Using Ontology-based Data Summarization to Develop Semantics-aware Recommender Systems*

Vito Walter Anelli[1], Tommaso Di Noia[1], Andrea Maurino[2], Matteo Palmonari[2], Anisa Rula[2]

[1] Polytechnic University of Bari, Via Orabona, 4, 70125 Bari, Italy
firstname.lastname@poliba.it
[2] University of Milano-Bicocca, Piazza dell'Ateneo Nuovo, 1, 20126 Milano, Italy
firstname.lastname@unimib.it

Abstract. In the current information-centric era, recommender systems are gaining momentum as tools able to assist users in daily decision-making tasks. Within the recommendation process, Linked Data have been already proposed as a valuable source of information to enhance the predictive power of recommender systems but an open issues is still related to feature selection of the most relevant subset of data in the whole semantic web. In this paper, we show how ontology-based (linked) data summarization can drive the selection of properties/features useful to a recommender system. In particular, we compare a fully automated feature selection method based on ontology-based data summaries with more classical ones, and we evaluate the performance of these methods in terms of accuracy and aggregate diversity of a recommender system exploiting the top-k selected features.

1 Introduction

Semantics-aware Recommender Systems (RSs) exploiting information held in knowledge graphs, as the ones available as Linked Data (LD), represent one of the most interesting and challenging application scenarios for LD [6]. A high number of solutions and tools have been proposed in the last years showing the effectiveness of adopting LD as knowledge sources to feed a recommendation engine (see [5] and references therein for an overview). Nevertheless, how to automatically select the "best" subset of a LD dataset to feed a LD-based RS without affecting the performance of the recommendation algorithm is still an open issue. Notice that the selection of the top-k features to use in a RSs means to discover which properties in a LD-dataset (e.g., DBpedia) encode the knowledge useful in the recommendation task and which ones are just noise [20]. In most of the approaches proposed so far, usually, the FS process is performed by human experts that manually choose properties resulting more "suitable" for a given scenario. Over the years, many algorithms and techniques for feature selection , e.g., Information Gain, Information Gain Ratio, Chi Squared Test and Principal Component Analysis, have been proposed with reference to machine learning tasks but they do not consider a characteristic which makes unique LD: they come with semantics attached.

The main objective of this paper is to investigate how ontology-based data summarization [28] can be used as a new and semantic-oriented feature selection technique for

* Discussion Paper. An extended version of this work has been presented at [7].
SEBD 2018, June 24-27, 2018, Castellaneta Marina, Italy.

LD-based RSs. We define a feature selection method that automatically extracts the top-k properties that are deemed to be more important to evaluate similarity between instances of a given class on top of data summaries built with the help of an ontology. We perform an experimental evaluation on three well-known datasets in the RS domain (*Movielens*, *LastFM*, *LibraryThing*) in order to analyze how the choice of a particular FS technique may influence the performance of recommendation algorithms in terms of typical accuracy and diversity metrics. Experimental results show that information provided in ontology-based data summaries selects features that achieve comparable, or, in most of the cases, better performance than state-of-the-art, semantic-agnostic analytical methods such as Information Gain [19]. We believe that these results are interesting also because of practical reasons. LD summaries are published on-line and summary-based FS can be performed even without acquiring the entire dataset and efficiently (on top of summary information). The use of frequency associated with schema patterns in a FS approach was initially tested in a previous work[24].

The paper is organized as follows: in Section 2, we introduce the ontology-based data summarization approach used in this work, while in Section 3, we describe the feature selection and recommendation methods. Section 4 is devoted to the explanation and discussion of the experimental results. Section 5 briefly reviews related literature for schema and data summarization as well as on recommender systems while Section 6 discuss conclusions and future work.

2 Ontology-driven Linked Data Summarization

While relevance-oriented data summarization approaches are aimed at finding subsets of a dataset or an ontology that are estimated to be more relevant for the users [30], vocabulary-oriented approaches are aimed at profiling a dataset, by describing the usage of vocabularies/ontologies used in the dataset. The summaries returned by these approaches are complete, i.e., they provide statistics about every element of the vocabulary/ontology used in the dataset [28]. Statistics captured by these summaries that can be useful for the feature selection process are the ones concerning the usage of properties for a certain class of items to recommend.

Patterns and frequency. In our approach we use pattern-based summaries extracted using the ABSTAT framework. Pattern-based summaries describe the content of a dataset using schema patterns having the form $\langle C, P, D\rangle$, where C and D, are types (either classes or datatypes) and P is a property. For example, the pattern $\langle \texttt{dbo:Film}, \texttt{dbo:starring}, \texttt{dbo:Actor}\rangle$ tells that films exist in the dataset, in which star some actors. Differently from similar pattern-based summaries [17], ABSTAT uses the subclass relations in the data ontology, represented in a *Type Graph*, to extract only *minimal type patterns* from relational assertions, i.e, the patterns that are more type-wise specific according to the ontology. A pattern $\langle C, P, D\rangle$ is a minimal type pattern for a relational assertion $\langle a, P, b\rangle$ according to a type graph $\mathcal{G}$ iff C and D are the types of a and b respectively, which are minimal in $\mathcal{G}$. In a pattern $\langle C, P, D\rangle$, C and D are referred to as *source* and *target* types respectively. A minimal type pattern $\langle \texttt{dbo:Film}, \texttt{dbo:starring}, \texttt{dbo:Actor}\rangle$ (simply referred to as *pattern* in the following) tells that there exist entities that have `dbo:Film` and `dbo:Actor` as minimal

ABSTAT summaries for several datasets can be explored at `http://abstat.disco.unimib.it`

If no ontology is specified, all types are minimal and patterns are extracted like in frameworks that do not adopt minimalization types which are connected through the property P. Non minimal patterns can be inferred from minimal patterns and the type graph. Therefore, they can be excluded as redundant without information loss, making summaries more compact [28]. Each pattern $\langle C, P, D\rangle$ is associated with a *frequency*, which reports the number of relational assertions $\langle a, P, b\rangle$ from which the pattern has been extracted.

Local cardinality descriptors. For this work, we have extended ABSTAT to extract *local cardinality descriptors*, i.e., cardinality descriptors of RDF properties, which are specific to the patterns in which the properties occur. To define these descriptors, we first introduce the concept of restricted property extensions. The *extension* of a property P restricted to a pattern $\langle C, P, D\rangle$ is the set of pairs $\langle x, y\rangle$ such that the relational assertion $\langle x, P, y\rangle$ is part of the dataset and $\langle C, P, D\rangle$ is a minimal-type pattern for $\langle x, P, y\rangle$. Given a pattern π with a property P, we can define the functions (that return the closest integer):

$minS(\pi)$, $maxS(\pi)$, $avgS(\pi)$: denoting respectively the minimum, maximum and average number of distinct subjects associated to unique objects in the extension of P restricted to π;

$minO(\pi)$, $maxO(\pi)$, $avgO(\pi)$: denoting respectively the minimum, maximum and average number of distinct objects associated to unique subjects in the extension of P restricted to π.

ABSTAT can also compute global cardinality descriptors by adjusting the above mentioned definition so as to consider unrestricted property extensions. Local cardinality descriptors carry information about the semantics of properties as used with specific types of resources (in specific patterns) and can be helpful for selecting features used to compute the similarity between resources. For example, to compute similarity for movies, one would like to discard properties that occur in patterns π with `dbo:Film` as source type and $avgS(\pi) = 1$. We remark that the values of local cardinality descriptors for patterns with a property P may differ from values of global cardinality descriptors for P. Some examples of local cardinality descriptors can be found in the faceted-search interface (ABSTATBrowse). In conclusion, ABSTAT takes a linked dataset and - if specified - one or more ontologies as input, and returns a summary that consists of: a type graph, a set of patterns, their frequency, local an global cardinality descriptors.

3 Semantics-aware Feature Selection

Feature selection is the process of selecting a subset of relevant attributes in a dataset. Thanks to the feature selection process it is possible to improve the prediction performance, and to give a better understanding of the process that generates the data [12]. There are three typical measures of feature selection (i) "**filters**", statistical measures to assign a score to each feature (here the feature selection process is a preprocessing step and can be independent from learning[11]); (ii) "**wrapper**" where the learning system is used as a black box to score subsets of features [14]; (iii) **embedded methods** that perform the selection within the process of training [12]. In the following, we discuss two approaches used for the feature selection task: the first operates on the summarization of the datasets and the second operates on the instances of the datasets.

Feature Selection with Ontology-based Summaries. As described in Section 2, the ABSTAT framework provides two useful statistics: the pattern frequency and the cardinality descriptors that are used in the feature selection process as described in Figure 1. The

http://abstat.disco.unimib.it/browse

process starts by considering all patterns $\Pi = \{\pi_1, \pi_2, \ldots, \pi_n\}$ of a given class C occurring as a source type. The example in Figure 1 shows a subset of Π with `dbo:Film` as source type. The first step of our approach (*FILTERBY*) filters out properties based on the local cardinality descriptors. In particular, it filters only properties for which the average number of distinct subjects associated with unique objects is more than one ($avgS > 1$). The second step of the process (*SELECTDISTINCTP*) selects all properties of the patterns in Π by applying the maximum of the pattern frequency ($\#$ in the Figure). Then, the properties are ranked (*ORDERBY*) in a descending order on pattern frequency and then k properties (*TOPK*) are selected (k=2).

In some datasets, such as DBpedia, properties may use redundant information by using same properties with different namespaces, e.g., `dbo:starring` and `dbp:starring`. For this reason, in such case, a pre-processing step for removing replicated properties to avoid redundant ones is requested (see Section 4).

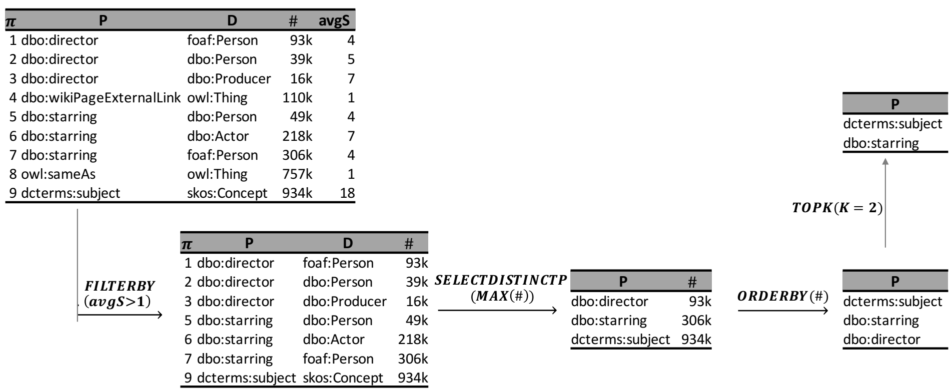

Fig. 1: Feature selection with ABSTAT with source type `dbo:Film`

Feature Selection with State-of-the-art Techniques. In this work we consider RDF properties as features, so among the different feature selection techniques available in the literature, we initially selected *Information Gain*, *Information Gain Ratio*, *Chi-squared test* and *Principal Component Analysis* as their computation can be adapted to categorical features as LD and we then evaluated their effect over the recommendation results. The features selected from each technique have been used as an input of the recommendation algorithm that uses the Jaccard index as similarity measure. In order to identify the best technique among the one we selected, they have been evaluated by using Information Gain (IG), Gain Ration and Chi Squared Test. At the end, IG[19] resulted as the best performing one. Then features are ranked according to their IG value and the top-k ones are returned.

Feature pre-processing. LD datasets usually have a quite large feature set that can be very sparse depending on the knowledge domain. For instance, taking into account the movies available in *Movielens*, properties as `dbp:artDirection` or `dbp:precededBy` are very specific and have a lot of missing values. On the other hand, properties as `dbo:wikiPageExternalLink` or `owl:sameAs` always have different and unique values, so they are not informative for a recommendation task.

For the sake of conciseness we do not report all the results here. Results obtained with other FS techniques can be found at `http://ow.ly/zAA530d0wu0`

For this reason, before starting the feature selection process with IG, we reduced *redundant* or *irrelevant* features. The pre-processing step has been done following [23]: we fixed a threshold $t_m = t_d = 97\%$ both for missing values and for distinct values and, then, we discarded features for which we had more than t_m of missing values and more than t_d of distinct values.

Recommendation Method. We implemented a content-based recommender system using an item-based nearest neighbors algorithm as in [20], where the similarity is computed by means of Jaccard's index (widely adopted for categorical features). In this work, the neighborhood of a resource includes all the nodes in the graph reachable starting from the resource i (respectively j) following the properties selected by the feature selection phase. The neighbors are thus one-hop features. The similarity values are then used to recommend to each user the items which result most similar to the ones she has liked in the past. Ratings are predicted as a normalized sum of neighbors ratings, weighted by their similarity values [21].

4 Experimental Evaluation

Datasets. The evaluation has been carried out on the three well-known datasets belonging to different domains, i.e. movies(Movielens 1M), books (LibraryThing), and music (`Last.fm`). The datasets contains, respectively, 1,000,209, 626,000 and 92,834 ratings. Movielens 1M and LibraryThing provide explicit ratings over 1-5 and 1-10 scales whereas `Last.fm` [3] provides users listening counts.

Measures. For evaluating the quality of our recommendation algorithm we are interested in measuring its performances in terms of accuracy of the predicted results and diversity. To evaluate recommendation **accuracy**, we used *Precision* (Precision@N) and *Mean Reciprocal Rank* (MRR). Precision@N is a metric denoting the fraction of relevant items in the Top-N recommendations. MRR computes the average reciprocal rank of the first relevant recommended item [26]. A good recommender system should provide recommendations equally distributed among the items, otherwise, even if accurate, they indicate a low degree of personalization [1]. To evaluate **aggregate diversity**, we considered *catalog coverage* (the percentage of recommended items in the catalog) and *aggregate entropy* [1]. Please note that here we are considering a global diversity rather than a personalized one [8].

Implementation. We tried different ranking and filtering functions to study their effect on feature selection. For lack of space we include the best combination from those proposed in section 3, **AbsOccAvgS**, that considers as input of *FILTERBY* the $avgS$ and *SELECTDISTINCTP* the maximum of the pattern frequency. Both for ABSTAT and IG we considered both the **Onlydbp** configuration in which, if among the first N features selected there are both `dbo:` and `dbp:` feature we consider only the `dbp:` one. Conversely in **Onlydbo** we take into account only the `dbo:` one.

ABSTAT Baseline. As a baseline for ABSTAT-based feature selection we use TF-IDF as is a well-known measure to identify most relevant terms (properties in this case) for a document (a class in this case). We adopt *TF-IDF* in our context where by document we refer to a set of patterns having the same subject-type and by term we refer to a property. *TF-IDF* is based on the number of properties occurring in a document (*TF*) and the logarithm of the ratio between the total number of documents and the ones containing the property (*IDF*).

`http://ir.ii.uam.es/hetrec2011/datasets.html`

	Precision@10		**MRR@10**		**catalogCoverage@10**		**aggrEntropy@10**	
Top K features	**5**	**20**	**5**	**20**	**5**	**20**	**5**	**20**
dbo.IG	.0841	.1076	.2961	.3390	.3372	.5226	7.94	8.44
dbo.AbsOccAvgS	**.1066**	.1067	**.3388**	.3402	.5344	.5208	8.68	8.51
dbo.TfIdf	.0823	.0856	.2994	.3123	.3520	.3908	7.83	7.99
dbp.IG	.0688	.1046	.2134	.3336	.2799	.5065	6.54	8.31
dbp.AbsOccAvgS	**.1065**	**.1059**	.3408	.3360	.5426	.5105	8.64	8.31
dbp.TfIdf	.0549	.0745	.1924	.2687	.2530	.3575	6.33	7.41

Table 1: Experimental results on the Movielens dataset.

	Precision@10		**MRR@10**		**catalogCoverage@10**		**aggrEntropy@10**	
Top K features	**5**	**20**	**5**	**20**	**5**	**20**	**5**	**20**
dbo.IG	.0571	.0579	.2346	.2274	.3988	.4037	10.47	10.44
dbo.AbsOccAvgS	.0561	.0593	.2328	.2329	.3982	.4030	10.54	10.48
dbo.TfIdf	.0579	.0605	.2374	.2477	.4086	.3991	10.55	10.20
dbp.IG	.0586	.0586	.2350	.2299	.4027	.4043	10.49	10.40
dbp.AbsOccAvgS	.0623	.0612	.2467	.2342	.3943	.4043	10.42	10.45
dbp.TfIdf	.0215	.0132	.1608	.1218	.1314	.2696	8.81	9.96

Table 2: Experimental results on the LastFM dataset.

	Precision@10		**MRR@10**		**catalogCoverage@10**		**aggrEntropy@10**	
Top K features	**5**	**20**	**5**	**20**	**5**	**20**	**5**	**20**
dbo.IG	.0411	.1319	.1989	.4083	.4425	.5053	11.06	11.20
dbo.AbsOccAvgS	.1283	.1292	.3986	.4063	.4915	.4949	11.14	11.14
dbo.TfIdf	.1024	.1132	.3064	.3554	.4026	.4508	10.76	10.96
dbp.IG	.0678	.1319	.2553	.4083	.4364	.5053	10.83	11.20
dbp.AbsOccAvgS	**.1319**	**.1316**	**.4026**	**.4113**	.4926	.5055	11.14	11.20
dbp.TfIdf	.0790	.1170	.2371	.3572	.3894	.4698	10.69	11.04

Table 3: Experimental results on the LibraryThing dataset.

Results Tables 1, 2, 3 show the experimental results obtained on, respectively, *Movielens*, *Last.FM* and *LibraryThing* datasets in terms of **Precision**, **MRR**, **catalogCoverage**, and **aggrEntropy**. Results are computed over lists of top-10 items recommended by the RS. We conducted experiments using top-k selected features for different k, i.e., $k = 5, 10, 15, 20$, and all configurations but we report only results for $k = 5, 20$ and best configurations . We highlight in bold only the values for which there is a statistical significant difference. For *Lastfm* dataset the differences are not statistical significant so the two methods are equivalent in selecting features.

Discussion. As an overall result, ABSTAT-based FS leads to the best results in terms of accuracy and diversity for both the movie and books domains while IG leads to better results (although not statistically significant) for music.

Specifically, considering the results on Movielens (Table 1), ABSTAT produces better accuracy with respect to IG in all the configurations both with 5 and 20 features. In terms of aggregate diversity, i.e. itemCoverage and aggrEntropy, ABSTAT is still the best choice, overcoming IG in almost all the situations. On Lastfm (Table 2) there are no particular differences, and hence the choice of the method seems irrelevant: both summarization-based and statistical methods are comparable. Eventually, on LibraryThing (Table 3), ABSTAT strongly beats IG in almost all the configurations. In particular, it gets more than twice of the precision and MRR respect to IG in top-5 features scenario. Summing up, ABSTAT beats IG in almost all the configurations on the two datasets Movielens and LibraryThing, while they act in the same way on the Lastfm dataset.

In order to investigate the reasons behind the different behaviors depending on the selected knowledge domain, we measured: (i) the number of minimal patterns and (ii) the average number of triples per resource and the corresponding variance. Regarding the former we may say that a higher number of minimal patterns means a richer and more diverse

The interested reader can find results for all values of k and configurations on GitHub: `http://ow.ly/zAA530d0wu0` ontological representation of the knowledge domain. As for the latter, a high variance in the number of triples associated to resources is a clue of an unbalanced representation of the items to recommend. Hence, items with a higher number of triples associated result "more popular" in the knowledge graph compared to those with only a few. This may reflect in the rising of a stronger content popularity bias while computing the recommendation results. If we look at the values represented in Table 4 we may assert that a higher

Domain	Number of Minimal Patterns	Average Number of Triples	Variance
Movies	57757	74,015	549,313
Books	41684	44,966	169,478
Music	40481	80,502	981,509

Table 4: Ontological and data dimensions of the three datasets

sparsity in the knowledge graph data may give chance to statistical methods to beat ontological ones. In other words, it seems that the higher the sparsity of the knowledge graph at the data level, the lower the influence of the ontological schema in the selection of the most informative features to build a pure content-based recommendation engine.

5 Related Work

Summarization. Different approaches have been proposed for schema and data summarization[28]. Several data profiling approaches are aimed to describe linked data by reporting statistics about the usage of the vocabularies, types and properties. SchemeEx extracts interesting measures , by considering the co-occurrence of types and properties [15]. Linked Open Vocabularies, RDFStats [16] and LODStats [2] provide such statistics. In contrast, ABSTAT represents connections between types using schema patterns, for which it also provides cardinality descriptors. does not include cardinality descriptors for properties or patterns. TermPicker extracts [25] patterns consisting in triples $\langle S, E, O\rangle$, where S and O are *sets* of types and E is a set of predicates. Instead, ABSTAT and Loupe extract patterns each consisting in a triple $\langle C, P, D\rangle$ where C and D are types and P a property. TermPicker summaries do not describe cardinality and are extracted from RDF data without considering relationships between types. According to [18], which proposes a method to define and discover classes of cardinality constraints with some preliminary results, current approaches focus only on mining keys or pseudo-keys (e.g., [27]). We discover richer statistics about property cardinality like the above mentioned work, and in addition we compute cardinality descriptors for properties occurring in specific schema patterns.

Recommender Systems. One of the first approaches for using LD in a recommender system was proposed by Heitmann and Hayes[13]. A system for recommending artists and music using DBpedia was presented in [22]. The task of cross-domain recommendation leveraging DBpedia as a knowledge-based framework was addressed in [10], while in [29] the authors present a semantics-aware approach to deal with cold-start situations. In [9] the authors use a a hybrid graph-based algorithm built upon DBpedia and collaborative information. To the best of our knowledge, the only approaches proposing an automatic selection of LD features are [19, 24]. Finally, we observe that even approaches that do not perform automatic FS like [19, 4, 9] used (hand-crafted) FS to improve their performance.

6 Conclusions

In this work we investigated the role of ontology-based data summarization for feature selection in recommendation tasks. Here we compare results coming from ABSTAT, a

http://lov.okfn.org/

schema summarization tool, with classical methods for feature selection and we show that the former are allowed to compute better predictions not just in terms of precision of the recommended items but also considering other dimensions such as diversity. Experiments have been carried out in three different knowledge domains thus showing the effectiveness of a feature selection based on schema summarization over classical techniques such as Information Gain.

References

1. G. Adomavicius and Y. Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Trans. on Knowl. and Data Eng.*, 24(5), 2012.
2. S. Auer, J. Demter, M. Martin, and J. Lehmann. LODStats - An Extensible Framework for High-Performance Dataset Analytics. In *EKAW (2)*, 2012.
3. I. Cantador, P. Brusilovsky, and T. Kuflik. 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In *RecSys*. ACM, 2011.
4. M. de Gemmis, P. Lops, C. Musto, F. Narducci, and G. Semeraro. Semantics-aware content-based recommender systems. In *Recommender Systems Handbook*. 2015.
5. T. Di Noia. Knowledge-enabled recommender systems: Models, challenges, solutions. In *Proceedings of the 3rd International Workshop on Knowledge Discovery on the WEB, Cagliari, Italy, September 11-12, 2017.*, 2017.
6. T. Di Noia, I. Cantador, and V. C. Ostuni. Linked open data-enabled recommender systems: ESWC 2014 challenge on book recommendation. In *Semantic Web Evaluation Challenge - SemWebEval 2014 at ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers*, pages 129–143, 2014.
7. T. Di Noia, C. Magarelli, A. Maurino, M. Palmonari, and A. Rula. Using ontology-based data summarization to develop semantics-aware recommender systems. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, pages 128–144, 2018.
8. T. Di Noia, V. Ostuni, J. Rosati, P. Tomeo, and E. Di Sciascio. An analysis of users' propensity toward diversity in recommendations. In *RecSys 2014 - Proceedings of the 8th ACM Conference on Recommender Systems*, pages 285–288. Association for Computing Machinery, Inc, 2014.
9. T. Di Noia, V. C. Ostuni, P. Tomeo, and E. Di Sciascio. Sprank: Semantic path-based ranking for top-*N* recommendations using linked open data. *ACM TIST*, 8(1):9:1–9:34, 2016.
10. I. Fernández-Tobías, I. Cantador, M. Kaminskas, and F. Ricci. A generic semantic-based framework for cross-domain recommendation. In *2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems*, 2011.
11. X. Geng, T.-Y. Liu, T. Qin, and H. Li. Feature selection for ranking. In *SIGIR*. ACM, 2007.
12. I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3, 2003.
13. B. Heitmann and C. Hayes. Using linked data to build open, collaborative recommender systems. In *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence*, 2010.
14. R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2), 1997.
15. M. Konrath, T. Gottron, S. Staab, and A. Scherp. Schemex - efficient construction of a data catalogue by stream-based indexing of linked data. *Web Semant.*, 16, 2012.
16. A. Langegger and W. Wöß. RDFStats - an extensible RDF statistics generator and library. In *DEXA Workshops*, pages 79–83. IEEE, 2009.
17. N. Mihindukulasooriya, M. Poveda Villalon, R. Garcia-Castro, and A. Gomez-Perez. Loupe - An Online Tool for Inspecting Datasets in the Linked Data Cloud. In *ISWC Posters & Demonstrations*, 2015.
18. E. Muñoz. On learnability of constraints from RDF data. In *ESWC*. Springer, 2016.
19. C. Musto, P. Lops, P. Basile, M. de Gemmis, and G. Semeraro. Semantics-aware graph-based recommender systems exploiting linked open data. In *UMAP*, 2016.
20. P. Nguyen, P. Tomeo, T. Di Noia, and E. Di Sciascio. An evaluation of SimRank and Personalized PageRank to build a recommender system for the web of data. In *WWW '15 Companion*, pages 1477–1482. ACM, 2015.
21. X. Ning, C. Desrosiers, and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender Systems Handbook*, pages 37–76. 2015.
22. A. Passant. Dbrec: Music Recommendations Using DBpedia. In *9th ISWC*, pages 209–224. Springer-Verlag, 2010.
23. H. Paulheim and J. Fümkranz. Unsupervised generation of data mining features from linked open data. In *WIMS*, 2012.
24. A. Ragone, P. Tomeo, C. Magarelli, T. Di Noia, M. Palmonari, A. Maurino, and E. Di Sciascio. Schema-summarization in linked-data-based feature selection for recommender systems. In *Proceedings of the Symposium on Applied Computing, SAC 2017, Marrakech, Morocco, April 3-7, 2017*, pages 330–335, 2017.
25. J. Schaible, T. Gottron, and A. Scherp. Termpicker: Enabling the reuse of vocabulary terms by exploiting data from the linked open data cloud. In *The Semantic Web. Latest Advances and New Domains - 13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Proceedings*, pages 101–117, 2016.
26. Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic. CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering. In *RecSys*. ACM, 2012.
27. T. Soru, E. Marx, and A. N. Ngomo. ROCKER: A refinement operator for key discovery. In *WWW*. ACM, 2015.
28. B. Spahiu, R. Porrini, M. Palmonari, A. Rula, and A. Maurino. ABSTAT: ontology-driven linked data summaries with pattern minimalization. In *2nd Workshop on Summarizing and Presenting Entities and Ontologies, co-located with ESWC.*, 2016.
29. P. Tomeo, I. Fernández-Tobías, I. Cantador, and T. Di Noia. Addressing the cold start with positive-only feedback through semantic-based recommendations. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 25(Supplement-2):57–78, 2017.
30. G. Troullinou, H. Kondylakis, E. Daskalaki, and D. Plexousakis. RDF Digest: Efficient Summarization of RDF/S KBs. In *ESWC*, 2015.