

Tuning Personalized PageRank for Semantics-Aware Recommendations Based on Linked Open Data

Cataldo Musto^(✉), Giovanni Semeraro, Marco de Gemmis, and Pasquale Lops

Department of Computer Science, University of Bari Aldo Moro, Bari, Italy
`{cataldo.musto,giovanni.semeraro,marco.degummis,pasquale.lops}@uniba.it`

Abstract. In this article we investigate how the knowledge available in the Linked Open Data cloud (LOD) can be exploited to improve the effectiveness of a semantics-aware graph-based recommendation framework based on Personalized PageRank (PPR).

In our approach we extended the classic *bipartite* data model, in which only user-item connections are modeled, by injecting the *exogenous* knowledge about the items which is available in the LOD cloud. Our approach works in two steps: first, all the available items are automatically mapped to a **DBpedia** node; next, the resources gathered from **DBpedia** that describe the item are connected to the item nodes, thus enriching the original representation and giving rise to a *tripartite* data model. Such a data model can be exploited to provide users with recommendations by running PPR against the resulting representation and by suggesting the items with the highest PageRank score.

In the experimental evaluation we showed that our semantics-aware recommendation framework exploiting **DBpedia** and PPR can overcome the performance of several state-of-the-art approaches. Moreover, a proper tuning of PPR parameters, obtained by better distributing the weights among the nodes modeled in the graph, further improved the overall accuracy of the framework and confirmed the effectiveness of our strategy.

Keywords: Graphs · Recommender systems · Linked open data · PageRank

1 Introduction

According to recent statistics¹, 150 billions of RDF² triples and almost 10.000 linked datasets are now available in the so-called Linked Open Data (LOD) cloud [6]. The *nucleus* of this set of interconnected semantic datasets is represented by **DBpedia** [1], the RDF mapping of Wikipedia that acts as a huge *hub* for most of the RDF triples made available in the LOD cloud.

¹ <http://stats.lod2.eu/>.

² <http://www.w3.org/TR/rdf-concepts/>.

The collaborative effort behind the LOD initiative is bringing several benefits: first, the rise of these *data silos* is contributing to push forward the vision of the *Web of Data*; second, the tremendous growth of semantically-annotated data is leading practitioners and researchers to investigate *whether* and *how* data gathered from the LOD cloud can be exploited to develop new services or to improve the performance of intelligent and knowledge-intensive applications.

As an example, graph-based recommender systems [13] can benefit of the information available in the LOD cloud. In this case the classic bipartite user-item representation can be easily extended by injecting in the graph the resources available in the LOD cloud which are connected to the properties describing the item. This can help to discover surprising connections between the movies: as an example, by mining the information available in the LOD cloud it emerges that both *The Matrix* and *The Lost World: Jurassic Park* movies have been shot in Australia, and this can in turn help to generate better (and unexpected) recommendations by exploiting such new information (Fig. 1).

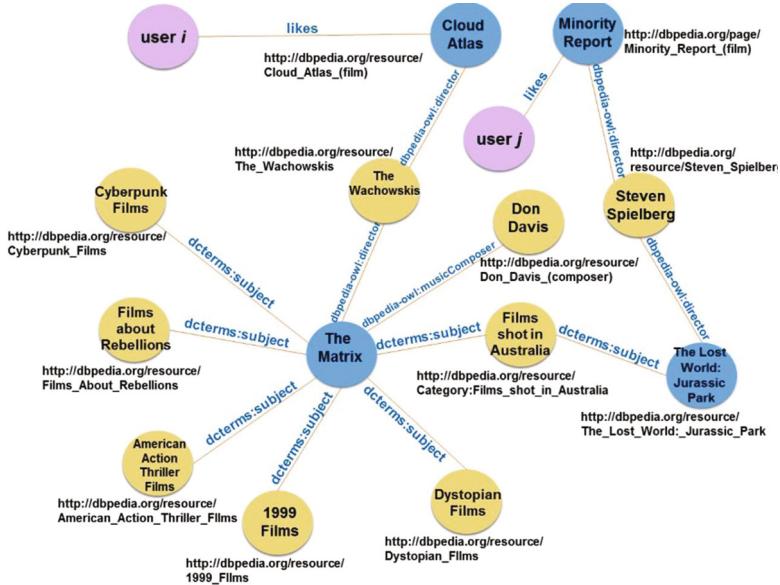


Fig. 1. A (tiny) portion of the connections between users, items and entities encoded in Linked Open Data cloud. Purple nodes represent users, blue nodes represent items, yellow nodes represent entities. (Color figure online)

According to these insights, it immediately emerges that recommender systems (RSs) may tremendously benefit of the data points available in the LOD cloud. To this end, in this article we investigate the impact of such *exogenous knowledge* on the performance of a graph-based recommendation framework. We focused our attention on graph-based approaches since they use a uniform formalism to represent both collaborative and LOD-based features. Indeed, in the

first case users and items are represented as *nodes* and preferences are represented as *edges*. Similarly, entities from the DBpedia are represented as *nodes* while the connections between them (expressed through RDF properties) are represented as *edges*. It is very straightforward that through a simple mapping of the items to be recommended with the URIs available in the LOD cloud, both representations can be connected and merged in a unique and powerful formalism. Given such a representation, we adopted Personalized Page Rank (PPR) [11] as recommendation algorithm and we suggested the items with the highest PageRank score. Specifically, we biased PageRank by modifying the personalization vector p (as shown in [11]) through the adoption of different combinations to distribute the weights.

In the experimental evaluation we showed that our approach is able to overcome the predictive accuracy obtained by several state of the art baselines. Moreover, in this work we made one step further with respect to our previous research [14], and we investigated how the distribution of the weights in PPR algorithm can influence the overall performance of the system. The results emerged from the experiments showed that a proper tuning of the weights, which also considers the nodes coming from the LOD cloud, may further improve the accuracy of the framework. To sum up, with this article we provide the following contributions:

1. We propose a methodology to feed a graph-based recommendation algorithm with features gathered from DBpedia;
2. We investigate whether different weights of the features gathered from DBpedia influence the recommendation accuracy;
3. We validated our methodology through several experiments, and we showed that our approach obtains higher results with respect to all the baselines we took into account.

The rest of the paper is organized as follows: Sect. 2 analyzes the related literature. The description of the graph-based recommender system and the overview the different methodologies for distributing weights in PPR are provided in Sect. 3. The details of the experimental evaluation on two state of the art datasets are described in Sect. 4, while conclusions and future work are drawn in Sect. 5.

2 Related Work

This work investigates two different research lines: graph-based recommender systems and LOD-based recommender systems. In the following we present the current literature in both areas.

Graph-based Recommender Systems. Most of the literature in the area of graph-based RS is inspired by PageRank [18] and random walk. As an example, FolkRank [12] is an adaptation of PageRank used for tag recommendation, that relies on a representation in which resources are modeled along with the tags the community used to annotate them. Next [7], Bogers proposed ContextWalk,

a movie recommender system relying on PageRank, modeling in a graph tags, genres and actors. Similarly, Baluja et al. [2] present a recommender system for YouTube videos based on random walks on the bipartite user-video graph. Finally, de Gemmis et al. [10] recently evaluated the applicability of Random Walk with Restart to Linked Data and investigated to what extent such graph-based representation can lead to *serendipitous* recommendations. A distinguishing aspect of this article with respect to the current literature is that none of the previously mentioned approaches investigated the integration of LOD-based data points. A similar attempt has been presented in [17], in which the paths connecting users and items via LOD-based properties are extracted and are used to train a classifier in a pure machine learning-based approach. Differently from this work, we encoded LOD-based features along with collaborative ones in a hybrid graph-based representation and we exploited Personalized PageRank as recommendation technique. Moreover, as previously introduced, in this article we continued carrying on our previous research [14] by better investigating how the distribution of the weights in PPR algorithm can influence the overall performance of the system.

LOD-based Recommender Systems. An updated and detailed review of the literature on recommendation approaches leveraging Linked Open Data is presented in [9]. In that survey, those approaches are classified as *top-down semantic approaches*, i.e. relying on the integration of external knowledge sources. In most of the current literature, properties gathered from DBpedia are exploited to define new similarity measures, as in [19]. The use of DBpedia for similarity calculation is also the core of the work presented by Musto et al. [15]: in that paper music preferences are extracted from Facebook and similarity measures are exploited to build a personalized music playlist. Moreover, a relevant research line investigated Linked Open Data to generate new descriptive features for the items. This is done in [8], where the authors exploit DBpedia to gather one or more labels describing the genre played by each artist the user liked, and by Baumann et al. [5], who extract features from Freebase³ to describe artists. The use of Freebase as knowledge base to feed recommendation algorithms is also investigated by Nguyen et al. [16], who assessed the effectiveness of such a source for music recommendations. Another interesting attempt is due to Basile et al. [3, 4] who obtained the best results in the ESWC 2014 Recommender Systems Challenge⁴ by proposing an ensemble of several widespread algorithms running on diverse sets of features gathered from the LOD cloud. Differently from the current literature, our work aim to exploit features gathered from the Linked Open Data to build a *hybrid model of user preferences* that merges both collaborative and LOD-based data points in a unique graph-based representation which exploits PPR as recommendation algorithm.

³ <https://www.freebase.com/>.

⁴ <http://challenges.2014.eswc-conferences.org/index.php/RecSys>.

3 Methodology

In this section we describe our graph-based recommendation methodology. First, we show how we extended the original *bipartite representation* based on user-item connections by introducing features gathered from DBpedia, next we provide some basics of PPR algorithm and we introduce the different approaches for distributing the weight in the tripartite LOD-enriched representation.

3.1 Graph-Based Representations

The main idea behind our graph-based model is to represent *users* and *items* as *nodes* in a graph. Formally, given a set of users $U = \{u_1, u_2, \dots, u_n\}$ and a set of items $I = \{i_1, i_2, \dots, i_m\}$, a graph $G = \langle V, E \rangle$ is instantiated. It is worth to note that G is a bipartite graph, since it models two different kind of entities (that is to say, *users* and *items*). Next, an edge connecting a user u_i to an item i_j is created for each positive feedback expressed by that user ($\text{likes}(u_i, i_j)$), thus $E = \{(u_i, i_j) | \text{likes}(u_i, i_j) = \text{true}\}$. Clearly, if each user and each item have at least a *positive* rating, then $|V| = |U| + |I|$. This is a very basic formulation, built on the ground of simple *collaborative* data points, since we just modeled user-item couples, as in collaborative filtering algorithms.

As previously explained, in this work we enriched this basic graph by introducing some *extra* nodes and *extra* edges, by exploiting the data points available in the LOD cloud. However, before performing this *enriching* process, it is mandatory to carry out a *mapping* step. The goal of the mapping procedure is to identify, for each item in the dataset, the corresponding DBpedia node the item refers to. As an example, we associate the book *The Shining* with its corresponding URI in the LOD cloud⁵. It is worth to emphasize that the mapping is a necessary and mandatory step to get an *entry point* to the LOD cloud. Once the mapping is completed, it is possible to gather all the extra features describing our items and to model our *tripartite graph* accordingly. Further details about the mapping procedure will be provided in Sect. 4.

Formally, after the mapping process we define an extended graph $G_{LOD} = \langle V_{ALL}, E_{ALL} \rangle$, where $V_{ALL} = V \cup V_{LOD}$ and $E_{ALL} = E \cup E_{LOD}$. In this case, E_{LOD} is the set of the new connections resulting from the properties encoded in the LOD cloud (e.g. *writer*, *subject*, *genre*, etc.), while V_{LOD} is the new set of nodes representing the resources gathered from the LOD cloud (e.g. *Stephen_King* or *Gothic_Novels*) that are connected to the items $i_1 \dots i_m \in I$.

As previously stated, G_{LOD} is a tripartite graph, since beyond users and items also the resources gathered from the LOD cloud describing the items are now modeled. However, one could argue that some of the properties used to gather new resources from the LOD cloud may not be relevant for the recommendation task, and should be filtered out from the representation. This issue can be solved by applying *features selection* (FS) techniques on the complete *tripartite* representation and to maintain in the graph only those labeled as

⁵ [http://dbpedia.org/resource/The_Shining_\(film\)](http://dbpedia.org/resource/The_Shining_(film)).

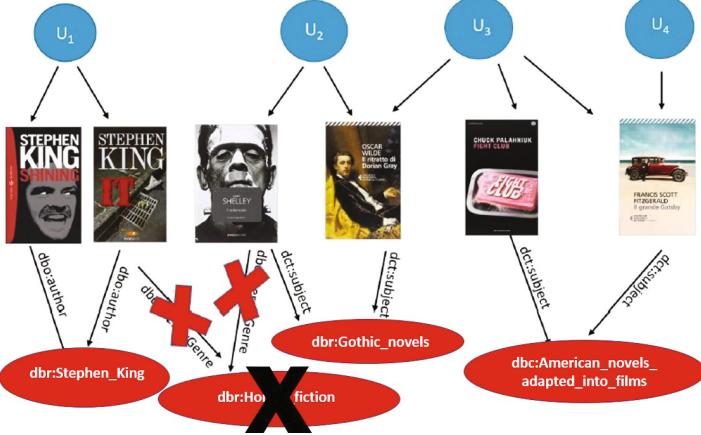


Fig. 2. The extended *tripartite* graph-based data model, including also (some of) the resources coming from DBpedia.

relevant by the algorithm. Due to space reasons, the discussion regarding the application of FS techniques on such a representation is out of the scope of the paper. However, in our previous research [14] we showed that the application of FS techniques can significantly improve the accuracy of the recommendation algorithm.

A partial example of such a representation is shown in Fig. 2. In this case, the resources connected to the properties available that are not deemed as relevant by a generic FS techniques (in this case, *dbr:Horror_fiction*) are filtered out.

3.2 Running Personalized PageRank

Given such *bipartite* and *tripartite* representations, we need an algorithm to provide each node $i \in I$ with a relevance score, in order to rank the available items and provide users with recommendations. To calculate the relevance of each item, we used a well-known variant of the PageRank called *Personalized PageRank (PPR)* [11]. In the original formulation of the PageRank an evenly distributed prior probability is assigned to each node ($\frac{1}{N}$, where N is the number of nodes). Differently from PageRank, PPR adopts a non-uniform personalization vector assigning different weights to different nodes to get a bias towards some nodes (specifically, the preferences of a specific user). As an example, Fig. 3 shows a configuration of PPR where 80% of the total weight is evenly distributed among items liked by user U_2 and 20% is evenly distributed among the remaining nodes. In this case, these values are set through a simple heuristic.

The main issue of the above described configuration is that the distribution of the weights does not consider the importance of the information coming from the LOD cloud. Indeed, all the nodes gathered from DBpedia (red nodes) are assigned with a very low probability, thus they poorly influence the random walks in the

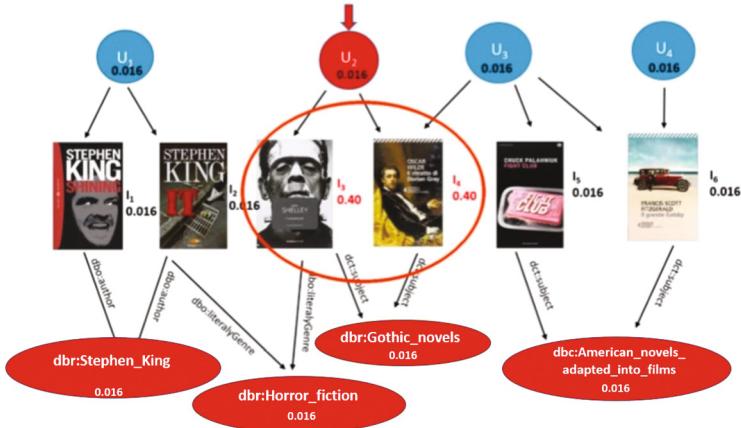


Fig. 3. Personalized PageRank distributing 80% of the weight to the items liked by User U_2 and 20% to the remaining nodes. (Color figure online)

graph and, in turn, the recommendations generated by the algorithm. This is a very important issue, since it is straightforward that a preference for a certain item also reflects some kind of preference for the *properties* of that item, as well (*e.g.*, the director, the genre of a movie, etc.), and the basic formulation of PPR ignores this aspect.

Accordingly, in this work we investigated whether different distribution of the weights in the *extended graph* may lead to an improvement of the accuracy of our recommendation strategy. As an example, in Fig. 4 we distributed half of the weight assigned to the items the user liked to the resources coming from DBpedia. In this way, it is more likely that the recommendations generated by the algorithms are also influenced by the preference the user expressed towards specific characteristics (resources) of the items she liked. In the experimental evaluation we tried to empirically validate this hypothesis by defining *different schemas* for distributing the weights among the available nodes and by analyzing their effectiveness in a *top-N recommendation task*.

4 Experimental Evaluation

In order to validate our approach, we carried out three experiments. First, we investigated whether graph-based recommender systems benefit of the introduction of LOD-based features (*Experiment 1*). Next, we evaluated whether a different distribution of the weights in Personalized PageRank may lead to better recommendations as well (*Experiment 2*). Finally, we perform a comparison with the performance of several state of the art techniques (*Experiment 3*).

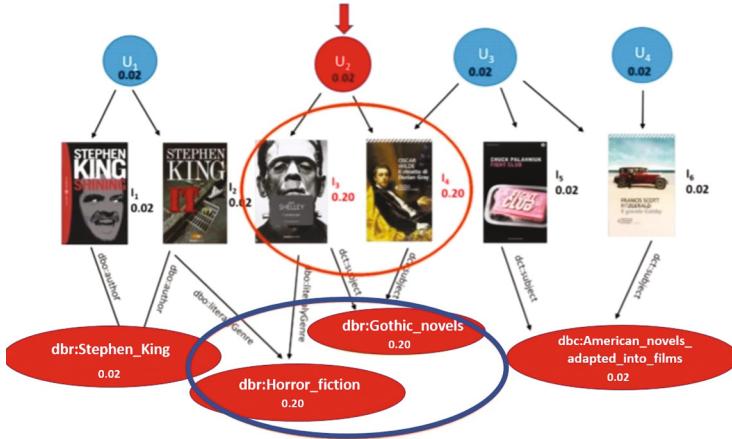


Fig. 4. Personalized PageRank distributing 40% of the weight to the items liked by User U_2 and 40% to the resources directly connected to the items. Finally, 20% is assigned to the remaining nodes.

4.1 Experimental Protocol

The evaluation was performed on two state of the art datasets, as MovieLens (ML1M)⁶, a dataset for movie recommendation, and DBbook⁷, a dataset for book recommendations which comes from the previously mentioned ESWC 2014 challenge. Some statistics about the datasets are provided in Table 1.

Experiments were performed by adopting a 5-fold cross validation as regards MovieLens, while a single training/test split for DBbook⁸. In the first case, we

Table 1. Description of the datasets.

	MovieLens	DBbook
Users	6040	6181
Items	3883	6733
Ratings	1,000,209	72372
Sparsity	96.42%	99.83%
Positive ratings	57.51%	45.85%
Avg. ratings/user $\pm \sigma$	165.59 ± 192.74	11.70 ± 5.85
Median/mode per user	96/21	11/5
Avg. ratings/item $\pm \sigma$	269.88 ± 384.04	10.74 ± 27.14
Median/mode per item	124/1	4/1

⁶ <http://grouplens.org/datasets/movielens/>.

⁷ http://www.di.uniba.it/%7Eswap/datasets/dbbooks_data.zip.

⁸ <http://challenges.2014.eswc-conferences.org/index.php/RecSys>.

built the splits on our own, while for DBBOOK we used the split used during the challenge. We face a Top- N recommendation task: given that MovieLens preferences are expressed on a 5-point Likert scale, we deem as *positive* ratings only those equal to 4 and 5. On the other side, DBbook is already available as *binarized*, thus no further processing was needed.

In order to enrich the graph G with LOD-based features, each item in the dataset was mapped to a DBpedia entry. As regards MovieLens, 3,300 movies were successfully mapped (85% of the items) while 6,600 items (98.02%) from DBbook dataset were associated to a DBpedia node. The mapping was performed by querying a DBpedia SQL Endpoint by using the *title* of the movie or the *name* of the book. Non-exact mappings were managed by exploiting a similarity measure based on Levenshtein distance⁹. We made our mappings available online¹⁰. The items for which a DBpedia entry was not found were only represented by using *collaborative* data points. MovieLens entries were described through 60 different DBpedia properties, while DBbook ones using 70 properties.

As recommendation algorithm we used the Personalized PageRank (damping factor equal to 0.85, as in [18]) and we compared the effectiveness of the graph topologies we previously introduced:

- G , which models the basics *collaborative* information about user ratings;
- G_{LOD} , which enriches G by introducing all the features gathered from DBpedia;
- G_{LOD-FS} , which filters the enriched graph by only considering a subset of the resources connected to the properties selected by a feature selection technique FS .

As regards G_{LOD-FS} configuration, we used Principal Component Analysis (PCA) and Information Gain (IG) as feature selection techniques, since in our previous research [14] they emerged as the best-performing ones for the datasets we took into account. We made available the list of the properties chosen by each algorithm¹¹. It is worth to note that also some *datatype* property (as movie length) is included in the list. For each of the above mentioned tripartite representations we evaluated *four different strategies* to distribute the weights in Personalized PageRank algorithm. The three values in the configurations describe the weight assigned to the *items* the user liked, to the *nodes* gathered from the LOD and to all the other nodes in the graph, respectively.

- 80/0/20, the original formulation which does not consider the resources gathered from DBpedia, thus only distributing 80% of the weights among the items the user liked, as in Fig. 3;
- 60/20/20, which distributes 60% of the weight among the items the user liked and gives the remaining 20% to the resources directly connected to the items;
- 40/40/20, which equally distributes the weights among the items and the resources describing the items, as in Fig. 4;

⁹ https://en.wikipedia.org/wiki/Levenshtein_distance.

¹⁰ <https://tinyurl.com/hj4q5go>.

¹¹ <http://tinyurl.com/zmfulol>.

- 20/60/20, which emphasizes the importance of the resources gathered from the LOD, giving them 60% of the weight.

To sum up, we evaluated different configurations with a fixed increase (20%) of the weight assigned to the nodes gathered from DBpedia. For each tripartite representation G_{LOD} and G_{LOD-FS} four different run of the experiments were carried out. Given that we evaluated two different FS techniques, twelve different configurations were compared in the experiments. Moreover, it is also worth to emphasize that Personalized PageRank has to be executed for each user in the dataset, since the distribution of the weights change as the preferences of the user change as well.

For each experimental session, the performance of each graph topology was evaluated by exploiting state-of-the-art ranking metrics as *F1-Measure*, *Mean Average Precision (MAP)* and *Normalized Discounted Cumulative Gain (NDCG)*. In order to ensure the reproducibility of the results, metrics were calculated by exploiting the Rival evaluation framework¹². Statistical significance was assessed by exploiting Wilcoxon and Friedman tests, chosen after running the Shapiro-Wilk test¹³, which revealed the non-normal distribution of the data. Finally, the source code of our graph-based recommendation framework has been published on GitHub¹⁴.

4.2 Discussion of the Results

Experiment 1. In the first Experiment we evaluated the effectiveness of our *tripartite* graph-based representation with respect to the original *bipartite* representation that did not include any feature gathered from the LOD cloud. Results are presented in Table 2.

As emerged from the results, the injection of the knowledge extracted from DBpedia significantly improves the accuracy of the basic bipartite representation (reported as G) for all the metrics we took into account. This validates our conjecture that the introduction of LOD-based data points positively affects the overall effectiveness of our recommendation algorithm. As expected, our approach also benefits of the adoption of feature selection techniques (with the exception of F1@5 on MovieLens data), since configurations based on PCA and IG obtained the best results on MovieLens and DBbook, respectively. The improvement is statistically significant ($p < 0.05$) only for G_{LOD} and G_{LOD-IG} while it is not for $G_{LOD-PCA}$. Even if the increase is typically tiny, the adoption of a nonparametric test as Mann-Whitney made the difference resulting as statistically significant.

It is worth to note that even without feature selection our enriched representation tend to overcome the baseline (with the exception of F1@5 on DBbook data). In this first experiment we did not modify the distribution of the weights

¹² <http://rival.recommenders.net/>.

¹³ http://en.wikipedia.org/wiki/Shapiro-Wilk_test.

¹⁴ <https://github.com/cataldomusto/lod-recsys/>.

Table 2. Results of Experiment 1. Configurations overcoming the baseline are reported in bold.

MovieLens	Weight	F1@5	MAP	NDCG
G		0.5396	0.7842	0.9003
G_LOD	80/0/20	0.5406	0.7845	0.9010
G_LOD-PCA	80/0/20	0.5398	0.7853	0.9014
G_LOD-IG	80/0/20	0.5404	0.7852	0.9013

DBbook	Weight	F1@5	MAP	NDCG
G		0.5507	0.7018	0.7861
G_LOD	80/0/20	0.5494	0.7040	0.7886
G_LOD-PCA	80/0/20	0.5508	0.7056	0.7897
G_LOD-IG	80/0/20	0.5528	0.7062	0.7898

in PPR algorithm, which is run by using the classic proportion (80% of the weights to the items the user liked, 20% to all the other nodes in the graph).

To sum up, we can conclude this first experiment by confirming our hypothesis that the exogenous knowledge encoded in DBpedia contributes to improving the accuracy of our graph-based recommender system.

Experiment 2. In the *second experiment* we validated our insight that a different weights distribution in PPR, which emphasizes the importance of nodes gathered from DBpedia, may lead to better recommendations. Our results, presented in Tables 3 and 4 for MovieLens and DBbook, respectively, gave interesting insights.

The first outcome of the experiment is the connection between the sparsity of the dataset and the optimal distribution of the weights. Indeed, the configuration that distributes most of the weight to the resources directly connected to the items the user liked (reported as 20/60/20) never overcame the baseline for all the graph topologies. This means that the when the datasets is not sparse (as for MovieLens) it is not necessary to distribute weights to the nodes gathered from the LOD, since random walks have to be mainly driven by *collaborative data points*. Moreover, a different distribution of the weights neither improve our metrics on G-LOD configurations. This is probably due to the fact that some of the movie-related resources injected in the graph are noisy, thus by distributing some of the weight to that resources it is likely that poorly relevant movies may be recommended.

The only configurations that benefit of a different weight distribution are those that use feature selection to filter out the resources connected to non-relevant properties. Indeed, all the metrics we reported got an statistically significant improvement ($p < 0.05$) on both G-LOD-PCA and G-LOD-IG graphs. Overall, the best-performing configuration is that based on IG which equally distributes the weight among item nodes and resource nodes (40/40/20). In this case we obtained a significant improvement over the baseline we took into account, thus confirming our experimental hypothesis.

Table 3. Results of Experiment 2 on MovieLens data. The baselines running PPR with the original weight distribution are highlighted with a grey background. The configurations overcoming the baselines are reported in bold, the overall best-performing configuration is highlighted with a dark grey background.

MovieLens	Weight	F1@5	MAP	NDCG
G-LOD	80/0/20	0.5406	0.7845	0.9010
G-LOD	60/20/20	0.5399	0.7833	0.9003
G-LOD	40/40/20	0.5389	0.7825	0.8999
G-LOD	20/60/20	0.5380	0.7813	0.8994
G-LOD-PCA	80/0/20	0.5398	0.7853	0.9014
G-LOD-PCA	60/20	0.5403	0.7855	0.9015
G-LOD-PCA	40/40/20	0.5401	0.7856	0.9016
G-LOD-PCA	20/60/20	0.5393	0.7847	0.9011
G-LOD-IG	80/0/20	0.5404	0.7852	0.9013
G-LOD-IG	60/20/20	0.5407	0.7852	0.9013
G-LOD-IG	40/40/20	0.5409	0.7857	0.9018
G-LOD-IG	20/60/20	0.5400	0.7851	0.9012

Our conjecture about the connection between the sparsity of the dataset and the distribution of the weights is also confirmed on DBbook. As shown in Table 4, a different distribution of the weights improve our metrics also on the complete *tripartite* graph G-LOD. This behavior is due to the high sparsity of the dataset (see Table 1), which makes collaborative data points coming from user-item connections poorly significant to properly model user preferences, thus also item-properties connections gathered from DBpedia have to be modeled as well.

In this case the best-performing configuration is again that based on Information Gain as feature selection technique and a distribution of the weights which assigns a small part of the weight (60/20/20) to the resources connected to the nodes extracted from the LOD cloud. However, also results emerging from DBbook confirmed our hypothesis since a different distribution of the weights improved the accuracy of our recommendation strategy on these data as well. All the improvements are statistically significant for $p < 0.05$.

Experiment 3. Finally, in the third experiment we compared our best-performing configuration to the results obtained by several baselines. To calculate the results, we exploited MyMediaLite Recommender System library¹⁵. As baselines, User-to-User Collaborative Filtering (U2U-KNN), Item-to-Item Collaborative Filtering (I2I-KNN), and a simple popularity-based approach suggesting the items the users (positively) voted the most were used. Moreover, we also compared our approach to Bayesian Personalized Ranking Matrix Factorization (BPRMF) presented in [20], and its extended version which also models side information, since they both emerged as the best-performing baselines in related

¹⁵ <http://www.mymedialite.net/>.

literature [17]. Due to space reasons, we only report the best performing configuration of each baseline: as regards U2U-KNN and I2I-KNN, neighborhood size was set to 80., while BPRMF was run by setting the number of latent factors equal to 100. Results are reported in Table 5. Unfortunately, it was not possible to compare our approach to other methodologies based on Linked Open Data (as [17] or [19]) since most of these approaches did not use publicly available datasets or were based on user studies.

Table 4. Results of Experiment 2 on DBbook data. The baselines running PPR with the original weight distribution are highlighted with a grey background. The configurations overcoming the baselines are reported in bold, the overall best-performing configuration is highlighted with a dark grey background.

DBbook	Weight	F1@5	MAP	NDCG
G-LOD	80/0/20	0.5494	0.7040	0.7886
G-LOD	60/20/20	0.5498	0.7041	0.7887
G-LOD	40/40/20	0.5504	0.7042	0.7888
G-LOD	20/60/20	0.5496	0.7042	0.7889
G-LOD-PCA	80/0/20	0.5507	0.7056	0.7897
G-LOD-PCA	60/20/20	0.5503	0.7054	0.7896
G-LOD-PCA	40/40/20	0.5499	0.7055	0.7899
G-LOD-PCA	20/60/20	0.5504	0.7054	0.7897
G-LOD-IG	80/0/20	0.5528	0.7062	0.7898
G-LOD-IG	60/20/20	0.5533	0.7065	0.7905
G-LOD-IG	40/40/20	0.5526	0.7064	0.7903
G-LOD-IG	20/60/20	0.5524	0.7060	0.7901

Table 5. Experiment 3. Comparison to baselines. Best-performing configurations are highlighted in grey.

MovieLens	F1@5	MAP	NDCG
G-LOD-IG (40/40)	0.5409	0.7857	0.9018
U2U	0.5220	0.7512	0.8721
I2I	0.5043	0.7230	0.8445
Popularity	0.514	0.7271	0.8602
BPRMF	0.5179	0.7338	0.8631
BPRMF+Side	0.5218	0.7498	0.8704

DBbook	F1@5	MAP	NDCG
G-LOD-IG (60/20)	0.5533	0.7065	0.7905
U2U	0.5193	0.6588	0.7417
I2I	0.5111	0.6448	0.7328
Popularity	0.5296	0.6549	0.7562
BPRMF	0.5290	0.6655	0.7589
BPRMF+Side	0.5304	0.6701	0.7631

As shown in the table, our approach based on PPR and LOD significantly outperforms all the baselines took into account. Results are particularly interesting since our methodology is able to overcome also the accuracy of widespread and well-performing baselines based on matrix factorization as BPRMF, which are commonly considered as *state-of-the-art techniques* in recommender systems community. These results definitely confirmed the soundness of our methodology as well as the insight of better distributing weights in PPR to emphasize the importance of the resources gathered from DBpedia.

5 Conclusions and Future Work

In this work we proposed a *semantics-aware recommendation methodology* based on Personalized PageRank algorithm, and we evaluated different techniques to distribute the weights in our graph-based representation. Experimental results showed that our graph-based recommender can benefit of the infusion of novel knowledge coming from the LOD cloud. Interestingly, the impact of LOD-based features is particularly positive when data are more sparse, making our framework very suitable for also for challenging recommendation settings. Finally, our methodology was able to overcome several baselines on two state of the art datasets. A publicly available implementation of the framework as well as of the splits used for the evaluation guarantee the reproducibility of the experimental results.

As future work we will investigate the impact of resources not directly connected with the items the user liked, in order to assess about the usefulness of injecting more knowledge in our representation, and we will also evaluate how a different distribution of the weights impacts on different recommender systems metrics as the *diversity* or the *serendipity* of the recommendations.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-76298-0_52](https://doi.org/10.1007/978-3-540-76298-0_52)
2. Baluja, S., Seth, R., Sivakumar, D., Jing, Y., Yagnik, J., Kumar, S., Ravichandran, D., Aly, M.: Video suggestion and discovery for YouTube: taking Random Walks through the view graph. In: Proceedings of the 17th International Conference on World Wide Web, pp. 895–904. ACM (2008)
3. Basile, P., Musto, C., de Gemmis, M., Lops, P., Narducci, F., Semeraro, G.: Aggregation strategies for linked open data-enabled recommender systems. In: European Semantic Web Conference (ESWC) (2014)
4. Basile, P., Musto, C., Gemmis, M., Lops, P., Narducci, F., Semeraro, G.: Content-based recommender systems + DBpedia knowledge = semantics-aware recommender systems. In: Presutti, V., Stankovic, M., Cambria, E., Cantador, I., Iorio, A., Noia, T., Lange, C., Reforgiato Recupero, D., Tordai, A. (eds.) SemWebEval 2014. CCIS, vol. 475, pp. 163–169. Springer, Cham (2014). doi:[10.1007/978-3-319-12024-9_21](https://doi.org/10.1007/978-3-319-12024-9_21)

5. Baumann, S., Schirru, R.: Using linked open data for novel artist recommendations. In: Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR (2012)
6. Bizer, C.: The emerging web of linked Data. *IEEE Intell. Syst.* **24**(5), 87–92 (2009)
7. Bogers, T.: Movie recommendation using Random walks over the contextual graph. In: Proceedings of the 2nd International Workshop on Context-Aware Recommender Systems (2010)
8. Bostandjiev, S., O'Donovan, J., Höllerer, T.: TasteWeights: a visual interactive hybrid recommender system. In: Proceedings of the Sixth ACM Conference on Recommender Systems, pp. 35–42. ACM (2012)
9. de Gemmis, M., Lops, P., Musto, C., Narducci, F., Semeraro, G.: Semantics-aware content-based recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.), *Recommender Systems Handbook*, pp. 119–159. Springer (2015)
10. de Gemmis, M., Lops, P., Semeraro, G., Musto, C.: An investigation on the serendipity problem in recommender systems. *Inf. Process. Manage.* **51**(5), 695–717 (2015)
11. Haveliwala, T.H.: Topic-sensitive PageRank: a context-sensitive ranking algorithm for web search. *IEEE Trans. Knowl. Data Eng.* **15**(4), 784–796 (2003)
12. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G., Althoff, K.-D.: FolkRank: A ranking algorithm for folksonomies. In: LWA, vol. 1, pp. 111–114 (2006)
13. Kantor, P.B., Rokach, L., Ricci, F., Shapira, B.: *Recommender Systems Handbook*. Springer, New York (2011)
14. Musto, C., Lops, P., Basile, P., de Gemmis, M., Semeraro, G.: Semantics-aware graph-based recommender systems exploiting linked open data. In: Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization, UMAP 2016, pp. 229–237. ACM, New York (2016)
15. Musto, C., Semeraro, G., Lops, P., Gemmis, M., Narducci, F.: Leveraging social media sources to generate personalized music playlists. In: Huemer, C., Lops, P. (eds.) EC-Web 2012. LNBP, vol. 123, pp. 112–123. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-32273-0_10](https://doi.org/10.1007/978-3-642-32273-0_10)
16. Nguyen, P.T., Tomeo, P., Noia, T., Sciascio, E.: Content-based recommendations via DBpedia and freebase: a case study in the music domain. In: Arenas, M., Corcho, O., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P., Dumontier, M., Heflin, J., Thirunarayanan, K., Staab, S. (eds.) ISWC 2015. LNCS, vol. 9366, pp. 605–621. Springer, Cham (2015). doi:[10.1007/978-3-319-25007-6_35](https://doi.org/10.1007/978-3-319-25007-6_35)
17. Ostuni, V.C., Di Noia, T., Di Sciascio, E., Mirizzi, R.: Top-N recommendations from implicit feedback leveraging linked open data. In: Proceedings of the ACM Conference on Recommender Systems, pp. 85–92. ACM (2013)
18. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web (1999)
19. Passant, A.: dbrec — Music recommendations using DBpedia. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010. LNCS, vol. 6497, pp. 209–224. Springer, Heidelberg (2010). doi:[10.1007/978-3-642-17749-1_14](https://doi.org/10.1007/978-3-642-17749-1_14)
20. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 452–461. AUAI Press (2009)