Knowledge-enabled Recommender Systems: Models, Challenges, Solutions (Extended Abstract)*

Tommaso Di Noia

SisInf Lab, Polytechnic University of Bari
tommaso.dinoia@poliba.it

Introduction

Together with the rapid growing of information we daily produce, we have assisted to the flourishing of new tools and techniques whose aim is to help users in accessing such information in a personalized way. In the Information Overload era we are living in, the amount of information exceeds the users capability of processing and using it [39]. Huge and fast growing number of possibilities overwhelms users, leading them to make poor decisions and feel anxiety and unsatisfaction [37]. Recommender Systems (RSs) [32] are a family of information filtering tools which have proven to be valuable means in assisting users to find, in a personalized manner, what is relevant for them in such overflowing complex information spaces. They provide users with personalized access to large collections of resources.

The main task of a recommendation engine is typically to estimate the relevance of unknown items for a target user and recommend the Top-N items by considering for each user the best N items with highest utility [32]. Typically, the utility of an item is represented by a numerical rating, which indicates how much a particular user liked such item. However, the utility is not defined for each pair of users and items and usually it is available only for a small subset of them. Such subset represents the input of a generic recommender system, whose objective is to estimate the utility of all the remaining pairs. A formal definition of the recommendation problem has been given in [1] and defined as follows.

Definition 1. *Let U represent the set of users and I the set of items in the system. Potentially, both sets can be very large. Let $f : U \times I \rightarrow R$, where R is a totally ordered set, be a utility function measuring the relevance of item $i \in I$ for user $u \in U$. Then, the recommendation problem consists in finding for each user u such item $i^{max,u} \in I$ maximizing the utility function f. More formally, this corresponds to the following:*

$$\forall u \in U,\ i^{max,u} = arg\ max_{i \in I}\ f(u, i)$$

* This is an extended abstract of the keynote lecture given at the Third International Workshop on Knowledge Discovery on the Web (KDWeb 2017)

Depending on the the way the utility function is estimated and the availability of additional data about the characteristics of items for example, there are different types of recommendation techniques. The main two are: collaborative filtering and content-based. A complete list of collaborative filtering and content-based techniques together with their hybrid versions is given in [5] and in [32].

Collaborative Filtering Recommendation. Collaborative Filtering is the process of filtering or evaluating items using the opinions of other people [36]. According to [4] there are two main types of collaborative filtering methods: memory-based and model-based. Memory-based CF uses a particular type of Machine Learning methods: nearest neighborhood (k-NN) algorithm. In particular, it does not require any preliminary model building phase, since predictions are made by aggregating the ratings of the closest neighbours. Conversely, model-based techniques first learn a predictive model which is eventually used to make predictions. Memory-based approaches can be classified either in user-based [15] or item-based [35].

Content-based Recommendation. Content-based RSs recommend an item to a user based upon a description of the item and a profile of the user's interests [30]. Briefly, the basic process performed by a content-based recommender consists is matching up the attributes of a user profile with the attributes of a content object (item) [8]. Differently from collaborative filtering, such recommendation approach relies on the availability of content features describing the items. A high level architecture of a content-based RS is presented in [8]. There are two main content-based recommendation approaches: *heuristic-based* or *model-based*. Approaches using heuristic functions have their roots in Information Retrieval and Information Filtering. Items are recommended based on a comparison between their content and a user profile. The idea is to represent both items and users using typical IR techniques [3], e.g. vectors of terms, and compute a match between their representations. The user profile consists in a vector of terms built from the analysis of the items liked by the user. Model-based approaches [29] use Machine Learning techniques to learn a model of the user's preferences by analyzing the content characteristics of items the user rated. Content-based methods can have several limitations. For a complete and detailed description of content-based recommendation techniques refer to [8,30]. The main one is the *content overspecialization* which consists in the incapability of the system to recommend relevant items which are different to the ones the user already knows.

Knowledge-enabled Recommender Systems.

In content-based approaches, the information exploited by recommendation engines is very often encoded by bag of words or, more recently, by word embeddings [20]. In all these cases, no explicit semantics is associated to the contextual data. Nowadays `Linked Open Data` datasets represent a huge repository of different kinds of knowledge spanning from sedimentary-one such as encyclopedic,

linguistic, common-sense and so on, to real-time one such as data streams, events, etc. If we consider encyclopedic datasets such as DBpedia [18] or Wikidata [41], we have access to a huge amount of factual knowledge referring to a variety of topics. In order to effectively incorporate Linked Open Data in recommendation applications there are several aspects to consider. Ultimately, the goal is to provide the system with background knowledge about the domain of interest in the form of a knowledge graph. In a high level architecture of a component in charge of retrieving portions of the LOD graph regarding the items in the system which are used to form the knowledge graph there are two main modules: the Item Linker and the Item Graph Analyzer [28].

Item Linker. The Item Linker addresses the task of linking the items in the system with the corresponding resources in the LOD knowledge bases. The aim of such component is bridging the gap between the items in the catalog and LOD. We have hypothesized two main ways for performing the linking task: Direct Item Linking and Item Description Linking. These modules take as input any dataset in the Linked Open Data cloud and the list of items in the catalog with associated side information, if available, and return either the mapping between items and URIs or the set of URIs found in each item description, depending on the selected module.

Direct Item Linking This approach is the more straightforward way for accessing LOD datasets. However, it requires that items have to be Linked Open Data resources, otherwise it cannot be used. Using movie title and year information it is possible to find the relative DBpedia resource. However, it is important to solve possible cases of ambiguity. The simplest solution is to exploit the class of the ontology that the item belongs to. For instance, in the movie domain we may select the resources with class dbo:Film.

Item Description Linking This approach bases on the exploitation of side information about the items such as textual descriptions or attributes. Such information can be used as input for entity linking tools in order to have access to LOD resources and link them to the item. Specifically, Entity Linking is the task of linking the entity mentioned in the text with the corresponding real world entity in the existing knowledge base [38]. Many Entity Linking tools have been proposed in the literature and made available on the Web. Some of them are: Babelfy [21], Dexter [7], DBpedia Spotlight [19], TAGME [14], NERD [33].

Item Graph Analyzer. This module is responsible of the extraction from the knowledge base of a descriptive and informative subgraph for each item, that is a set of RDF triples somehow related to the item resource. Eventually, all the extracted portions of the original graph can be merged to obtain a specific knowledge graph representative of the domain of interest covered by the recommender. It takes as input the list of items URI returned by the Item Linker and returns a set of RDF triples for each item. Potentially, each item resource may be connected to a big portion of the LOD graph. However, not all entities and relations may

be informative and descriptive of the item content. Several strategies to select a relevant subset of RDF triples for each item may be considered and adopted [22,31]. One strategy can be to manually define a set of properties or sequences of properties by using some domain knowledge. SPARQL queries are a powerful tool to pre-filter subgraphs relevant for the recommendation task.

Evaluating LOD-based RSs

There are many datasets available for the evaluation of recommender systems. However, such datasets are not appropriate for evaluating LOD-based recommendation algorithms because they do not contains links to URIs. In order to evaluate LOD-based RSs, three datasets belonging to different domains (movies, music and book) have been processed to compute a mapping between items (movies, artists, books) and their corresponding DBpedia URIs. The mappings for the datasets is available at https://github.com/sisinflab/LODrecsys-datasets.

Movielens. This dataset is based on the MovieLens 1M dataset (http://www.grouplens.org/node/73) released by the GroupLens research group. The original dataset contains 1,000,209 1-5 stars ratings given by 6,040 users to 3,883 movies. We found a valid mapping for 3,300 out of the all movies.

LibraryThing. Derived from the LibraryThing (http://www.librarything.com) dataset (http://www.macle.nl/tud/LT/). This dataset is related to the book domain and contains 7,112 users, 37,231 books and 626,000 ratings ranging from 1 to 10. In this case we found a match for 11,694 books.

LastFM. Differently from the previous ones this third dataset is based on implicit feedback consisting of user-artist listening data. Its data come from recent initiatives on information heterogeneity and fusion in recommender systems (http://ir.ii.uam.es/hetrec2011/datasets.html) [6] and has been built on top of the Last.fm music system (http://www.lastfm.com). The original dataset contains 1,892 users, 17,632 artists and 92,834 relations between a user and a listened artist together with their corresponding listening counts. For this dataset we found a match for 11,180 out of all artists.

Open Challenges

Recommender systems can be considered as a killer application for the exploitation of the huge knowledge encoded in LOD datasets freely available on the Web. Although, many solutions have been proposed and implemented to deliver this new generation of knowledge enabled recommendation engines (see [31,12,11,28,24,23,10,40,25,9,27,26,16,2,17,34,13]) some important issues remain still open to a deeper investigation. Among them we cite the most important ones: feature selection, distributed computing, cross-domain recommendation, computing explanation, the role of formal reasoning in the recommendation process.

feature selection The richness of `Linked Open Data` datasets may result in a pitfall for data-intensive tasks (as computing recommendations) as they potentially introduce noise in the data. Selecting the right features in `LOD`-enabled recommender systems results not just in getting the minimal meaningful subset of properties which are directly connected to an item but, given the graph-based nature of the underling data, the minimal meaningful set of semantic paths, of arbitrary length, which result representative of the item itself.

distributed computing Over the last years solutions to horizontally distribute graph-based data manipulation have been proposed also boosted by the increasing production of data coming from social networks. All the methods, algorithms and frameworks work quite well with multi-relational graphs where the number of possible relations are just a few compared to that of `Linked Open Data`. New approaches need to be proposed and developed to easily deal with all the semantics encoded in `LOD` datasets.

cross-domain recommendation The highly interconnected nature of datasets such as `DBpedia` or `Wikidata` represents an opportunity to develop cross-domain recommender systems. That is, systems able to recommend items in a knowledge domain which is not the same of the user profile. As an example, we may be able to recommend books given the user profile collects information on movies.

computing explanation Sometimes, receiving recommendations may result frustrating as we do not know the reason why the system suggested such items to us. Computing explanation for recommendations has been identified as a *must have* feature for the new generation of recommender systems. In this direction, all the knowledge available as `Linked Open Data` may surely play a key role.

formal reasoning `LOD` datasets are not just a mere collection of data represented in a graph-based way. They usually refer to a rich ontology which in turns can be represented by means of expressive logical languages as Description Logics. The adoption of such languages enable the application of formal logical reasoning over the underling data. As of today, due to its high computational complexity, such reasoning has not been exploited to its full potential but it can surely add new value to real knowledge-enabled recommender systems.

References

1. Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge Data Engineering*, 17(6):734–749, 2005.
2. Vito Walter Anelli, Tommaso Di Noia, Pasquale Lops, and Eugenio Di Sciascio. Feature factorization for top-n recommendation: From item rating to features relevance. In *Proceedings of the 1st Workshop on Intelligent Recommender Systems by Knowledge Transfer & Learning co-located with ACM Conference on Recommender Systems (RecSys 2017), Como, Italy, August 27, 2017.*, pages 16–21, 2017.

3. Marko Balabanović and Yoav Shoham. Fab: Content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, March 1997.
4. John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 43–52, 1998.
5. Robin D. Burke. Hybrid web recommender systems. In *The Adaptive Web, Methods and Strategies of Web Personalization*, pages 377–408, 2007.
6. Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In *Proceedings of the 5th ACM conference on Recommender systems*, RecSys 2011, New York, NY, USA, 2011. ACM.
7. Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. Dexter: An open source framework for entity linking. In *Proceedings of the Sixth International Workshop on Exploiting Semantic Annotations in Information Retrieval*, ESAIR '13, pages 17–20, New York, NY, USA, 2013. ACM.
8. Marco de Gemmis, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. *Semantics-Aware Content-Based Recommender Systems*, pages 119–159. Springer US, Boston, MA, 2015.
9. Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, and Davide Romito. Exploiting the web of data in model-based recommender systems. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 253–256, New York, NY, USA, 2012. ACM.
10. Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, and Markus Zanker. Linked open data to support content-based recommender systems. In *Proceedings of the 8th International Conference on Semantic Systems*, I-SEMANTICS '12, pages 1–8, New York, NY, USA, 2012. ACM.
11. Tommaso Di Noia, Vito Claudio Ostuni, Paolo Tomeo, and Eugenio Di Sciascio. Sprank: Semantic path-based ranking for top-n recommendations using linked open data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):9:1–9:34, 2016.
12. Tommaso Di Noia, Jessica Rosati, Paolo Tomeo, and Eugenio Di Sciascio. Adaptive multi-attribute diversity for recommender systems. *Information Sciences*, pages –, 2016.
13. Ignacio Fernández-Tobías, Paolo Tomeo, Iván Cantador, Tommaso Di Noia, and Eugenio Di Sciascio. Accuracy and diversity in cross-domain recommendations for cold-start users with positive-only feedback. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, pages 119–122, 2016.
14. Paolo Ferragina and Ugo Scaiella. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1625–1628, New York, NY, USA, 2010. ACM.
15. Rong Jin, Joyce Y. Chai, and Luo Si. An automatic weighting scheme for collaborative filtering. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 337–344, New York, NY, USA, 2004. ACM.
16. Aleksandra Karpus, Tommaso Di Noia, and Krzysztof Goczyla. Top k recommendations using contextual conditional preferences model. In *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017, Prague, Czech Republic, September 3-6, 2017.*, pages 19–28, 2017.

17. Aleksandra Karpus, Tommaso Di Noia, Paolo Tomeo, and Krzysztof Goczyla. Rating prediction with contextual conditional preferences. In *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2016) - Volume 1: KDIR, Porto - Portugal, November 9 - 11, 2016.*, pages 419–424, 2016.
18. Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195, 2015.
19. Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, pages 1–8, New York, NY, USA, 2011. ACM.
20. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc.
21. Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity Linking meets Word Sense Disambiguation : a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL)*, 2014.
22. Cataldo Musto, Pasquale Lops, Pierpaolo Basile, Marco de Gemmis, and Giovanni Semeraro. Semantics-aware graph-based recommender systems exploiting linked open data. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, UMAP '16, pages 229–237, New York, NY, USA, 2016. ACM.
23. Phuong Nguyen, Paolo Tomeo, Tommaso Di Noia, and Eugenio Di Sciascio. An evaluation of simrank and personalized pagerank to build a recommender system for the web of data. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 1477–1482, New York, NY, USA, 2015. ACM.
24. Phuong T. Nguyen, Paolo Tomeo, Tommaso Di Noia, and Eugenio Di Sciascio. *Content-Based Recommendations via DBpedia and Freebase: A Case Study in the Music Domain*, pages 605–621. Springer International Publishing, Cham, 2015.
25. Vito Claudio Ostuni, Tommaso Di Noia, Eugenio Di Sciascio, and Roberto Mirizzi. Top-n recommendations from implicit feedback leveraging linked open data. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 85–92, New York, NY, USA, 2013. ACM.
26. Vito Claudio Ostuni, Tommaso Di Noia, Roberto Mirizzi, and Eugenio Di Sciascio. A linked data recommender system using a neighborhood-based graph kernel. In *The 15th International Conference on Electronic Commerce and Web Technologies*, Lecture Notes in Business Information Processing. Springer-Verlag, 2014.
27. Vito Claudio Ostuni, Giosia Gentile, Tommaso Di Noia, Roberto Mirizzi, Davide Romito, and Eugenio Di Sciascio. Mobile movie recommendations with linked data. In *Human-Computer Interaction & Knowledge Discovery @ CD-ARES'13*, IFIP International Cross Domain Conference and Workshop on Availability, Reliability and Security, CD-ARES 2013. Springer, 2013.
28. Vito Claudio Ostuni, Sergio Oramas, Tommaso Di Noia, Xavier Serra, and Eugenio Di Sciascio. Sound and music recommendation with knowledge graphs. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(2):21:1–21:21, 2016.
29. Michael Pazzani and Daniel Billsus. Learning and revising user profiles: The identification ofinteresting web sites. *Mach. Learn.*, 27(3):313–331, June 1997.

30. Michael J. Pazzani and Daniel Billsus. The adaptive web. chapter Content-based Recommendation Systems, pages 325–341. Springer-Verlag, Berlin, Heidelberg, 2007.
31. Azzurra Ragone, Paolo Tomeo, Corrado Magarelli, Tommaso Di Noia, Matteo Palmonari, Andrea Maurino, and Eugenio Di Sciascio. Schema-summarization in linked-data-based feature selection for recommender systems. In *32nd ACM SIGAPP Symposium On Applied Computing*. ACM, 2017.
32. Francesco Ricci, Lior Rokach, and Bracha Shapira, editors. *Recommender Systems Handbook*. Springer, 2015.
33. Giuseppe Rizzo and Raphaël Troncy. Nerd: A framework for unifying named entity recognition and disambiguation extraction tools. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 73–76, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.
34. Jessica Rosati, Petar Ristoski, Tommaso Di Noia, Renato De Leone, and Heiko Paulheim. RDF graph embeddings for content-based recommender systems. In *Proceedings of the 3rd Workshop on New Trends in Content-Based Recommender Systems co-located with ACM Conference on Recommender Systems (RecSys 2016), Boston, MA, USA, September 16, 2016.*, pages 23–30, 2016.
35. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, 2001.
36. J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. The adaptive web. chapter Collaborative Filtering Recommender Systems, pages 291–324. Springer-Verlag, Berlin, Heidelberg, 2007.
37. Barry Schwartz. *The Paradox of Choice: Why More Is Less*. Harper Perennial, January 2005.
38. Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. Linden: Linking named entities with knowledge base via semantic knowledge. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 449–458, New York, NY, USA, 2012. ACM.
39. Cheri Speier, Joseph S. Valacich, and Iris Vessey. The Influence of Task Interruption on Individual Decision Making: An Information Overload Perspective. *Decision Sciences*, 30(2):337–360, March 1999.
40. Paolo Tomeo, Ignacio Fernández-Tobías, Tommaso Di Noia, and Iván Cantador. Exploiting linked open data in cold-start recommendations with positive-only feedback. In *Proceedings of the 4th Spanish Conference on Information Retrieval*, CERI '16, pages 11:1–11:8, New York, NY, USA, 2016. ACM.
41. Denny Vrandečić. Wikidata: A new platform for collaborative data collection. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, pages 1063–1064, New York, NY, USA, 2012. ACM.