Predicting RDF Triples in Incomplete Knowledge Bases with Tensor Factorization

Lucas Drumond*
Information Systems and Machine Learning Lab
University of Hildesheim
31141 Hildesheim, Germany
ldrumond@ismll.de

Steffen Rendle
Social Network Analysis
University of Konstanz
78457 Konstanz, Germany
steffen.rendle@uni-konstanz.de

Lars Schmidt-Thieme
Information Systems and Machine Learning Lab
University of Hildesheim
31141 Hildesheim, Germany
schmidt-thieme@ismll.de

ABSTRACT

On RDF datasets, the truth values of triples are known when they are either explicitly stated or can be inferred using logical entailment. Due to the open world semantics of RDF, nothing can be said about the truth values of triples that are neither in the dataset nor can be logically inferred. By estimating the truth values of such triples, one could discover new information from the database thus enabling to broaden the scope of queries to an RDF base that can be answered, support knowledge engineers in maintaining such knowledge bases or recommend users resources worth looking into for instance. In this paper, we present a new approach to predict the truth values of any RDF triple. Our approach uses a 3-dimensional tensor representation of the RDF knowledge base and applies tensor factorization techniques that take open world semantics into account to predict new true triples given already observed ones. We report results of experiments on real world datasets comparing different tensor factorization models. Our empirical results indicate that our approach is highly successful in estimating triple truth values on incomplete RDF datasets.

1. INTRODUCTION

The Semantic Web and its goal of representing the knowledge contained in Web pages in a machine readable way has generated different standards for this task. RDF is one of such standards and also a W3C recommendation. However, just representing data with RDF is not enough without effective means to access and retrieve this data. Moreover, given the open world semantics of RDF, it is expected that RDF bases are incomplete, thus mining such data for new information is an important task. For instance, many successful approaches for accessing RDF data have been proposed [7][10][19] but they have one limitation in common: they are not able to answer queries whichs answer are not explicitly encoded in the data nor captured by inference rules. One way to answer such queries is to predict which missing RDF triples related to the query are true given the observed triples. In this paper we propose to go beyond the stored triples and predict new ones given the data, thus being able to answer queries when their answers are not explicit given in the knowledge base. Besides extending the capabilities of existing query mechanisms, such techniques can also be used to aid RDF bases maintainers or support user navigation.

RDF triples are composed by a subject, a predicate and an object. In [5], the Semantic Web data is represented as a 3-dimensional tensor where one dimension represents the subjects, one the objects and the last one the predicates. Each entry in the tensor has a value 1 for the triples in the database and 0 otherwise. PARAFAC analysis aka Canonical Decomposition [2] is then used for deriving authority and hub scores for RDF resources. Here, however, we are interested in predicting the truth value of unobserved triples and hence we argue that, in contrast to [5], one should use a sparse tensor representation, where the positions in the tensor corresponding to triples not in the database are considered to be not observed instead of having a zero value. The task is then to predict the unobserved part of the tensor given the observed one. Methodologically, this scenario is similar (but not equal) to the recommender systems one, where matrix and tensor factorization models have proven to be the best performing models up to this date [13][12]. In this work we formalize the problem of discovering new RDF triples given already existing ones and investigate how two state-of-the-art tensor factorization models, namely Canonical Decomposition [2] and Pairwise Interaction Tensor Factorization (PITF) [13], perform on this problem.

The contributions of this work are as follows:

1. We propose to provide probabilistic estimates of the truth values of RDF triples that are neither explicitly stated in the knowledge base nor can be inferred from logical entailment.

2. We approach the problem of triple prediction with tensor factorization models and argue that entries corresponding to triples not in the database should be regarded as unobserved entries rather than having value zero.

3. Our experiments show that the tensor factorization models can achieve good performance on the task approached here and show that representing RDF data as

*Scholarship holder of CNPq, an institution of the Brazilian government for scientific and technological development

SAC'12 March 26-30, 2012, Riva del Garda, Italy.

a sparse tensor yields better results than dense representations proposed in previous work. Also we observe that, by using an appropriate factorization model, filtering the results with basic RDF reasoning does not imply in a significant performance improvement.

2. RELATED WORK

There is a vast literature on approaches for querying RDF data. Many of them express RDF data in another formalism such as Frame-Logic [3][19] and Horn Logic [14] and then exploit the query capabilities of such formalisms. Another set of approaches propose RDF query languages like RQL [7] and SPARQL [10] and propose ways to evaluate queries in such languages as in [9]. Elbassuoni et al [4] proposed to augment RDF queries with keywords and use statistical information retrieval models to rank query results. Although these are relevant and successful ideas, they have one limitation in common: they do not exploit facts that are not explicitly encoded in the data and cannot be captured by inference rules. The approach proposed here relies on machine learning techniques to predict the truth value of such RDF triples. Work has been done on introducing uncertainty in RDF by attaching probabilities to triples [16] and on how to query such probabilistic RDF databases [6]. The approach proposed in this paper, however, discovers new information in traditional RDF bases.

Tensor Factorization models have been studied and applied for many years in many different fields. One prominent and general approach is Tucker decomposition [15]. A special case of Tucker decomposition (TD) is the Canonical Decomposition (CD) [2]. A special case of CD, the Pairwise Interaction Tensor Factorization (PITF) model has been proposed in [13], which is capable of capturing pairwise interactions between entities in the data. In [5], a tensor based representation of the Semantic Web is proposed. They have shown that making predictions over RDF triples can be cast as a tensor factorization task. In their work they used CD to derive authority and hub scores for RDF resources in order to rank them. Here instead, we are interested in predicting the truth values of triples that do not appear in the dataset. Nickel et al. [8] also propose a tensor factorization model for relational data. Both [8] and [5] implicitly assume that triples not appearing in the dataset are false. In the context of this work, the truth values of triples which are not in the dataset are considered to be unobserved and the goal of the models proposed here is to predict which triples are most likely to be true. Here we apply the PITF model optimized for the Bayesian Personalized Ranking (BPR) criterion (as in [13]) and analyze how the pairwise interactions modeled by this approach capture the interactions among subjects, predicates and objects of RDF triples.

3. PREDICTING RDF TRIPLES

According to the W3C RDF specification [18], an RDF dataset is a set of statements, each one of them consisting of a subject, a predicate and an object. Let S be the set of all subjects, P the set of all predicates and O the set of all objects. An RDF dataset is denoted by $T \subseteq S \times P \times O$. Also, some ontological information can be encoded in the dataset through RDF Schema (RDFS) [17]. RDFS is a basic knowledge representation language intended to structure RDF resources. Through RDFS one can define a class hierarchy, class membership relations (to which class each RDF resource belongs to) and the domain and range of RDF properties, which means that a given triple (s, p, o) can be true iff the subject s belongs to the domain of the property p, i.e. $s \in \mathit{Domain}(p)$ and the object o belongs to the range of p, i.e. $o \in \mathit{Range}(p)$.

We define the task of RDF triple prediction as generating a list of objects that, together with a given a subject predicate pair, constitute a true triple. One approach to this problem would be to generate a list with all the objects that meet the range restriction of the property in the query. However, in real world datasets, some (or most) of the information about properties ranges and domains and class membership may be missing. Furthermore, not all objects will form a true triple with the given pair. Thus it makes sense to deliver a list of objects O ranked according to the likelihood that (s, p, o) is true. If some information about the range of the property in the pair is available, the objects $o \notin \mathit{Range}(p)$ can be excluded from the ranked list. Therefore this can be formulated as a ranking problem, where the goal is to predict a total order $>_{s,p} \subset O \times O$ over objects given (s, p). One way to derive a total order is to predict a scoring function $\hat{Y} : S \times P \times O \to \mathbb{R}$ and rank the objects according to their scores. If two objects have the same score for the same pair, one is placed randomly before the other.

4. PREDICTING RDF TRIPLES BY TENSOR FACTORIZATION

As already shown in [5], RDF data can be represented as a tensor where the triples in T are the positive observations. Tensor factorization models approximate the original tensor by a set of low-rank matrices and differ in the number of such matrices and in the way they are combined to reconstruct the tensor. The low-rank matrices used are the ones that are optimal according to some loss function (e.g. the reconstruction error or some other optimization criterion suited for the task at hand). In the following we discuss the optimization criterion and the factorization models applied for predicting RDF triples.

4.1 Factorization Models

Next, we show how factorization can model the latent dependencies in RDF graphs. We discuss the Canonical decomposition (CD) or PARAFAC analysis, which has already been applied for analyzing Semantic Web data [5], as well as the more recent Pairwise Interaction Tensor Factorization (PITF) model [13]. We investigate also what the underlying assumption of each model means for representing RDF triples.

4.1.1 Canonical Decomposition

The CD model (Canonical Decomposition) decomposes a tensor $T \in \mathbb{R}^{|S|\times|P|\times|O|}$ into three matrices. Such matrices are $\hat{S} \in \mathbb{R}^{|S|\times k}$, $\hat{P} \in \mathbb{R}^{|P|\times k}$ and $\hat{O} \in \mathbb{R}^{|O|\times k}$ and constitute the parameters of the model. Through CD, each subject, predicate and object is represented as a vector of k latent features and the scoring function is given by:

$$\hat{Y}^{\mathrm{CD}}_{s,p,o} := \sum_{f=1}^{k} \hat{s}_{s,f} \cdot \hat{p}_{p,f} \cdot \hat{o}_{o,f} \qquad (1)$$

In [5] the CD model is used for analyzing Semantic Web data. Since their aim was to derive authority and hub scores for RDF resources, their approach is not suitable for triple prediction, as shown in Section 5, since it lacks an appropriate optimization criterion. Another shortcoming of this approach when predicting RDF triples (as observed in the experiments in Section 5) is that it considers that the whole tensor is observed (i.e. triples not on the data are considered to be zeros on the tensor). Thus we call it CD-Dense.

A more appropriate optimization is to use the ranking interpretation (see section 3) which leads to pairwise logistic regression (see section 4.2) and [11]). Also, due to open world assumption, it makes sense to consider that only one portion of the tensor is observed (i.e. triples that are not in the dataset are considered as unobserved data) thus making use of a sparse representation of the tensor. We will follow [11] and refer to the pairwise ranking optimization as BPR – i.e. a CD model optimized for BPR is called CD-BPR.

4.1.2 PITF

One problem with CD is that it considers only the three-wise interaction among subjects, predicates and objects. In the problem definition, however, it was stated that properties are related to subjects through their domain and to objects through their range. Interactions between subjects and objects are also relevant. For instance, musicians are more likely to be related to songs, bands and musical instruments than to birds, or touristic destinations. Thus it makes sense to use a factorization model that explicitly takes into account those pairwise interactions.

PITF decomposes a tensor $T \in \mathbb{R}^{|S|\times|P|\times|O|}$ into six matrices, namely $\hat{S}^O \in \mathbb{R}^{|S|\times k}$, $\hat{S}^P \in \mathbb{R}^{|S|\times k}$, $\hat{P}^S \in \mathbb{R}^{|P|\times k}$, $\hat{P}^O \in \mathbb{R}^{|P|\times k}$, $\hat{O}^S \in \mathbb{R}^{|O|\times k}$, and $\hat{O}^P \in \mathbb{R}^{|O|\times k}$, where k is the number of latent features. Each matrix contains the latent features involved in each pairwise interaction between subjects and objects ($\hat{S}^O$ and $\hat{O}^S$), subjects and predicates ($\hat{S}^P$ and $\hat{P}^S$) and predicates and objects ($\hat{P}^O$ and $\hat{O}^P$).

This way, PITF explicitly models the two-way interactions between subjects, predicates and objects by factorizing each of the three relationships:

$$\hat{Y}^{PITF}_{s,p,o} = \sum_{f=1}^{k} \hat{s}^O_{s,f} \cdot \hat{o}^S_{o,f} + \sum_{f=1}^{k} \hat{p}^O_{p,f} \cdot \hat{o}^P_{o,f} + \sum_{f=1}^{k} \hat{s}^P_{s,f} \cdot \hat{p}^S_{p,f} \tag{2}$$

It is important to state that the subject-predicate interaction vanishes for predicting rankings of objects for a given *(subject, predicate)* pair. Indeed, when computing the rank of an object o for a given pair (s, p), the subject and the predicate of the triple are known in advance. Therefore we want to predict only the object, thus being interested only in the interactions between s and o and between p and o. This way the matrices $\hat{S}^P$ and $\hat{P}^S$ are no longer needed. Thus, the final model parameters are $\hat{S} \in \mathbb{R}^{|S|\times k}$, $\hat{P} \in \mathbb{R}^{|P|\times k}$, $\hat{O}^S \in \mathbb{R}^{|O|\times k}$, and $\hat{O}^P \in \mathbb{R}^{|O|\times k}$ and the final model equation for PITF is:

$$\hat{Y}^{PITF}_{s,p,o} = \sum_{f=1}^{k} \hat{s}_{s,f} \cdot \hat{o}^S_{o,f} + \sum_{f=1}^{k} \hat{p}_{p,f} \cdot \hat{o}^P_{o,f} \tag{3}$$

As shown in [13], this is exactly what happens when optimizing PITF for the BPR-Opt criterion (see Section 4.2).

4.2 Dealing with the open world assumption through the BPR Framework

One crucial difference between the approaches from [5] and [8] and the one proposed here is that we consider the missing triples as unobserved data rather as negative examples. One reason for this is that, according to the RDF specification [18], open world semantics is assumed, i.e. the truth values of triples that are not in the dataset are considered to be unobserved instead of being false.

Since there are no negative examples we use the approach presented in [12] to generate training examples. Like in [13], we assume that, given a pair (s, p), an object o_A is more relevant than another object o_B iff (s, p, o_A) has been observed and (s, p, o_B) has not been observed. This can be seen as a ranking constraint, i.e. the object o_A should be ranked higher than the object o_B. The training data D_T is a set of such pairwise constraints and is defined as:

$$D_T := \{(s, p, o_A, o_B) : (s, p, o_A) \in T \wedge (s, p, o_B) \notin T\}$$

In Section 3, the triple prediction problem was formulated as a ranking problem. Thus, in order to obtain good results on this task, one should use parameters that are optimal according to a ranking criterion. Thus, both PITF and the sparse version of CD are optimized here for the Bayesian Personalized Ranking optimization criterion (BPR-Opt) [11], shown in equation 4. In order to get better rankings, BPR-Opt should be maximized, i.e. one should search for the model parameters that deliver its highest value.

$$\text{BPR-OPT} := \sum_{(s,p,o_A,o_B)\in D_T} \ln \sigma(\hat{Y}_{s,p,o_A} - \hat{Y}_{s,p,o_B}) - \lambda_\Theta ||\Theta||^2_F \tag{4}$$

In equation 4, σ is the logistic function $\sigma(x) := \frac{1}{1+e^{-x}}$, Θ are the model parameters and λ_Θ is the regularization constant. In the experiments performed in this work, the latent factor matrices of both CD-BPR and PITF are the ones that maximize BPR-Opt in the training data. They are learned using a stochastic gradient descent algorithm. For the complete derivation of this optimization criterion and the learning algorithm for optimizing the parameters for BPR-Opt, the reader is referred to [13][11].

4.3 Filtering the Results of Tensor Factorization Models

In order to improve the performance of the approach proposed here, we propose to filter out unreasonable results using range and class membership information. The filter works as follows. Given a pair (s, p), a tensor factorization model can derive a total order over the objects in O. The filter takes this order and returns an ordered list containing only the objects that belong to the range of property p, i.e. $o \in \mathit{Range}(p)$. This is done by changing the scoring function of the models as follows:

$$\hat{Y}'_{s,p,o} = \begin{cases} -\infty & \text{if } o \notin \mathit{Range}(p) \\ \hat{Y}_{s,p,o} & \text{otherwise.} \end{cases} \tag{5}$$

where $\hat{Y}_{s,p,o}$ is the scoring function of the unfiltered model.

If no information about $\mathit{Range}(p)$ is given, the results remain as they are, i.e. $\hat{Y}'_{s,p,o} = \hat{Y}_{s,p,o}$.

Table 1: Dataset characteristics in terms of subjects, predicates, objects and triples

dataset	$\|S\|$	$\|P\|$	$\|O\|$	$\|T\|$
Beatles2	158	279	5771	11728
James	21	112	851	1335
Properties	200	194	9524	11363
SWChallenge	1010	130	31180	43753

5. EVALUATION

In our evaluation, we study the prediction quality of the proposed approaches for the problem of predicting RDF triples. We investigate empirically whether Tensor Factorization models alone are able to provide reasonable predictions and whether taking the open world assumption into account translates in better results. Then, we evaluate the impact the of proposed filter on the overall performance of the best factorization models.

5.1 Datasets

We used four datasets for evaluation:

- *Beatles2* - the same dataset as used in [5]. It was extracted from dbpedia.org by crawling other resources starting from *The Beatles*;
- *James* - the same as used in [5]. It was extracted from dbpedia.org by crawling other resources starting from *James Bond*. Both *Beatles2* and *James* datasets are available for download[1];
- *Properties* - triples containing the properties of the Infobox Ontology[2]. It can be downloaded at the DBPedia website[3];
- *SWChallenge* - contains the first 1 million triples of the dataset used in the ISWC Billion Triples Challenge 2009[4].

For the *Properties* and *SWChallenge* datasets we removed the triples with subjects appearing on less than 50 triples. For the *Beatles2* and *James* datasets, this threshold was set to 10, since these datasets are smaller. The characteristics of the preprocessed datasets can be found in table 1.

5.2 Methods

The methods used in the evaluation were the following:

- Most Frequent - suggest the objects that appear in the highest number of triples on the training data;
- Most Frequent per Predicate - for a given pair (s, p), suggest the objects that co-occur most frequently with the predicate p on the training data;
- CD-Dense - Canonical decomposition on a dense tensor, i.e. triples that are not on the dataset are considered to be false. This is an application of the method in [5] to the task of triple prediction. We used the same approach for factorizing the tensor as in [5] but we used the scoring function from equation 1 instead;
- CD-BPR - sparse Canonical decomposition optimized for BPR;
- PITF-BPR - PITF model optimized for BPR.

[1] http://isweb.uni-koblenz.de/Research/DataSets

[2] http://wiki.dbpedia.org/Ontology

[3] http://wiki.dbpedia.org/Downloads

[4] http://challenge.semanticweb.org/

5.3 Evaluation Methodology

We split the data into training and test set as follows: for each subject, one subject-predicate pair is randomly chosen and all the triples containing it are put into the test set. The triples containing the other pairs with the same subject are put into the training set. What we want to verify with this evaluation protocol is whether the proposed method is able to predict correctly the triples that were hidden from it (i.e. the ones on the test set).

Once the splits were generated, the models were trained on the training set and the prediction quality on the test set was measured. This way we can evaluate if the models are able to answer the queries in the test set by predicting the triples that were actually asserted by the developer of the RDF dataset (i.e. the triples in the test set). The evaluation measures used here were the Precision and Recall in TopN-lists. Each experiment was repeated 10 times by sampling new train and test sets and applying the algorithms to them. The results reported here are the averages over the 10 runs.

The CD-BPR and PITF models were learned using the LearnBPR algorithm [13] and CD-Dense was trained using the alternating least squares algorithm implementation from the MATLAB tensor toolbox[5] [1]. The hyperparameters were searched on the first five training splits of the *Beatles2* dataset. According to these results, the learn rate was set to 0.05 and the regularization constant to 0.001 for both PITF and CD-BPR. The number of dimensions used was 64 for PITF, 45 for CD-BPR and 32 for CD-Dense. PITF ran for 500 iterations, while CD-BPR for 1000. We observed that, for CD-Dense, using higher dimensions leads to a deterioration of the prediction quality, since the model assumes that unobserved triples have value 0 on the tensor. This way, the prediction for all the test triples also gets closer to 0.

5.4 Results

Figure 1 shows the Precision-Recall curves on all datasets for TopN-lists varying from 1 to 10. There one can see that PITF achieves a higher prediction quality on all datasets. For both the smaller and larger datasets PITF outperformed CD-BPR and the other baselines. This result constitutes an evidence that the subject-object and predicate-object pairwise interactions captured by PITF do play an important role when determining new triples. The results also show that the CD-Dense is not appropriate for this task. One reason for this is the implicit assumption that triples not on the dataset are false. The values for such triples on the tensor are set to zero and the learned model tends to predict low scores for any triple that was not on the training data. Please note that in [5], this approach was developed for a different task. The results here show that using a dense tensor representation is not suitable for our scenario.

5.5 Post Filter Evaluation

This filter was evaluated on the *Beatles2* dataset. In this experiment we compare plain PITF-BPR and plain CD-

[5] Please note that this toolbox has also been used by [5] for deriving authorities and hub scores for RDF resources.

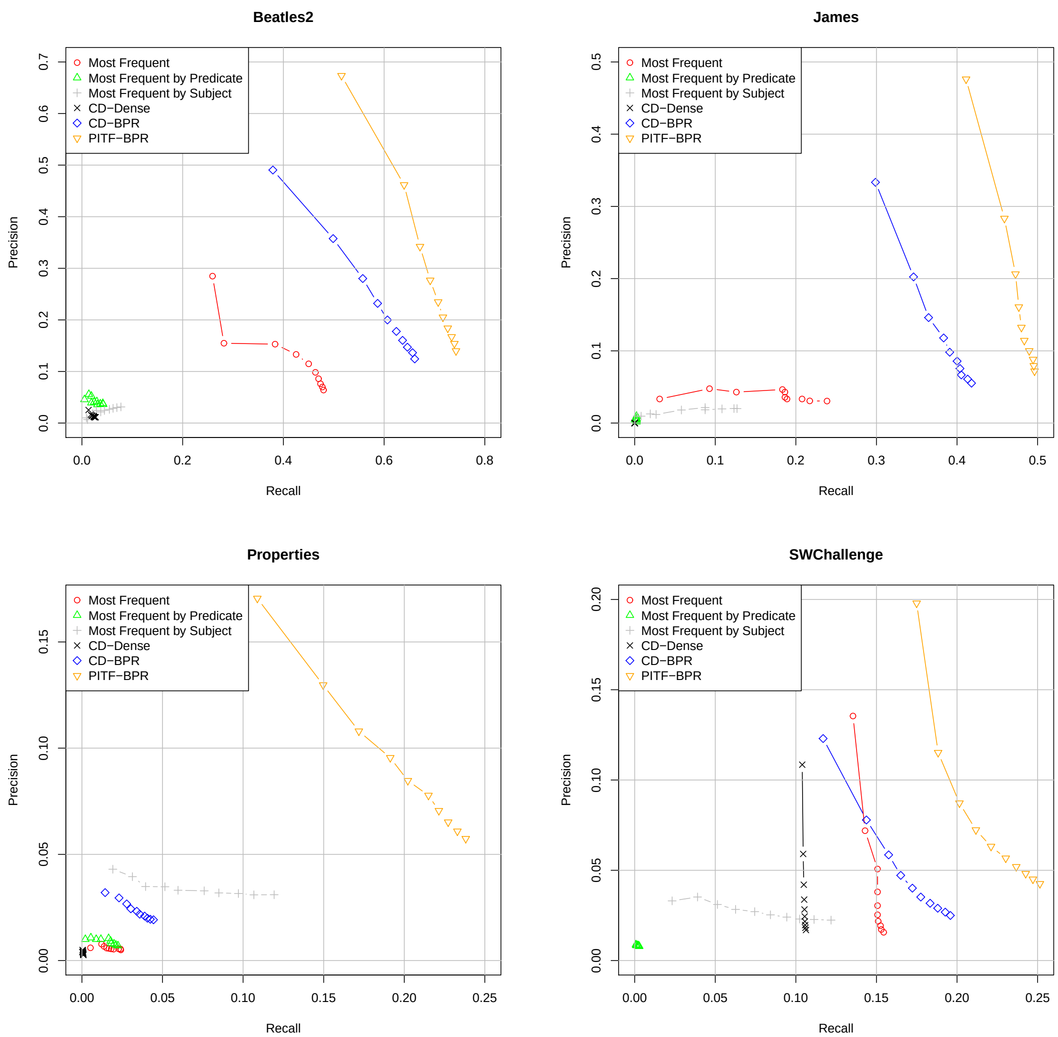

Figure 1: Precision Recall Curves for all datasets

BPR – i.e. without the filter, as in the previous experiment – against their filtered versions. The first conclusion drawn from the experiments is that the performance improvement is not significant. The main reason for this is that on this dataset (as well as on the others considered in this work) most of the information about property ranges and class membership was missing. Thus, this dataset was manually completed with the missing information (a time demanding and tedious task with huge costs, especially if one considers the size of datasets available on the Web today) and the filter was applied using the new manually added information. The results are shown in Figure 2, where three performance curves are shown: one for the plain tensor factorization model (PITF-BPR or CD-BPR), one for the factorization model plus the filter version on the original dataset and finally one for the method with filter on the manually enriched version of the dataset.

6. CONCLUSIONS

We have proposed an approach for mining RDF datasets. Our approach exploits the open world semantics of RDF data and predicts new RDF triples based on the already existing ones, thus being able to provide answers with triples that are not on the original dataset but still are true. We formalized the problem of predicting new RDF triples and approached it with tensor factorization models. We have also provided empirical evidence that, due to the open world semantics of RDF, using sparse tensors for representing RDF data (where nothing can be said about the truth value of triples not in the dataset) is a more appropriate approach for this task than using dense ones (where triples not in the dataset are considered to be false). We have shown that the PITF and CD models, when optimized for BPR, are able to provide reasonable predictions of triples that are true in the real world. It can be seen from the experiments that the quality of the results of those models can be improved by completing the datasets with some additional information with which some RDF reasoning can be made. However, the improvement (especially for the PITF model) was not significant compared to the cost of manually completing the datasets. This suggests that the pairwise interactions explicitly modeled by PITF do play an important role on RDF triple truth value prediction.

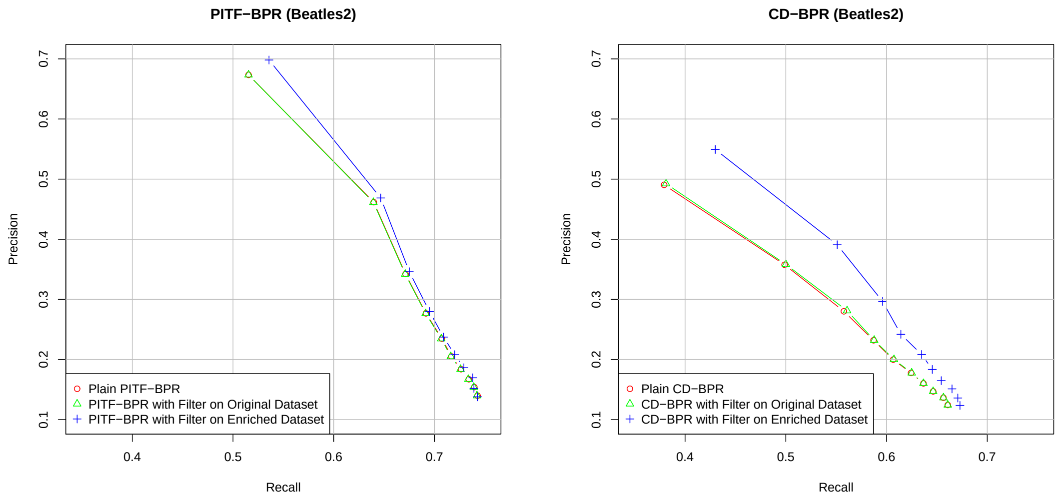

Figure 2: Comparison of the filters with and without annotations against the plain tensor factorization through PITF-BPR and CD-BPR. Note that plain PITF-BPR performs better than the filtered CD-BPR.

7. REFERENCES

[1] B. W. Bader and T. G. Kolda. Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. *ACM Transactions on Mathematical Software*, 32(4):635–653, December 2006.

[2] J. D. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. *Psychometrika*, 35(3):283–319, Sept. 1970.

[3] S. Decker, D. Brickley, J. Saarela, and J. Angele. A query and inference service for rdf. In *Online Proceedings of the QL'98 - The Query Languages Workshop*, 1998.

[4] S. Elbassuoni, M. Ramanath, R. Schenkel, M. Sydow, and G. Weikum. Language-model-based ranking for queries on rdf-graphs. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 977–986, New York, NY, USA, 2009. ACM.

[5] T. Franz, A. Schultz, S. Sizov, and S. Staab. Triplerank: Ranking semantic web data by tensor decomposition. In *International Semantic Web Conference (ISWC)*, pages 209–224. Springer-Verlag, Oct. 2009.

[6] H. Huang and C. Liu. Query evaluation on probabilistic rdf databases. In G. Vossen, D. Long, and J. Yu, editors, *Web Information Systems Engineering - WISE 2009*, volume 5802 of *Lecture Notes in Computer Science*, pages 307–320. Springer, 2009.

[7] G. Karvounarakis, A. Magganaraki, S. Alexaki, V. Christophides, D. Plexousakis, M. Scholl, and K. Tolle. Querying the semantic web with rql. *Comput. Netw.*, 42(5):617–640, 2003.

[8] M. Nickel, V. Tresp, and H.-P. Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 809–816, New York, NY, USA, June 2011. ACM.

[9] E. Oren, C. Guéret, and S. Schlobach. Anytime query answering in rdf through evolutionary algorithms. In *ISWC '08: Proceedings of the 7th International Conference on The Semantic Web*, pages 98–113, Berlin, Heidelberg, 2008. Springer-Verlag.

[10] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. Technical report, W3C, 2006.

[11] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009)*, 2009.

[12] S. Rendle, L. Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2009)*, pages 727–736, New York, NY, USA, 2009. ACM.

[13] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Third ACM International Conference on Web Search and Data Mining (WSDM 2010)*, pages 81–90, New York, NY, USA, 2010. ACM.

[14] M. Sintek, D. Gmbh, and S. Decker. Triple - an rdf query, inference, and transformation language. In *In Deductive Databases and Knowledge Management (DDLP 2001)*, October 2001.

[15] L. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):297–311, 1966.

[16] O. Udrea, V. Subrahmanian, and Z. Majkic. Probabilistic rdf. In *IEEE International Conference on Information Reuse and Integration*, pages 172–177. IEEE, 2006.

[17] W3C. *RDF Vocabulary Description Language 1.0: RDF Schema*, 2004.

[18] W3C. *Resource Description Framework (RDF): Concepts and Abstract Syntax*, 2004.

[19] G. Yang and M. Kifer. Reasoning about anonymous resources and meta statements on the semantic web. *J. Data Semantics*, 1:69–97, 2003.