked Data*, indicates a new generation of technologies responsible for the evolution of the current Web [2] from a Web of interlinked documents to a Web of interlinked data. The goal is to discover new knowledge and

[*] The authors are listed in alphabetical order.

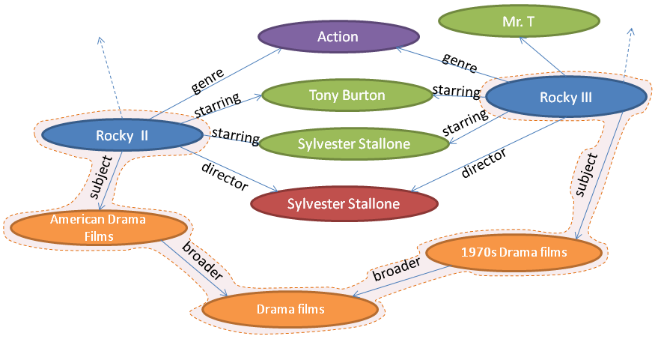
[1] http://www.w3.org/TR/rdf-sparql-query/

RecSys'12, September 9–13, 2012, Dublin, Ireland.

value from data, by publishing them using Web standards (primarily RDF) and by enabling connections between heterogeneous datasets. As for the traditional Web, the *Web of Data* spans multiple domains – people, movies, music, books, scientific publications, just to cite a few. In particular, the term Linked Open Data (LOD) denotes a set of best practices for publishing and linking structured data on the Web. The project includes dozens of RDF datasets interlinked with each other to form a giant global graph, the so called Linked Open Data cloud. DBpedia is a first-class citizen in this cloud since it represents the nucleus of the entire LOD initiative [1]. The data are automatically extracted from freely available Wikipedia dumps and each article in Wikipedia is represented by a corresponding resource URI in DBpedia. Several RDF statements are generated for each resource by extracting information from various parts of the Wikipedia articles (e.g., from the categories at the bottom of the page and from the *infoboxes* at the right side of the page). This allows automatic agents to exploit the extracted structured information by querying the dataset via its SPARQL endpoint. Being based on Wikipedia, DBpedia is multi-lingual and cross-domain. This feature makes DBpedia a hub for Linked Open Data: domain-specific datasets can be connected to it to form a single, interconnected data space. Most of the semantic information encoded in DBpedia is represented via the properties dcterms:subject and skos:broader. They are used to represent a relation of hyponymy between resources. In particular, they link respectively a resource (e.g., a movie) to its category (e.g., *American drama films*) and more specific categories to more generic ones (e.g., *American drama films* to *Drama films*). In Figure 1(a) we show an excerpt of the graph containing properties and resources coming both from DBpedia and from LinkedMDB (i.e., the RDF version of IMDB).

3. MODEL-BASED RECOMMENDATIONS WITH LINKED DATA

The user profile consists of a model about the user preferences, i.e., a description of the types of items the user is interested in. There are many possible alternative representations of this description, but a common one is a function that for any item predicts the likelihood that the user is interested in that item. The application of Machine Learning techniques is a typical way to achieve the task of learning user profiles in model-based recommender systems. Creating a model of the user preferences from the user history is a form of classification learning wherein each item has to be classified as interesting or not with respect to the user tastes. Model-based recommender systems and in particular content-based ones share some characteristics with text categorization tasks. Machine learning techniques for text categorization/classification has been extensively applied in the field of recommender systems but, to our knowledge, they have not previously been used to build content-based recommender systems that benefits from the usage of Linked Open Data. Since in our system the items to be recommended are resources belonging to semantic datasets, we need to build a model able to deal with such data. In reference to Figure 1(a), the items to be recommended are the movies and they are described by the nodes they are connected to. The example is about the movie domain, nevertheless this approach can be extended to any domain covered by the Web of Data. By exploiting the ontological information encoded via dcterms:subject and skos:broader properties, we are allowed to perform a semantic expansion of the item description and then to catch implicit relations and hidden information, i.e., information that is not detectable just looking at the nodes directly linked to the item. As an example, if we look at the graph in Figure 1(a), we see that the two movies *Rocky II* and *Rocky III* implicitly have the *Drama films* category in common. The information discovered by exploiting the taxonomic structure of the categories increases the number of common features between two items. In Section 4 we will show how this semantic expansion improves the results of the recommendation.



(a)

	STARRING			DIRECTOR	GENRE	SUBJECT+BROADER		
	Tony Burton	Mr. T	Sylvester Stallone	Sylvester Stallone	Action	1970s Drama Films	Drama Films	American Drama Films
Rocky II	0.67	0	0.31	0.66	0.11	0	0.13	0.23
Rocky III	0.67	0.56	0.31	0.66	0.11	0.22	0.13	0

(b)

Figure 1: (a) Sample RDF graph extracted from DBpedia and LinkedMDB; (b) Matrix representation of *property resource-indexes*.

In our approach we transform the RDF graph describing a domain of interest in a feature vector representation that is suitable for the classification task. In a classic *bag of words* model, documents are represented by a set of representative keywords (index terms). We adapt the bag of words model in order to deal with RDF triples to obtain a **bag of resources** model. Taken an item from the collection, for each property we extract all the resources that are linked by the current property to the item and we build an index of resources corresponding to that property (i.e., a *property resource-index*). With respect to a given property, each item (i.e., movie) is represented by a vector in a multi-dimensional space, where each dimension corresponds to a resource from the vocabulary. For example, referring to Figure 1(a), the resource-index for the property *starring* is constituted by the resources *Mr. T*, *Tony Burton* and *Sylvester Stallone*. Considering all the properties, each item is represented as a unique vector of weights where each weight indicates the degree of association between the item and the resource with respect to a property. These weights are the TF-IDFs and they are computed distinctly for each property resource-index. Figure 1(b) shows the matrix of TF-IDF weights obtained from the graph in Figure 1(a). We point out that each property resource-index is separated from the others. For example, the resource *Sylvester Stallone* is more frequent in the *starring resource-index* than in the *director resource-index*, because *Sylvester Stallone* starred in more movies than he directed. Another important aspect we want to stress is that if we did not consider the *subject-broader* path (i.e., our semantic expansion), we would lose some matchings between the feature vectors describing the two movies *Rocky II* and *Rocky III*. In fact, as stated before, if we considered only the *subject*, we would not have the category *Drama Films* in the *subject resource-index*.

Support Vector Machine. SVM classifier is based on statistical learning theory developed by Vapnik [14], which uses the principle of Structural Risk Minimization instead of Empirical Risk Minimization, as a supervised machine learning technique. We chose SVM because it is well known that it works well in text classification tasks and our classification problem for learning the user profile has a lot of commonalities with them. Some of these commonalities are the sparse nature of the feature vector and the high dimensionality of the input space. As argued by Joachims [4], SVMs offer two important advantages for text classification task: (1) term selection is often not needed, as SVMs tend to be fairly robust with respect to over-fitting and can scale up to considerable dimensionalities; (2) no human and machine effort in parameter tuning on a validation set is needed. When the decision boundary is not linear we need to transform data into a higher dimensional space using a mathematical transformation known as the kernel trick. We tested three of the main used kernel functions: *Linear*, *Polynomial* and *RBF* kernel. We selected the *RBF* kernel because it proved to be the best performing in our domain. As SVM tool, we used the WEKA[2] SVM(SMO) implementation. The outputs of the SVM are used to build a logistic model able to give posterior probability estimates for the classes. The output of the recommender will be a ranked list of values between 0 and 1 obtained from the logistic model.

4. EVALUATION

The evaluation we present here aims to analyze two different aspects of the proposed approach: (1) inspecting what is the information contained within the datasets of `Linked Open Data` that allows the system to achieve the best results in terms of precision and recall; (2) comparing our system with other relevant approaches, both content-based, collaborative filtering and hybrid ones, in terms of quality of the results. To these purposes, we conducted several experiments using the 1M `MovieLens` dataset. Being our approach based on `LOD` as knowledge base, the first step we had to do was to align the movies in the `MovieLens` dataset with the movies in `DBpedia`. The alignment was mainly done querying `DBpedia` via its `SPARQL` endpoint. A dump of the obtained mapping is available at the url: `http://sisinflab.poliba.it/mapping-movielens-dbpedia-1M.zip`. The test set we extracted from the `MovieLens` dataset has 20 rates per user. For this reason, we were able to compute *Precision@N* and *Recall@N* for values of *N* in the interval $[1, 20]$. In order to avoid potential bias by some user profiles, we carried out a 5-fold cross-validation. Being our approach based on a binary classifier, we converted the 1-5 Likert scale used by Movielens into a binary one, where ratings above 3 are considered as *like* and the others as *dislike*.

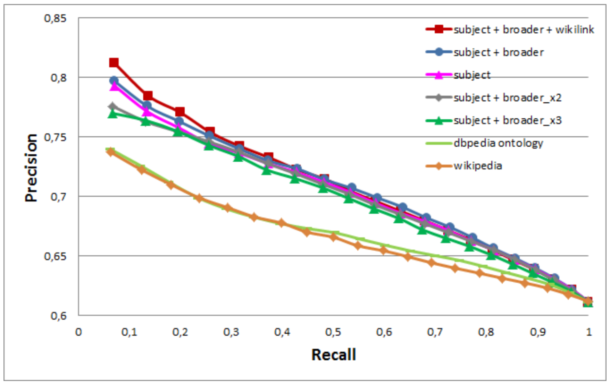


Figure 2: Precision and Recall curves obtained with different groups of properties in `LOD`.

In Figure 2 we plot the precision and recall curves for different sets of properties selected from the knowledge base we consider. The intent is to analyze the contribution that ontological information contained within `LOD` datasets may give to the recommendation. In particular, all the curves in Figure 2 refer to our approach and are based on the SVM classifier. We focused on evaluating the importance of the properties `dcterms:subject` and `skos:broader`. Moreover, in this test we evaluated also the importance of the property `dbpedia-owl:wikiPageWikiLink`, that indicates a link between two pages in Wikipedia. Intuitively, if there is a link from one page to another page in Wikipedia, it is reasonable to think the two pages are somehow related. Finally, in this part of the evaluation we also included the properties of the `DBpedia`-Ontology (e.g., in the movie domain some examples of these properties are `dbpedia-owl:starring`, `dbpedia-owl:director`, etc.). The red curve with square markers evidences the best results are achieved for a combination of properties that include: (a) the `DBpedia`-Ontology properties, (b) the `dbpedia-owl:wikilink` property, (c) the `dcterms:subject` property and (d) the first level of the `skos:broader` property (i.e., the categories directly linked to movies and the categories directly linked to them). The next four curves differ from the previous one because they do not include the `dbpedia-owl:wikilink` property and moreover they consider different levels for the `skos:broader` property: the more we use general categories for the recommendation (after the first level), the worse the results are. Finally, in the last two curves we do not consider any taxonomic information at all, but only the standard structured information we can usually find in content-based RS. In this case we obtain comparable results when using only either the `DBpedia`-ontology or Wikipedia. In Figure 3 we compare our approach (indicated by the red curve with square markers) with recent related work, both content-based, collaborative-filtering and hybrid ones. The azure curve with circle markers refers to the hybrid approach presented in [13]. The magenta curve with triangle markers shows the results of precision and recall for a collaborative-filtering approach where the measure of similarity between the ratings of two users is the Pearson correlation coefficient. It is the measure most commonly used in neighborhood-based CF systems [7]. We used the Apache Mahout[3] implementation for CF. The results are fairly comparable to our approach. The orange curve with diamond markers exploits Wikipedia to feed a

[2] `http://www.cs.waikato.ac.nz/ml/weka/`

[3] `http://mahout.apache.org/`

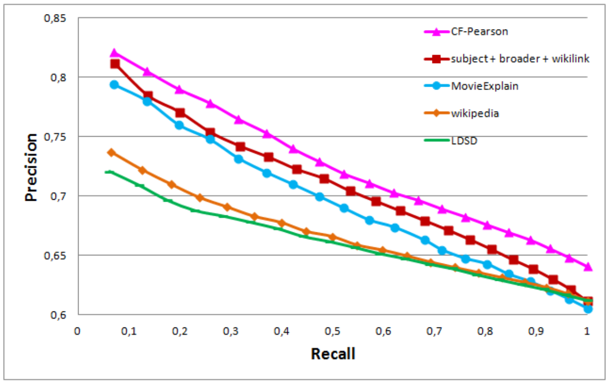


Figure 3: Comparison with several content-based, collaborative filtering and hybrid approaches.

content-based recommender system, as proposed in [5]. Finally, the green curve with dash markers refers to a CB recommender system leveraging DBpedia [9].

5. RELATED WORK

Using the Web of Data as knowledge base for recommender systems is a quite innovative and recent idea. A lot of approaches have been proposed to tackle the well-known issues of recommender systems (both content-based and collaborative filtering), but there are few of them that exploit the huge amount of information encoded in Linked Open Data. Since it would be impossible to give an exhaustive and worth view of all the approaches proposed in state-of-the-art, we refer the interested reader to [7, 12, 10]. In the following we analyze relevant approaches to recommendation exploiting the Web of Data and approaches we compared to in Section 4. One of the approaches we compared to is the *Linked Data Semantic Distance* (*LDSD*) [9]. There DBpedia is used as information source to compute recommendations. Differently than our approach, they do not perform any semantic expansion of the resources. However, such expansion proved to improve the overall quality of the results (cf. Figure 2 and Figure 3). *MoviExplain* [13] is another system we compared to in the evaluation section. The authors present a hybrid approach where they group together users exhibiting highly correlated ratings on set of movies. They leverage only text information sources, while we exploit structured and disambiguated information contained within RDF triples. In [5] the authors use the text content and the hyperlink structure of Wikipedia pages to identify similarities between movies. The aim is to check whether Wikipedia may improve the results of recommendations. However, being the approach based only on unstructured text and on hyperlinks, the approach does not significantly improves the accuracy of the system. In [3] the authors suggest to use Linked Open Data, to alleviate well known issues of CF recommender systems, such as new-user, new-item and sparsity problems. Sen et al. [11] investigate how the use of tags generates high-quality recommendations.

6. CONCLUSION AND FUTURE WORK

As of today, the Web of Data contains a huge amount of structured information publicly available to end-users and service providers. In this paper we have shown how the knowledge encoded in the Linked Open Data cloud can be effectively exploited to model a performing content-based (CB) recommender system. One of the advantages of using the LOD data for CB engines is the mitigation of the *limited content analysis* issue. Indeed, the heterogeneity of topics and contexts represented in the cloud as well as its interlinked nature favors an easy selection and exploitation of new diverse features/properties for a specific domain. The ontological nature of the data we find in LOD datasets has proven to be useful to increase the overall accuracy of the system. The results presented here are an initial step of a comprehensive analysis and investigation on the different uses of LOD data in the field of RSs. We believe the very promising results of our experiments have set a good point in favor of the exploitation of these datasets for recommendation tasks. We are currently working on the modeling of similarity measures different than TF-IDF, such as BM25, and we are investigating on how to have a semantic expansion of the resources used to compute recommendation results by exploiting not only the taxonomy of categories but also other semantic relations occurring within the RDF semantic graph. We are also experimenting the use of other classifiers different from the one used in this paper, such as kNN and Naïve-Bayes and we are performing experiments on different domains.

Acknowledgments. The authors acknowledge partial support of HP IRP 2011. Grant CW267313.

7. REFERENCES

[1] S. Auer et al. Dbpedia: a nucleus for a web of open data. In *Proc. of 6th ISWC and 2nd ASWC*, ISWC'07/ASWC'07, pages 722–735, 2007.

[2] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, 2011.

[3] B. Heitmann and C. Hayes. Using linked data to build open, collaborative recommender systems. In *AAAI Spring Symposium: Linked Data Meets AI*, 2010.

[4] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proc. of 10th ECML-98*, pages 137–142, 1998.

[5] J. Lees-Miller, F. Anderson, B. Hoehn, and R. Greiner. Does wikipedia information help netflix predictions? In *Proc. of the 7th Int. Conf. on Machine Learning and Applications*, ICMLA '08, pages 337–343, 2008.

[6] P. Lops, M. Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, pages 73–105. 2011.

[7] P. Melville and V. Sindhwani. Recommender systems. In *Encyclopedia of Machine Learning*, pages 829–838. 2010.

[8] S. E. Middleton, D. D. Roure, and N. R. Shadbolt. Ontology-based recommender systems. *Handbook on Ontologies*, 32(6):779–796, 2009.

[9] A. Passant. dbrec: music recommendations using dbpedia. In *Proc. of 9th Int. Sem. Web Conf.*, ISWC'10, pages 209–224, 2010.

[10] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

[11] S. Sen, J. Vig, and J. Riedl. Tagommenders: connecting users to items through tags. In *Proc. of 18th WWW*, WWW '09, pages 671–680, 2009.

[12] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009, 2009.

[13] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Moviexplain: a recommender system with explanations. In *Proc. of the 3rd ACM Conf. on RSs*, pages 317–320, 2009.

[14] V. N. Vapnik. *The nature of statistical learning theory.* Springer, New York, 1995.